

UNIVERSIDAD DE OVIEDO



ESCUELA DE INGENIERÍA INFORMÁTICA

TRABAJO FIN DE GRADO

“Plataforma para la generación automática de actas a partir de la comunicación por correo electrónico”

AUTOR

Óscar Sánchez Campo

██████████@uniovi.es

DIRECTORES

Edward Rolando Núñez Valdez

██████████@uniovi.es

Vicente García Díaz

██████████@uniovi.es

Agradecimientos

A mios pas, pol so sofitu y por trabayar ensin tasa pa que yo tea equí. A los profesores qu'amosaron el camín con procuru y bon llabor. Y a los collacios y bona xente que bogaron nesti abegosu viaxe. A toos, gracias.

A mis padres, por su apoyo y por trabajar duramente para que yo esté aquí. A los profesores que mostraron el camino, con dedicación y buen trabajo. Y a los amigos y buena gente que remaron en este complicado viaje. A todos, gracias.

Resumen

Dentro de un proyecto o trabajo colaborativo las comunicaciones entre los diferentes miembros que componen el equipo son muy importantes. En muchas ocasiones, el contenido de estas comunicaciones se pierde por el camino y no queda constancia de él. Las conversaciones se dividen entre diferentes medios, algunos en aplicaciones de mensajería, otras son fruto de la comunicación oral y a veces simplemente se quedan en una pila de correos electrónicos difícil de manejar a la hora de recuperar la información.

Para dar solución a este último problema, nace Unvíos Na Andecha (UNA), un software que permite recopilar una serie de correos y generar con ellos un acta personalizada, al estilo de las reuniones tradicionales y con el mismo objetivo: dejar constancia de las conversaciones mantenidas, bien sea para su archivo, para un acceso posterior simplificado o simplemente para la difusión de las conclusiones alcanzadas.

El proyecto que se contempla en esta documentación busca desarrollar una aplicación web que permita generar esas actas, pero no ha de quedarse ahí, sino que también permitirá incluir cálculos en ellas, y hacerlo en diferentes formatos. Y no sólo eso, también deberá ser una herramienta adecuada para un público inexperto, que requiera el menor esfuerzo posible de uso y adaptación.

En definitiva, una herramienta fácil de utilizar que permita el tratamiento de correos electrónicos para la generación de documentos formales.

Palabras Clave

Correo Electrónico, Comunicación Formal, Generación de Documentos, Aplicaciones Web, Usabilidad, Trabajo en equipo.

Abstract

When you are on a project or on a collaborative work, communication between group members is very important. Many times, the content of this communication is lost, therefore is not persistent. Conversations are divided between different media, sometimes in messaging apps, others came from oral communication, and sometimes, they just simply stay on a email stack, difficult to manage if you need to retrieve information.

To solve this last problem, Unvíos Na Andecha (UNA) is born. A software able to collect a group of mails and generate minutes based on them, just like in classic meetings and with the same aim: to reflect the conversations held, for its archive, a latter access or just to broadcast the conclusions agreed.

The project included in this documentation aims to develop a web application able to generate those minutes, but furthermore, it should also allow to include calculations on them, and even in different formats. Not only that, but it must also be a handy tool for an inexperienced user, requiring little effort for its use and acceptance.

In brief, an easy tool which must allow email processing in order to generate formal documents.

Keywords

Email, Formal Communication, Document Generation, Web Applications, Usability, Teamwork.

Índice General

CAPÍTULO 1. MEMORIA DEL PROYECTO	15
1.1 RESUMEN DE LA MOTIVACIÓN, OBJETIVOS Y ALCANCE DEL PROYECTO	15
1.2 RESUMEN DE TODOS LOS ASPECTOS	16
CAPÍTULO 2. INTRODUCCIÓN	18
2.1 JUSTIFICACIÓN DEL PROYECTO	18
2.2 OBJETIVOS DEL PROYECTO	19
2.3 ESTUDIO DE LA SITUACIÓN ACTUAL	20
2.3.1 <i>Descripción de la Situación Actual</i>	20
2.3.2 <i>Evaluación de Alternativas</i>	21
CAPÍTULO 3. ASPECTOS TEÓRICOS	24
3.1 CORREO ELECTRÓNICO.....	24
3.2 JAVAMAIL	26
3.3 SPRINGBOOT	27
3.3.1 <i>Inversión de control</i>	28
3.3.2 <i>Patrón MVC</i>	28
3.4 MAVEN.....	29
3.5 ANGULAR	29
3.6 DOCKER	30
CAPÍTULO 4. PLANIFICACIÓN DEL PROYECTO Y PRESUPUESTO INICIALES.....	32
4.1 PLANIFICACIÓN INICIAL.....	32
4.2 PRESUPUESTO INICIAL.....	36
4.2.1 <i>Presupuesto Anual de la Empresa</i>	36
4.2.2 <i>Presupuesto del Proyecto</i>	39
CAPÍTULO 5. ANÁLISIS	42
5.1 DEFINICIÓN DEL SISTEMA.....	42
5.1.1 <i>Determinación del Alcance del Sistema</i>	42
5.2 HISTORIAS DE USUARIO.....	43
5.2.1 <i>EPICs</i>	43
5.2.2 <i>Historias del Primer Sprint</i>	44
5.2.3 <i>Historias del Segundo Sprint</i>	46
5.2.4 <i>Historias del Tercer Sprint</i>	50
5.3 REQUISITOS DEL SISTEMA	57
5.3.1 <i>Obtención de los Requisitos del Sistema</i>	58
5.3.2 <i>Identificación de Actores del Sistema</i>	66
5.3.3 <i>Especificación de Casos de Uso</i>	67
5.4 IDENTIFICACIÓN DE LOS SUBSISTEMAS EN LA FASE DE ANÁLISIS	70
5.4.1 <i>Descripción de los Subsistemas</i>	70
5.4.2 <i>Descripción de los Interfaces entre Subsistemas</i>	71
5.5 DIAGRAMA DE CLASES PRELIMINAR DEL ANÁLISIS.....	72
5.5.1 <i>Diagrama de Clases</i>	72
5.5.2 <i>Descripción de las Clases</i>	73

5.6	ANÁLISIS DE CASOS DE USO Y ESCENARIOS.....	81
5.6.1	<i>Caso de Uso 1 – Abrir conexión.....</i>	81
5.6.2	<i>Caso de Uso 2 – Cambiar idioma</i>	81
5.6.3	<i>Caso de Uso 3 – Cerrar conexión</i>	82
5.6.4	<i>Caso de Uso 4 – Seleccionar correos.....</i>	82
5.6.5	<i>Caso de Uso 5 – Personalizar acta</i>	83
5.6.6	<i>Caso de Uso 6 – Generar PDF</i>	83
5.6.7	<i>Caso de Uso 7 – Generar CSV.....</i>	84
5.6.8	<i>Caso de Uso 8 – Exportar Archivos Adjuntos</i>	84
5.7	ANÁLISIS DE INTERFACES DE USUARIO	85
5.7.1	<i>Descripción de la Interfaz</i>	85
5.7.2	<i>Descripción del Comportamiento de la Interfaz</i>	103
5.8	ESPECIFICACIÓN DEL PLAN DE PRUEBAS	105
5.8.1	<i>Pruebas Unitarias</i>	105
5.8.2	<i>Pruebas de Integración</i>	106
5.8.3	<i>Pruebas de las Historias de Usuario.....</i>	106
5.8.4	<i>Pruebas Funcionales</i>	107
5.8.5	<i>Pruebas de Rendimiento.....</i>	107
5.8.6	<i>Pruebas de Accesibilidad</i>	107
5.8.7	<i>Pruebas de Usabilidad</i>	107
CAPÍTULO 6.	DISEÑO DEL SISTEMA.....	108
6.1	ARQUITECTURA DEL SISTEMA	108
6.1.1	<i>Diagramas de Paquetes.....</i>	108
6.1.2	<i>Diagramas de Componentes.....</i>	111
6.1.3	<i>Diagramas de Despliegue</i>	112
6.2	DISEÑO DE CLASES	113
6.2.1	<i>Diagrama de Clases</i>	113
6.3	DIAGRAMAS DE INTERACCIÓN.....	124
6.3.1	<i>Abrir una conexión.....</i>	124
6.3.2	<i>Cambiar idioma</i>	125
6.3.3	<i>Cerrar Conexión</i>	126
6.3.4	<i>Obtener Carpetas.....</i>	126
6.3.5	<i>Obtener Correos.....</i>	127
6.3.6	<i>Generar Acta.....</i>	129
6.4	DIAGRAMAS DE ACTIVIDADES	131
6.5	DISEÑO DE LA INTERFAZ	132
6.5.1	<i>Componente de conexión</i>	134
6.5.2	<i>Componente de selección</i>	134
6.5.3	<i>Componente de exportación.....</i>	136
6.5.4	<i>Aviso de cookies.....</i>	137
6.5.5	<i>Actas</i>	138
6.6	ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS.....	140
6.6.1	<i>Pruebas Unitarias</i>	140
6.6.2	<i>Pruebas de Integración</i>	150
6.6.3	<i>Pruebas de las Historias de Usuario.....</i>	151
6.6.4	<i>Pruebas Funcionales</i>	158
6.6.5	<i>Pruebas de Rendimiento.....</i>	159
6.6.6	<i>Pruebas de Accesibilidad</i>	160
6.6.7	<i>Pruebas de Usabilidad</i>	160

CAPÍTULO 7. IMPLEMENTACIÓN DEL SISTEMA	164
7.1 ESTÁNDARES Y NORMAS SEGUIDOS	164
7.1.1 <i>Java Code Conventions</i>	165
7.1.2 <i>JSON (JavaScript Object Notation)</i>	165
7.2 LENGUAJES DE PROGRAMACIÓN.....	166
7.2.1 <i>Java</i>	166
7.2.2 <i>TypeScript</i>	166
7.2.3 <i>HTML</i>	166
7.2.4 <i>CSS</i>	166
7.3 HERRAMIENTAS Y PROGRAMAS USADOS PARA EL DESARROLLO	168
7.3.1 <i>IntelliJ IDEA</i>	168
7.3.2 <i>WebStorm</i>	168
7.3.3 <i>Trello</i>	169
7.3.4 <i>Docker Desktop</i>	169
7.3.5 <i>Git/GitHub</i>	169
7.3.6 <i>Otras herramientas</i>	170
7.4 CREACIÓN DEL SISTEMA	171
7.4.1 <i>Problemas Encontrados</i>	171
CAPÍTULO 8. DESARROLLO DE LAS PRUEBAS	175
8.1 PRUEBAS UNITARIAS.....	175
8.2 PRUEBAS DE INTEGRACIÓN.....	175
8.3 PRUEBAS DE LAS HISTORIAS DE USUARIO	175
8.4 PRUEBAS FUNCIONALES	176
8.5 PRUEBAS DE RENDIMIENTO.....	176
8.6 PRUEBAS DE ACCESIBILIDAD	177
8.6.1 <i>Revisión Preliminar</i>	177
8.6.2 <i>Revisión Manual WCAG 2.1</i>	179
8.6.3 <i>Accesibilidad con Dispositivos Móviles</i>	185
8.7 PRUEBAS DE USABILIDAD.....	192
8.7.1 <i>Resultados de las pruebas</i>	192
8.7.2 <i>Interpretación de los resultados</i>	199
8.8 MODIFICACIONES TRAS LAS PRUEBAS	199
8.8.1 <i>Descripción de los cambios</i>	199
8.8.2 <i>Análisis</i>	207
8.8.3 <i>Diseño</i>	209
8.8.4 <i>Cambios sobre la interfaz</i>	215
8.8.5 <i>Desarrollo de los cambios</i>	217
CAPÍTULO 9. MANUALES DEL SISTEMA.....	219
9.1 MANUAL DE INSTALACIÓN	219
9.2 MANUAL DE EJECUCIÓN	220
9.3 MANUAL DE USUARIO	221
9.3.1 <i>Introducción</i>	221
9.3.2 <i>Cambiar Idioma</i>	221
9.3.3 <i>Abrir/Cerrar conexión</i>	222
9.3.4 <i>Listar Correos</i>	223
9.3.5 <i>Exportación</i>	225
9.3.6 <i>Acta</i>	227

9.3.7	<i>Cookies</i>	228
9.3.8	<i>Para programadores</i>	229
9.4	MANUAL DEL PROGRAMADOR.....	231
9.4.1	<i>Hosts</i>	231
9.4.2	<i>Opciones de exportación</i>	232
9.4.3	<i>Eliminador de Respuestas</i>	232
CAPÍTULO 10.	CONCLUSIONES Y AMPLIACIONES	234
10.1	CONCLUSIONES.....	234
10.2	AMPLIACIONES.....	236
CAPÍTULO 11.	PLANIFICACIÓN DEL PROYECTO Y PRESUPUESTO FINALES	237
11.1	PLANIFICACIÓN FINAL.....	237
11.1.1	<i>Primer Sprint</i>	239
11.1.2	<i>Segundo Sprint</i>	240
11.1.3	<i>Tercer Sprint</i>	241
11.2	PRESUPUESTO FINAL.....	242
11.2.1	<i>Presupuesto de Costes</i>	242
11.2.2	<i>Presupuesto de Cliente</i>	243
CAPÍTULO 12.	REFERENCIAS BIBLIOGRÁFICAS	245
CAPÍTULO 13.	APÉNDICES	247
13.1	GLOSARIO Y DICCIONARIO DE DATOS.....	247
13.2	CONTENIDO ENTREGADO EN EL ARCHIVO ADJUNTO.....	249
13.2.1	<i>Contenidos</i>	249
13.2.2	<i>Código Ejecutable e Instalación</i>	250
13.3	ÍNDICE ALFABÉTICO.....	252

Índice de Figuras

Figura 1 – Arquitectura del correo electrónico	25
Figura 2 – Acceso a los correos usando JavaMail.....	27
Figura 3 – Comparación entre máquinas virtuales y Docker	31
Figura 4 – Planificación Inicial del Proyecto	33
Figura 5 – Casos de Uso del Usuario no conectado.....	67
Figura 6 – Casos de Uso del usuario conectado	68
Figura 7 – Diagrama de Clases del front-end	72
Figura 8 – Diagrama de Clases del back-end.....	72
Figura 9 – Interfaz del componente de conexión para la historia H1	86
Figura 10 – Interfaz del componente de conexión para la historia H2	87
Figura 11 – Pantalla principal de la conexión con los cambios de la historia H3.	88
Figura 12 – Pantalla principal de la aplicación, tras la historia H4.	89
Figura 13 – Propuesta de interfaz para la pantalla principal en la historia H5.	90
Figura 14 – Propuesta de interfaz inicial para el componente de exportación.	91
Figura 15 – Interfaz del componente de correos en la historia H6.....	92
Figura 16 – Modificaciones sobre el componente de exportación (Historia H7)	93
Figura 17 – Pantalla principal de la aplicación UNA (Historia H8)	94
Figura 18 – Propuesta aceptada de interfaz para la paginación.....	95
Figura 19 – Interfaz para la generación de actas PDF	96
Figura 20 – Interfaz para la generación de actas CSV	96
Figura 21 – Componente de conexión renovado para la historia H15.	97
Figura 22 – Lista de correos con las modificaciones de la historia H16.	98
Figura 23 – Propuesta de interfaz para el aviso de cookies.	99
Figura 24 – Componente de conexión con los iconos de la historia H19.....	100
Figura 25 – Posicionamiento del aviso de la historia H19.	100
Figura 26 – Distribución de botones para la historia H20.	101
Figura 27 – Posición del botón de “vuelta atrás” para la historia H22.	102
Figura 28. Pirámide de ‘Testing’	105
Figura 29 – Diagrama de paquetes del front-end	108
Figura 30 – Diagrama de paquetes del back-end.....	109
Figura 31 – Diagrama de componentes del front-end	111
Figura 32 – Diagrama de componentes del back-end	112
Figura 33 – Diagrama de despliegue de la aplicación	113
Figura 34 – Diagrama general de clases del back-end	115
Figura 35 – Diagrama de clases (paquete beans).....	116
Figura 36 – Diagrama de clases (paquete controladores)	117
Figura 37 – Diagrama de clases (paquete entities)	118
Figura 38 – Diagrama de clases (paquete exceptions)	118
Figura 39 - Diagrama de clases (paquete generators).....	119
Figura 40 – Diagrama de clases (paquete gestores).....	120
Figura 41 – Diagrama de clases (paquete parsers)	120
Figura 42 – Diagrama de clases (paquete parsers)	121
Figura 43 – Diagrama de clases (paquete util).....	122
Figura 44 – Diagrama de clases del front-end.....	123
Figura 45 – Diagrama de Interacción (Abrir conexión).....	125

Figura 46 – Diagrama de Interacción (Cambiar Idioma).....	125
Figura 47 – Diagrama de Interacción (Obtener carpetas)	127
Figura 48 – Diagrama de interacción (Obtener correos)	128
Figura 49 – Diagrama de interacción (Generar acta)	129
Figura 50 – Diagrama de actividades (Generar un acta/Exportar archivos)	131
Figura 51 – Vista General de la Aplicación (Listado)	132
Figura 52 – Vista general de la aplicación (Exportación).....	133
Figura 53 – Componente de conexión	134
Figura 54 – Aclaraciones para la conexión con Gmail	134
Figura 55 – Listado de correos.....	135
Figura 56 – Detalles de un correo seleccionado	135
Figura 57 – Componente de exportación para archivos PDF	136
Figura 58 – Componente de exportación para archivos CSV.....	137
Figura 59 – Aviso de Cookies	137
Figura 60 – Icono para mostrar/ocultar el aviso de cookies.	138
Figura 61 – Ejemplo de acta generada.....	139
Figura 62 – Logo Java ©Oracle	166
Figura 63 – Logo TypeScript ©Microsoft	166
Figura 64 – Logo HTML ©W3C	166
Figura 65 – Logo CSS © W3C	167
Figura 66 – Logo IntelliJ ©JetBrains	168
Figura 67 – WebStorm © JetBrains	168
Figura 68 – Tablero Trello del Proyecto	169
Figura 69 – Resumen de los resultados de las pruebas con WAVE	178
Figura 70 - Detalle de los errores encontrados en las pruebas con WAVE	178
Figura 71 - Detalle de la interfaz	179
Figura 72 – Resumen del cumplimiento de las directrices WCAG por niveles tras el análisis manual de la accesibilidad.	183
Figura 73 – Resumen del cumplimiento de las directrices del WCAG por niveles tras los cambios realizados.	185
Figura 74 – Problema 1 (Detección)	186
Figura 75 – Problema 1 (Solución).....	187
Figura 76 – Problema 2 (Detección)	188
Figura 77 – Problema 2 (Solución).....	188
Figura 78 - Problema 3 (Detección)	189
Figura 79 – Problema 3 (Solución).....	190
Figura 80 – Problema 3 (Solución).....	190
Figura 81 – Diagrama de clases del back-end modificado.....	208
Figura 82 – Diagrama E-R de la Base de Datos.....	210
Figura 83 – Diagrama de paquetes modificado	210
Figura 84 – Diagrama de componentes modificado	211
Figura 85 – Diagrama de despliegue modificado	212
Figura 87 – Diagrama de interacción (sincronizar base de datos)	213
Figura 88 – Diagrama de interacción modificado (obtener carpetas)	214
Figura 88 – Diagrama de interacción modificado (exportación)	215
Figura 89 – Interfaz del componente de listado de correos.	216
Figura 90 – Nuevo acta mejorado.	217
Figura 91 – Componente cambio de idioma	222
Figura 92 – Componente abrir/cerrar conexión	222
Figura 93 – Componente Listado de Correos	224

Figura 94 – Datos del correo (extendido)	225
Figura 95 – Componente de exportación	226
Figura 96 - Acta	228
Figura 97 – Paquete elimina_respuestas	232
Figura 99 – Planificación Final del Proyecto	238
Figura 99 – Burndown Chart del primer sprint.	240
Figura 100 – Burndown Chart del segundo sprint.....	241
Figura 101 – Burndown Chart del tercer sprint	241

Índice de Tablas

Tabla 1 – Fechas y duración de los ‘sprints’	34
Tabla 2 – Calendario de Pruebas	36
Tabla 3 – Costes anuales de los servicios de la empresa.....	37
Tabla 4 – Costes de los equipos de la empresa.....	37
Tabla 5 – Costes de las licencias de la empresa	37
Tabla 6 – Cálculo de los costes salariales de los empleados de la empresa	38
Tabla 7 – Costes y beneficios anuales de la empresa divididos por tipo	38
Tabla 8 – Resumen de los costes salariales del personal	39
Tabla 9 – Resumen de costes finales de los empleados.....	39
Tabla 10 – Presupuesto de Costes del Proyecto	40
Tabla 11 – Presupuesto de Cliente del Proyecto	41
Tabla 12 – Resumen por partidas del presupuesto de cliente	41
Tabla 13 – Lista de requisitos funcionales del primer sprint.....	60
Tabla 14 – Requisitos funcionales del segundo sprint	62
Tabla 15 – Requisitos funcionales del tercer sprint	65
Tabla 16 – Lista de requisitos no funcionales	66
Tabla 17 – Resultados de las pruebas de rendimiento	177
Tabla 18 – Resultados de las pruebas de accesibilidad	183
Tabla 19 – Lista de requisitos funcionales tras las pruebas	207
Tabla 20 – Referencia de la API	230
Tabla 21 – Resumen final de fechas del sprint.....	239
Tabla 22 – Presupuesto de costes final del proyecto.....	243
Tabla 23 – Presupuesto de cliente final del proyecto	244
Tabla 24 – Resumen final de las partidas del proyecto.....	244

Capítulo 1. Memoria del Proyecto

En esta documentación se contempla un conjunto de aspectos del proyecto tales como la justificación, su análisis y desarrollo, sus manuales de uso e incluso aspectos de gestión como la planificación o su presupuesto. También se contienen las instrucciones para la localización, instalación, despliegue y ejecución del código del proyecto. En este primer capítulo veremos un resumen de todos los aspectos que se incluyen.

1.1 Resumen de la Motivación, Objetivos y Alcance del Proyecto

El correo electrónico es una de las herramientas de comunicación más usadas en todo el mundo, especialmente en el ámbito laboral y educativo. Muchas de las conversaciones y decisiones tomadas en estos ambientes, quedan perdidas entre un montón de correos, difíciles de separar, ordenar y localizar.

La aplicación que aquí se contempla pretende dar solución a esto, permitiendo el tratamiento de esos correos de manera que se pueda generar con ellos un documento que sirva como acta. Este es el objetivo principal, pero hay una serie de subobjetivos que se deberán lograr para un producto satisfactorio: aspectos tales como la usabilidad, la portabilidad y la personalización serán de vital importancia.

El producto será una aplicación web con varias capas que permita realizar una conexión con un proveedor de correo de manera que se puedan mostrar los mensajes allí contenidos. Una vez realizada, el usuario podrá navegar por ellos, seleccionando los que le interese añadir al acta. Posteriormente podrá personalizar el acta, añadiendo campos de texto, cálculos y otras personalizaciones de ese estilo. Finalmente se podrá descargar el documento en el formato que desee y sus archivos adjuntos.

1.2 Resumen de Todos los Aspectos

En las próximas líneas se resumirá brevemente el contenido del resto de capítulos.

Capítulo 2. Introducción. Se comienza indicando cuál es la necesidad que pretende cubrir el proyecto, los objetivos que se quieren conseguir y cuál es el estado del arte en este ámbito. También se incluyen una serie de decisiones justificadas respecto a ciertos aspectos básicos del proyecto.

Capítulo 3. Aspectos Teóricos. Antes de centrarnos en el proyecto en sí, introducimos una serie de conceptos teóricos de cierta importancia, con el objetivo de que el lector refuerce su conocimiento de ellos y pueda comprender mejor los capítulos siguientes.

Capítulo 4. Planificación del Proyecto y Presupuesto Iniciales. Aquí se plantearán las tareas que será necesarias para llevar a completar el proyecto y se estimará su duración. La metodología de trabajo se establece aquí también. Basándose en la estimación realizada, se planteará el presupuesto.

Capítulo 5. Análisis. Este apartado contiene todos los artefactos (historias de usuario, requisitos, diagramas) utilizados en las fases de análisis del proyecto. Es una buena referencia para ver qué funcionalidades concretas se plantean y que restricciones puede haber.

Capítulo 6. Diseño del Sistema. Se recopilan en este punto los diagramas y especificaciones concretas de la aplicación y de su plan de pruebas. También las interfaces finales. Todo ello es una evolución de lo analizado en el capítulo anterior.

Capítulo 7. Implementación del Sistema. Este capítulo se reserva para la descripción de las herramientas y lenguajes usados y, principalmente, para describir detalladamente los problemas encontrados durante esta fase, así como las decisiones tomadas.

Capítulo 8. Desarrollo de las Pruebas. Los resultados de los diferentes *tests* llevados a cabo para la validación y verificación del producto se incluyen aquí. Junto a ellos se describen las acciones correctivas llevadas a cabo.

Capítulo 9. Manuales del Sistema. Para la instalación, despliegue, uso y modificación de la aplicación se incluyen varios manuales con las indicaciones oportunas. Se recomienda usarlos si se va a probar la aplicación.

Capítulo 10. Conclusiones y Ampliaciones. Una vez vistos los aspectos técnicos del proyecto se pasa a hacer balance del trabajo realizado. También se fijan aquí las directrices para una ampliación futura de la aplicación, con una lista de posibles mejoras.

Capítulo 11. Planificación del Proyecto y Presupuesto finales. Recordando lo planteado en el capítulo 4, se compara con la planificación y presupuesto reales, es decir, aquellos que se generan del desempeño del trabajo realizado.

Capítulo 12. Referencias Bibliográficas. Las fuentes a las que se hace referencia en la documentación se incluyen aquí.

Capítulo 13. Apéndices. Una miscelánea de aspectos tales como el glosario, el índice alfabético y la enumeración de los archivos adjuntos.

Capítulo 2. Introducción

En este capítulo realizaremos una aproximación general al proyecto. Ha de entenderse como una puesta en contexto de los puntos que después trataremos. Aquí explicaremos la necesidad de un producto de este estilo y los objetivos que pretendemos lograr con él. También se incluye la argumentación de importantes decisiones estructurales que marcarán el devenir del proyecto.

2.1 Justificación del Proyecto

Hoy en día la mitad de la población mundial dispone de una cuenta de correo electrónico [1]. Además de las cuentas personales que puedan tener, casi todas las instituciones, empresas, asociaciones y otras organizaciones brindan a sus miembros una cuenta con la que comunicarse dentro de su ámbito. Y no son una herramienta más, se usan de manera frecuente para múltiples temas. Desde difundir una información relevante para los miembros, mantener conversaciones con los compañeros sobre diferentes temas, comunicarse con los clientes o transferir documentos relevantes. Infinidad de motivos son los que hay. No es menester desgranarlos ni enumerarlos aquí, simplemente dejar patente la importancia de su uso.

El correo electrónico presenta ventajas sobre otro tipo de comunicaciones. Por ejemplo, sobre un servicio de mensajería instantánea, como Whatsapp o Telegram. Mientras que la inmediatez y sencillez de estos les hace más ágiles a la hora de comunicarse rápidamente, el correo electrónico es mucho más útil a la hora de transmitir información densa y extensa, de adjuntar archivos y de maquetarla mejor para que se entienda claramente. Por este motivo, en el ámbito profesional, los correos electrónicos son un medio muy importante.

Sin embargo, uno de los grandes problemas que tiene, es la dificultad para organizar semejante vorágine de correos entrantes. Si bien es cierto que muchos clientes de correo (Gmail, Outlook...) nos permiten verlos organizados en sistemas de carpetas, es muy frecuente que nos cueste encontrar una conversación pasada. Muchas veces hablamos de un tema importante y queremos dejar constancia de la conversación. En las reuniones mantenidas en el ámbito profesional, suele hacerse por medio de un acta, pero cuando estas no tienen un lugar y hora fijos, si no que se mantienen de manera asíncrona entre varios participantes y a través de correos electrónicos, nos topamos con el problema de dejar constancia de los temas discutidos y las decisiones tomadas.

Otra situación que provocan los correos electrónicos es que es difícil tratarlos a gran escala de manera sencilla. Por ejemplo, si quieres guardar una conversación con las instrucciones de uso de una determinada herramienta, simplemente necesitas un subconjunto de todos los mensajes enviados y sólo el contenido de los mismos. Así que podríamos deshacernos de un montón de datos como la hora de los mensajes y los destinatarios.

Además, no podemos procesarlos a gran escala, por ejemplo, si queremos contar el número de destinatarios diferentes a los que enviamos correos a lo largo del mes pasado, no podremos hacerlo de manera sencilla. Aunque si generásemos un archivo CSV con los destinatarios de nuestros correos enviados, podríamos con una función de Microsoft Excel, contarlos en muy pocos segundos.

En definitiva, la necesidad de poder almacenar de manera eficiente los correos importantes, separándolos de los que no lo son y poder generar documentos con la información estrictamente necesaria está aún sin cubrir.

Cada plataforma nos propone una solución diferente a este problema. En esencia, lo mejor que nos dan es poder imprimir los correos (o guardarlos en PDF) con el formato que ellos te den. Tampoco nos ofrecen casi ninguna posibilidad de pasarlos a un formato que podamos procesar rápidamente, por ejemplo, CSV, o en el caso de proporcionarla es difícil de usar para el usuario inexperto.

La idea de este proyecto, Unvíos na Andecha (UNA), es dar una solución a estos problemas, desarrollando un software que nos permita escoger, de entre nuestros correos electrónicos, aquellos que queramos exportar, y nos deje personalizar que datos queremos que se muestren en los archivos generados.

2.2 Objetivos del Proyecto

Viendo las necesidades que hay de un producto de este estilo, el objetivo de este proyecto es desarrollar un software que proporcione de manera sencilla al usuario la posibilidad de generar actas a partir de correos electrónicos, así como otras funcionalidades de exportación de estos. Esto lo lograremos cumpliendo una serie de objetivos menores:

- La aplicación debe poder conectarse al servidor de correo del usuario de manera sencilla, para poder descargarse los correos que este haya recibido/enviado.
- También tiene que permitirnos seleccionar esos correos y exportarlos a diferentes formatos, como PDF (Portable Document Format) o CSV (Comma Separated Values), conteniendo la información que el usuario decida.
- El PDF que se exporte debe tener el formato de un acta de reunión, añadiéndole a los correos electrónicos la información necesaria para que así sea.
- La aplicación debe ser muy sencilla de utilizar. Cualquier usuario que sepa usar las funcionalidades básicas del correo electrónico debería ser capaz de manejarse con ella sin ninguna complicación.
- Su adaptabilidad a múltiples proveedores debe ser otro punto importante que desarrollar, pues la sencillez de tratarlos a todos desde una única aplicación puede ser clave a la hora de captar usuarios.
- No sólo se busca la interacción con el usuario final. También proporcionará una API, para que cualquiera pueda acceder sin necesidad de utilizar una interfaz gráfica y facilitar el tratamiento masivo de los datos de correo propios.

- La internacionalización no solo es un objetivo, es una necesidad. Los usuarios deberán poder navegar tanto en castellano, como en asturiano o inglés.

En esencia, estos son los objetivos que se pretenden alcanzar con el desarrollo de este proyecto y serán en los que principalmente nos centraremos en todas las fases del mismo.

2.3 Estudio de la Situación Actual

En la sección anterior, indicamos la falta de un sistema de exportación adecuado para los usuarios no expertos, en los clientes de correo electrónico. Si bien algunos sí brindan opciones para esto, son complejas y poco manejables y/o personalizables. Por ello, en las próximas líneas, explicaremos en detalle lo que verdaderamente ofrecen, siendo esta la situación actual. Nos centraremos en dos de los clientes más usados, Outlook y Gmail.

También examinaremos diferentes alternativas al desarrollo de esta aplicación y los motivos que hacen viable y adecuado su desarrollo.

2.3.1 Descripción de la Situación Actual

Cada cliente de correo electrónico ofrece unas posibilidades diferentes. No es necesario hacer aquí un análisis exhaustivo de todas ellas, así que vamos a centrarnos en dos de los más usados, Outlook y Gmail. De esta manera mostramos las opciones que tienen la mayoría de usuarios de correo electrónico.

2.3.1.1 Outlook

Este cliente de correo, desarrollado por Microsoft, ofrece al usuario varias alternativas de exportación.

Por un lado, se pueden guardar correos individuales o conversaciones en múltiples formatos (texto plano, HTML, el formato propio de mensajes de Outlook...). También pueden imprimirse, y por tanto guardarse como PDF. Sin embargo, esto está destinado a ser procesado posteriormente por otras máquinas, o como una manera muy rudimentaria de archivo. No son personalizables las exportaciones, es decir, no se puede elegir que campos mostrar o añadirle títulos, descripciones, fechas ni nada. Tampoco se imprimen de manera elegante, simplemente consiste en copiar lo que se ve en pantalla a otro formato.

También existe la exportación masiva de correos, pues se puede descargar una carpeta entera de en formato CSV. Sin embargo, tampoco es manejable, pues volvemos al problema de la falta de personalización. Tampoco se podría usar de manera sencilla, pues es bastante espartano intentar tratar los correos, por ejemplo, con Microsoft Excel, ya que no nos deja seleccionar un subconjunto de ellos. Habría que hacer todo sobre la carpeta entera.

2.3.1.2 Gmail

De la misma manera, Google, en su cliente de correo, Gmail, permite imprimir conversaciones enteras o correos únicos. Esto es sencillo, simplemente se hace usando un icono, pero, al igual que Outlook, no permite elegir que ver y qué no en el archivo impreso, pues vuelve a ser una pura copia de lo que está en pantalla.

La exportación masiva es aún peor y tiene ciertas peculiaridades. Google tiene una página donde se centraliza la obtención de los datos de usuario que la empresa almacena desde sus diferentes productos. Para la parte de Gmail, sólo permite guardar los correos en formato MBOX. Además, no te genera el archivo al momento, si no que tienes que esperar horas (o incluso días, según su advertencia) a que lo genere y te envíe un enlace para descargarlo.

2.3.2 Evaluación de Alternativas

Uno de los primeros pasos en un proyecto es decidir cómo se van a hacer las cosas, pues pueden ocurrirnos muchas soluciones para el problema que se nos plantea. Por tanto, en este punto, propondremos diferentes alternativas y veremos qué es lo que más nos conviene. En las próximas líneas detallaremos las disyuntivas más importantes a las que nos hemos enfrentado en esta fase decisoria.

2.3.2.1 Extensión

Lo primero que podríamos pensar es: “si le falta alguna funcionalidad al cliente de correo, añadámosela”. Por tanto, la idea de crear una extensión (*plugin, add-on...*), es lo primero que nos planteamos. Tanto Outlook como Gmail, tienen un sistema de extensiones que permite añadirles funcionalidades. Puede ser más sencillo crear un plugin de estos que desarrollar una aplicación completa como la que pretendemos, ya que los métodos que proporcionen sus API (*Application Programming Interface*) pueden ser más manejables que una librería genérica de tratamiento de correo. Sin embargo, nos estaríamos saltando uno de los Objetivos del Proyecto, el ser multi-plataforma, que nos integre varios servicios de correo electrónico. Por este motivo, hemos descartado hacer una extensión.

2.3.2.2 Número de capas

Una vez decidido que vamos a hacer una aplicación independiente, tenemos que ver cómo vamos a hacerlo. Partiendo de la base de que queremos proporcionar una API para que puedan acceder los usuarios expertos y de que también buscamos que el usuario inexperto pueda usarla sin mucha dificultad, evidentemente a través de una GUI (*Graphic User Interface*), que la aplicación tenga dos capas es la conclusión natural. Por tanto, debemos tener un *front-end* y un *back-end*. Estos serán independientes entre sí, ya que cualquier otra posibilidad complicaría el diseño innecesariamente y daría problemas de usabilidad y mantenimiento.

2.3.2.3 Elección del back-end

Para esta parte de la aplicación debemos usar una tecnología fiable. Que sirva tanto para acceder desde un *front-end*, como a través de llamadas GET y POST directamente del usuario. Para esto hay muchas tecnologías válidas, de las que podemos valorar principalmente dos aspectos, la agilidad de cara al desarrollo y la seguridad que nos proporciona, tanto a nivel de código como de configuración. La experiencia del equipo usando las diferentes tecnologías también debe tenerse en cuenta. Por todos estos motivos hemos decidido usar SpringBoot para el *back-end*, ya que combina varias de estas cualidades, una es que requiere poca configuración, por ejemplo, respecto a Spring. Aunque no sería tan sencillo como, por ejemplo, crear un servicio web en Python, si nos proporciona frente a *frameworks* de este estilo la seguridad que da el lenguaje Java, evitando muchos errores en tiempo de ejecución que se detectan en compilación. Finalmente, aunque tengo cierta experiencia con varios, SpringBoot es uno de los que más domino, por haberlo usado durante cierto tiempo en el terreno profesional.

2.3.2.4 Elección del front-end

Una vez elegido el *back-end* llega la hora de elegir el *front-end*. Aquí consideramos dos alternativas, usar un motor de plantillas como Thymeleaf o una *Single-Page Application* (SPA) por ejemplo con React o Angular.

La decisión final ha sido Angular, por varios motivos. Lo primero en descartarse fue usar un motor de plantillas, pues ya que uno de los Objetivos del Proyecto es una excelente experiencia de usuario, suele ser más sencillo de lograr usando una SPA. También facilita enormemente la reutilización de código y además podemos usar la API del back-end directamente. Respecto al uso de Angular frente a otras librerías, la experiencia personal utilizando este *framework* es lo que nos ha hecho decidirnos, aunque React también sería una buena opción.

2.3.2.5 Uso de una base de datos

Teniendo la división en capas clara, hay que pensar la necesidad de una base de datos. Normalmente es un sí en todas las aplicaciones que se diseñan, y, de hecho, muchas veces es el “corazón” de las mismas, actuando como el repositorio principal del que se obtienen los datos. Sin embargo, pensándolo fríamente, esta aplicación que vamos a diseñar tiene otro repositorio central, alrededor del cual construirse. Es el Servidor de Correo del proveedor que use el cliente. UNA accederá a través de él a los correos del usuario. Las funcionalidades que tendrá la aplicación no necesitarán almacenarlos, ni ninguna información relativa a ellos y trabajará con los datos que obtenga. La mayoría de las funciones de la aplicación serán accesos a esos correos, o procesamientos de otros anteriormente obtenidos. Esto implica que podemos prescindir de una base de datos, al menos en cuanto a la relación con los correos. De esta manera, evitaremos introducir complejidad innecesaria en el sistema, eliminando riesgos de seguridad, que tendríamos que considerar muy seriamente si decidiésemos almacenar información privada del usuario, como son sus correos.

2.3.2.6 Autenticación

Pero en las aplicaciones, no sólo tenemos los datos puros del modelo. También suele ser necesario almacenar otros datos, como los usuarios registrados en el sistema. Esta funcionalidad es muy común y permite acceder a los datos privados almacenados o a determinadas funciones propias de usuarios autenticados. Por tanto, debemos considerar si UNA necesita un sistema de autenticación, en cuyo caso sí habría que incluir una base de datos en la que almacenar las credenciales de estos.

¿Qué ventajas proporcionaría un sistema de autenticación en la aplicación? Pues de tenerlo, los usuarios podrían introducir sus cuentas de correo y almacenar las credenciales de estas, permitiendo el acceso a diferentes proveedores con un solo clic. También podría llevarse algún tipo de historial de acciones realizadas, como, por ejemplo, las actas generadas previamente.

¿Qué inconvenientes produciría? Además del incremento necesario en medidas de seguridad que indicábamos en el punto anterior, un sistema de autenticación obligaría al usuario a recordar una contraseña más, que debe ser ciertamente compleja, pues no hay que olvidar, que, si se filtrase, todas las cuentas de correo que el usuario incluyese en la aplicación estarían gravemente comprometidas. Con medidas de seguridad adecuadas podría incluirse, usando métodos como la autenticación en dos pasos o similar. Sin embargo, tampoco proporciona grandes funcionalidades. El almacenamiento de contraseñas es algo que los navegadores tienen integrado hoy en día, con un par de acciones ya tenemos nuestras credenciales puestas, sin necesidad de recordarlas en cada acceso. Si bien es cierto, que, si cambiamos de equipo o usamos uno público, no podemos usar este método y deberemos introducir los datos a mano. Este es un inconveniente que ya tenemos cuando nos autenticamos en nuestro cliente de correo.

Por otro lado, el historial de acciones realizadas podría estar bien. Sin embargo, es una funcionalidad tan simple, que creemos que no justifica tanto tiempo de desarrollo. Podría hacerse más compleja, con cosas como informes detallados de las acciones, o poder repetirlos. Pero todo esto requeriría grandes almacenamientos de datos y una lógica compleja que escapa de los verdaderos objetivos de la aplicación. La sencillez de uso y similitud con un cliente de correo electrónico convencional se verían comprometidos.

Todo esto requeriría un almacenamiento indiscriminado de datos personales, que, aunque hoy en día sean comunes en la mayoría de aplicaciones, presentan un debate ético de grandes proporciones [2].

Por todos estos motivos, se decidió prescindir de una base de datos en primera instancia, y así se tratará hasta el capítulo 8. Sin embargo, en el punto 8.8, en base a los resultados de las pruebas sí que se decide su uso y se incluye una pequeña base de datos en la aplicación final.

Capítulo 3. Aspectos Teóricos

Este capítulo es bastante amplio y contiene muchos temas que se relacionan con otros puntos de la documentación. Aquí explicaremos una serie de conceptos teóricos con la idea de que sirvan como base a los lectores y les permitan entender mejor los siguientes apartados. Se describirán temas como el funcionamiento del correo electrónico, el uso de la librería JavaMail o las aplicaciones de una sola página.

Muchos de estos aspectos lidian con productos o herramientas específicas, de los que se indicará la justificación de usar cada uno de ellos, aunque los más importantes ya se han valorado en la Evaluación de Alternativas.

No se incluirá en este capítulo la descripción de los lenguajes o entornos de desarrollo usados, cuya explicación está contenida en el Capítulo 7 (Implementación del Sistema). Las herramientas de menor calado, con un uso cotidiano, se incluirán allí también, limitando este apartado a los conceptos más importantes, que son base del proyecto.

3.1 Correo Electrónico

A lo largo de todo este documento habrá múltiples referencias al correo electrónico, no en vano, la aplicación que se desarrollará tiene una fortísima dependencia de él. Es importante tener claros algunos conceptos sobre esta tecnología para entender bien la aplicación que se plantea en las próximas líneas. Aunque tiene un largo recorrido en el mundo de la informática y estamos muy familiarizados con él y con su uso cotidiano a través de clientes gráficos, no todo el mundo conoce lo que sucede por debajo, precisamente por esa larga historia que tiene ya hay herramientas suficientemente perfeccionadas que nos permiten abstraernos de todos los aspectos técnicos. Sin embargo, de cara a desarrollar un producto de estas características es necesario conocer, al menos, los siguientes conceptos.

Desde los años sesenta, el correo electrónico se ha venido usando como método para el intercambio de mensajes. Hoy en día se basa en un sistema en red que permite el envío de mensajes por medio de varios actores (clientes y servidores) que usan una serie de protocolos. El gráfico de la Figura 1 resume la arquitectura del correo electrónico.

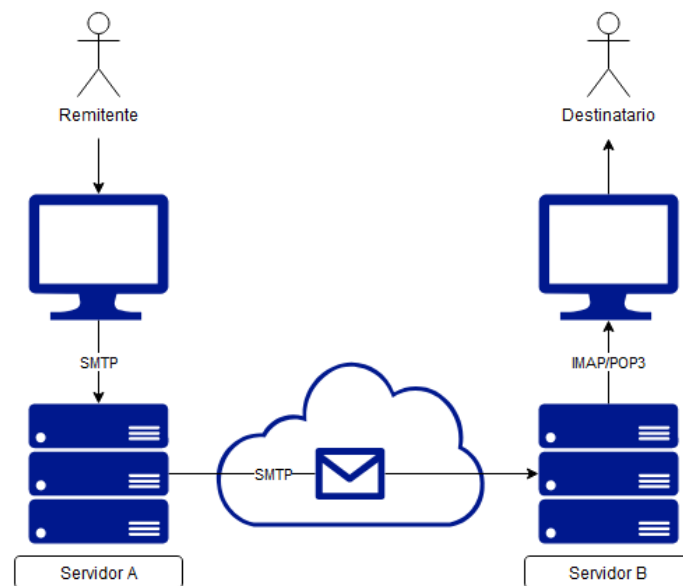


Figura 1 – Arquitectura del correo electrónico

El remitente es la persona que envía el correo. Este usuario, usa un cliente web, por ejemplo, la página de Gmail¹, para crear el correo electrónico que desea enviar. Por medio del protocolo SMTP (Simple Mail Transfer Protocol) el correo creado se envía al servidor del proveedor del usuario (Servidor A). A través de internet y usando el mismo protocolo (SMTP), el correo se transfiere al Servidor B, que es el del proveedor del destinatario. Esta transferencia se hace de la manera habitual en internet, por medio de la resolución de nombres usando servidores DNS y vía TCP/IP. Cuando el destinatario se conecta, por ejemplo, a través de la aplicación de escritorio de Outlook, descargará su correo desde servidor de su proveedor, usando el protocolo IMAP o POP3, que tienen sustanciales diferencias. La aplicación que se desarrolle en este proyecto usará estos protocolos para conectarse al servidor del destinatario (Servidor B).

El acceso a los servidores para la descarga de correos necesita varios parámetros, principalmente cuatro:

- Las credenciales (dirección de correo y contraseña) del usuario.
- El host, es decir, la dirección del servidor de correo.
- El puerto en el servidor de correo al que nos queremos conectar. Por defecto, IMAP usa los puertos 143 y 993 y POP3 usa los puertos 110 y 995.
- El protocolo a utilizar.

Un punto importante y que siempre se recomienda encarecidamente es el uso de una conexión cifrada con el servidor. Generalmente, para ello se usan los puertos 993 o 995 en función del protocolo.

¹ <https://mail.google.com/>

Los protocolos POP3 e IMAP permiten la descarga de correos del servidor. La principal diferencia entre ambos es que POP3 descarga los correos del servidor y los almacena en local, mientras que utilizando IMAP se almacenan en el servidor. Este segundo protocolo, tiene la ventaja de que todas las acciones que se realicen desde diferentes clientes estarán sincronizadas. Por su parte, POP3 presenta la ventaja de que los correos no ocupan espacio en el servidor [3]

En el servidor, los correos están organizados en carpetas. Por ejemplo, la carpeta "INBOX" contiene los correos recibidos por el usuario. También puede haber otra para correos enviados, para spam...

Por último, está bien reseñar la estructura de un correo electrónico. Están compuestos de unas cabeceras y un cuerpo. Las cabeceras permiten especificar datos como el remitente o el destinatario. Algunas de las más importantes y que usaremos en este proyecto son [4]:

- **From:** Especifica el remitente del correo.
- **To:** Especifica el destinatario del correo.
- **CC:** Especifica quién recibe una copia de carbón del correo.
- **Message-ID:** Es un identificador único del correo, debe ser diferente para cada correo.
- **In-Reply-To:** Indica el identificador el correo ("Message-ID") del que este correo es respuesta.

El cuerpo está compuesto de diferentes partes, y cada una puede tener un tipo MIME diferente. Por ejemplo, un correo que enviemos con un PDF adjunto, tendrá dos partes, un texto de tipo HTML (text/html) será una y un archivo PDF (application/pdf) otra.

3.2 JavaMail

Como comentábamos anteriormente, la aplicación se conectará al servidor del destinatario para obtener los correos. Los diferentes proveedores proporcionan APIs específicas para la descarga, sin embargo, nosotros queremos que la aplicación llegue al mayor público posible. Para ello, buscamos usar una librería genérica, que nos permita conectarnos a todo tipo de proveedores. En el mundo Java, una de las librerías esenciales para la gestión de correos es JavaMail² [5] y es la que se va a usar.

JavaMail nos permite conectarnos a cualquier proveedor especificando los datos antes enumerados (credenciales, host, puerto y protocolo). Una vez conectados, nos proporciona una interfaz para acceder a las carpetas y una vez en ellas a sus correos. La Figura 2 explica el proceso que debe seguirse.

El primer paso es obtener un objeto `Session` especificando los datos de la conexión. Usando ese objeto, podremos obtener el `Store` correspondiente al protocolo que usamos. Ese

² <https://javaee.github.io/javamail/>

`Store` contiene las carpetas, a las que debemos acceder especificando su nombre. También contiene un método que nos permite obtener los nombres de todas las carpetas almacenadas. Por último, una vez que tenemos el objeto `Folder`, que representa una carpeta, podemos acceder a los correos allí almacenados. El acceso se hace utilizando un número identificador, siendo el 1 el correo más antiguo y el número más alto el correo más reciente. Para estos cálculos, el objeto `Folder` también nos indica el número total de correos en la carpeta.

Por último, el objeto `Message`, representa un correo y nos permite acceder a los datos propios de un correo electrónico, por medio de métodos proporcionados al efecto. Podremos acceder a sus cabeceras y a las diferentes partes de su cuerpo.

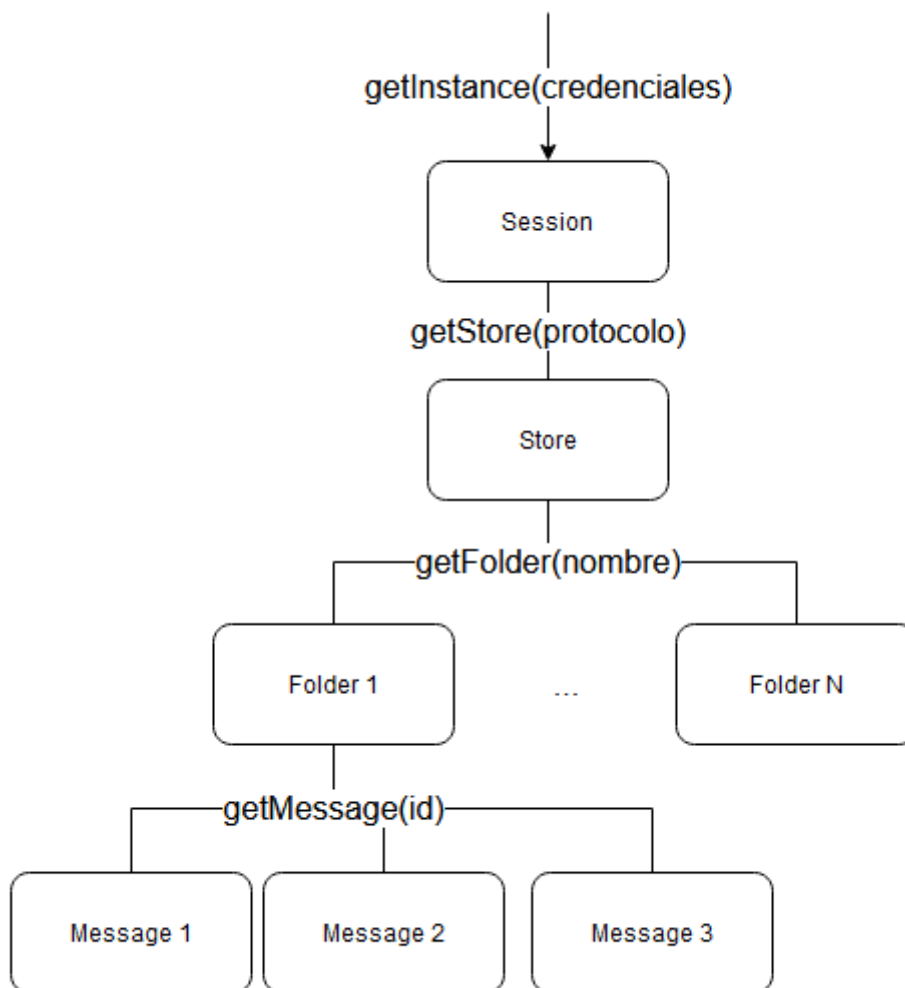


Figura 2 – Acceso a los correos usando JavaMail

3.3 SpringBoot

Una vez sentadas las bases del correo electrónico, es importante apuntar ciertas nociones de las tecnologías que vamos a usar. En este apartado comentaremos los aspectos más

importantes de SpringBoot, que será la que usemos para vertebrar la API de la aplicación, y cuyo uso hemos justificado en el punto 2.3.2.3 - Elección del *back-end*.

Esta tecnología simplifica la creación de aplicaciones basadas en Spring, de manera que realizando unas pocas configuraciones se pueda tener en funcionamiento la aplicación [6]. Por su parte, Spring es un *framework* para el desarrollo de aplicaciones, que se caracteriza por la aplicación de conceptos como la inyección de dependencias, el uso del patrón MVC (Modelo-vista-controlador) y la facilidad para incluir otros módulos como el mapeo objeto-relacional, la inclusión de módulos de seguridad que faciliten la autenticación y la autorización o herramientas de *testing*.

SpringBoot aprovecha todo esto y facilita su uso, utilizando el concepto de convención antes que configuración. Por ejemplo, usando Spring habrá que configurar el puerto donde funcionará la aplicación, sin embargo, usando SpringBoot se establecerá el puerto 8080 por defecto, válido en la mayoría de los casos, y que se podrá sobrescribir si se desea establecer otro de manera explícita.

Como comentábamos antes, Spring usa varios conceptos que lo caracterizan, algunos de los más interesantes son los siguientes.

3.3.1 Inversión de control

Tradicionalmente, la programación se hace de forma imperativa, es decir, por medio de una serie de llamadas a métodos, que ejecutan diferentes acciones cuando son activados. Cuando usamos el concepto de Inversión de Control (IoC) especificamos las respuestas deseadas, es decir, los objetos que queremos obtener, siendo otra entidad la que se encargue de realizar las acciones que se requieran, en el orden necesario, para obtenerlos [7].

En Spring esto se hace por medio de la Inyección de Dependencias. Esto es un patrón de diseño donde se suministran objetos a una clase, en lugar de ser la propia clase la que los cree [8]. El propio *framework* se encarga de esto, el desarrollador crea, por ejemplo, un servicio y ese servicio puede inyectarlo en otros componentes siendo el *framework* quien se encargue de coordinar ambas clases.

3.3.2 Patrón MVC

Spring usa este patrón para la separación de conceptos. Se basa en tres partes, por un lado está el modelo, que implementa la lógica de negocio de la aplicación, por otro, la vista, que presenta al usuario la información y por último, el controlador, que se encarga de invocar al modelo en función de las acciones que desee realizar el usuario. En nuestra aplicación el modelo lo compondrán las entidades, servicios y demás clases que procesan la información de los correos, los controladores mantendrán este nombre y se encargarán de la gestión de las peticiones y las vistas serán teóricamente los objetos JSON devueltos en las peticiones.

3.4 Maven

Antes de pasar a la parte del *front-end*, no está de más comentar algunos de los aspectos más importantes de Maven, la herramienta que usaremos para la construcción del proyecto. Se basa en el uso de un archivo POM (Project Object Model), donde se define el proyecto y sus dependencias, por ejemplo, diferentes librerías externas incluidas en su repositorio central, como JavaMail [9]. Una vez definidas, Maven realiza varias acciones de su ciclo de vida para construir el proyecto. Primero compila el código fuente, después ejecuta los test automáticos que se hayan definido, empaqueta los archivos binarios en un JAR, copia el paquete generado a una carpeta local y finalmente, copia el fichero en el servidor remoto donde se ejecute nuestro proyecto.

3.5 Angular

Como bien especificamos en el punto 2.3.2.4, Angular es la herramienta elegida para el desarrollo del *front-end*. Este *framework* para el desarrollo de aplicaciones web permite la creación de webs SPA (Single Page Application) por medio también del uso del patrón MVC, explicado previamente.

Una SPA es una aplicación que rompe con el modelo tradicional en el que los usuarios van navegando entre las diferentes páginas de una web. Por ejemplo, cuando se clica en un enlace se hace una petición al servidor que devuelve otra página diferente. Este proceso es temporalmente costoso, pues se necesita cierto tiempo para transferir todo el contenido, que además muchas veces está compuesto por elementos repetidos, como, por ejemplo, las cabeceras y los pies de la página. Las SPA evitan este proceso cargando sólo una página, que posteriormente va modificando dinámicamente sus componentes según el usuario realiza distintas acciones.

Angular 2 proporciona varias ventajas sobre otras herramientas más sencillas, como podría ser un gestor de plantillas. Una de las principales es la división y la reutilización de código. Por un lado, se hace la división entre servicios y componentes, sirviendo los primeros para la lógica de negocio (en nuestro caso principalmente para realizar llamada al *back-end*) y los segundos para el manejo de los eventos del usuario y todos los aspectos de presentación. De esta manera, no se incluye código que busca simplemente enriquecer la experiencia de usuario junto a elementos del modelo [10].

Cada componente de Angular está compuesto de cuatro partes: un archivo HTML con los elementos que se mostrarán en la interfaz, un archivo CSS para especificar los estilos y dos archivos Typescript, uno con el código fuente y otro especificando las pruebas del componente. Hay que remarcar que el archivo HTML no sólo puede incluir los elementos de HTML5, sino que también pueden anidarse otros componentes, que pueden ser parametrizados y generar eventos, siendo esto clave para la reutilización de código. Por supuesto, las variables del código fuente, pueden mostrarse en el HTML y este puede llamar a funciones del código en respuesta a eventos de usuario, por ejemplo. Como puede verse, las posibilidades que el *framework* ofrece son amplias.

3.6 Docker

Otro de los conceptos con el que más se va a trabajar en las próximas líneas, sobre todo en la parte de despliegue es el de Docker. Tradicionalmente cuando se desarrollaba una página web había que realizar una serie de tareas de configuración, instalación y despliegue. En el servidor en el que quisiéramos desplegar debíamos instalar todas las tecnologías necesarias, por ejemplo, un balanceador de carga, un servidor web, una máquina virtual de Java, una base de datos... todo ello con los consiguientes problemas de versiones, de instalación, de configuración, en definitiva, una labor muy propensa a errores. Para evitar esto, y además pensando en facilitar la tarea a los profesores que evalúen este TFG usaremos contenedores de Docker para desplegar nuestra aplicación.

Para tener una idea del funcionamiento de Docker hay que tener claros algunos conceptos que vamos a ir explicando en las próximas líneas tales como los contenedores, las imágenes o Docker Compose.

El concepto de contenedor es similar al de máquina virtual, aunque más sencillo, la Figura 3 pretende mostrar las diferencias y similitudes entre ambos. El uso de máquinas virtuales requiere de un Software de Virtualización, por ejemplo, Virtualbox, donde se crearán “equipos virtuales” que funcionarán como equipos reales, es decir, tendrán su propio sistema operativo y sobre él se instalarán los diferentes servicios que vayamos a utilizar. Por otro lado, usando Docker, instalaremos este software y en él crearemos diferentes contenedores, más ligeros y muy optimizados, cada uno con una “imagen”, esto es, un servicio configurado de manera estándar. Las imágenes para las diferentes tecnologías se pueden descargar del Docker Hub, un repositorio centralizado. Como vemos, cada contenedor contendrá un servicio, por ejemplo, una base de datos MySQL y un Servidor Apache. Para coordinar estos contenedores, una herramienta muy útil es Docker Compose, que nos permitirá definir y configurar varios en un solo archivo.

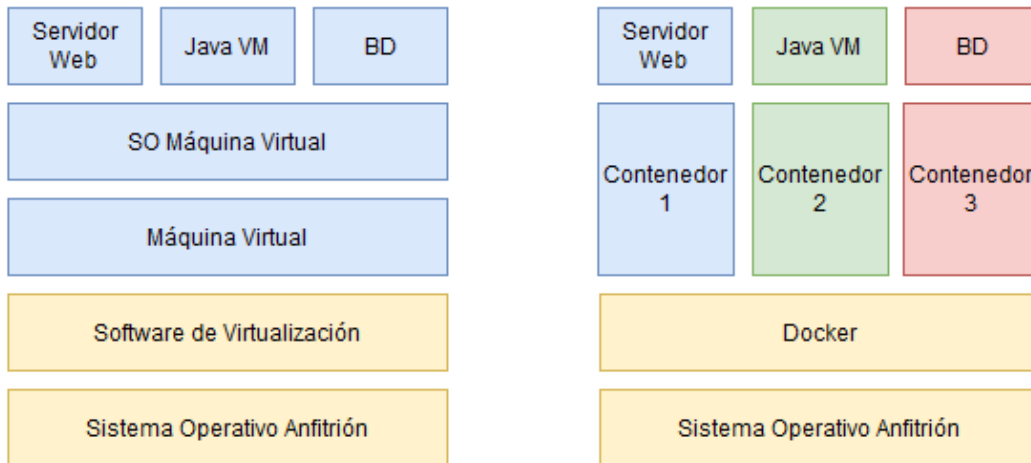


Figura 3 – Comparación entre máquinas virtuales y Docker

El uso de Docker presenta varias ventajas. Como se puede deducir de la explicación anterior permite la separación de responsabilidades, facilita la instalación de diferentes softwares, evita largos periodos de configuración, ahorra elevados tiempos de espera en el proceso de instalación, ocupa menos espacio que otras alternativas y en nuestro caso especialmente, facilita el despliegue de cara a las pruebas de este TFG eliminando posibles errores de compatibilidad al desplegar la aplicación en diferentes equipos.

Capítulo 4. Planificación del Proyecto y Presupuesto Iniciales

En este punto del proyecto vamos a plantear cómo se va a ejecutar este proyecto, tanto a través de su planificación temporal, como del presupuesto inicial adaptándonos a las particularidades de un TFG y realizando todos los planteamientos que sean necesarios.

4.1 Planificación Inicial

Como hemos desgranado en capítulos anteriores, el objetivo principal de este proyecto es el desarrollo de un software que dé solución al problema que se plantea. En nuestra área de conocimiento, tenemos diferentes metodologías entre las que escoger a la hora de planificarlo. Pero dentro de este proyecto, no solo vamos a desarrollar un software, también debemos crear esta documentación, con una serie de puntos que consideramos importantes y se desmarcan un poco de lo que es un proyecto típicamente empresarial. Para afrontar esto hemos considerado dividir la ejecución del proyecto en dos partes. Por un lado, agrupamos lo que es el análisis, la construcción, el diseño de las pruebas, su ejecución y la documentación del software, en lo que llamaremos desarrollo. Por otro, estará la redacción, maquetación y entrega de la documentación y otros entregables del proyecto.

Siendo esto así, nos encontramos ante dos partes que afrontaremos de diferente modo. El desarrollo será por medio de una metodología ágil, en este caso Scrum, por concurrir aquí muchas de las situaciones que recomiendan su uso. Por ejemplo, no estar ceñidos a una serie de requisitos iniciales precisos y concretos. Sin embargo, la otra parte, es mucho más sencilla de realizar con una aproximación clásica, pues consistirá simplemente en redactar, maquetar y ordenar aquellos diagramas, historias y requisitos que ya hemos creado y utilizado en el desarrollo.

A continuación, iremos desgranando en detalle la planificación inicial del proyecto, siguiendo esta aproximación dual.

Pero, antes de nada, hay que definir el calendario. Se trabajarán 3 horas al día, de lunes a viernes y empezando el día 6 de junio de 2021. A esto hay que descontarle dos semanas de vacaciones, entre el 19 y el 30 de julio.

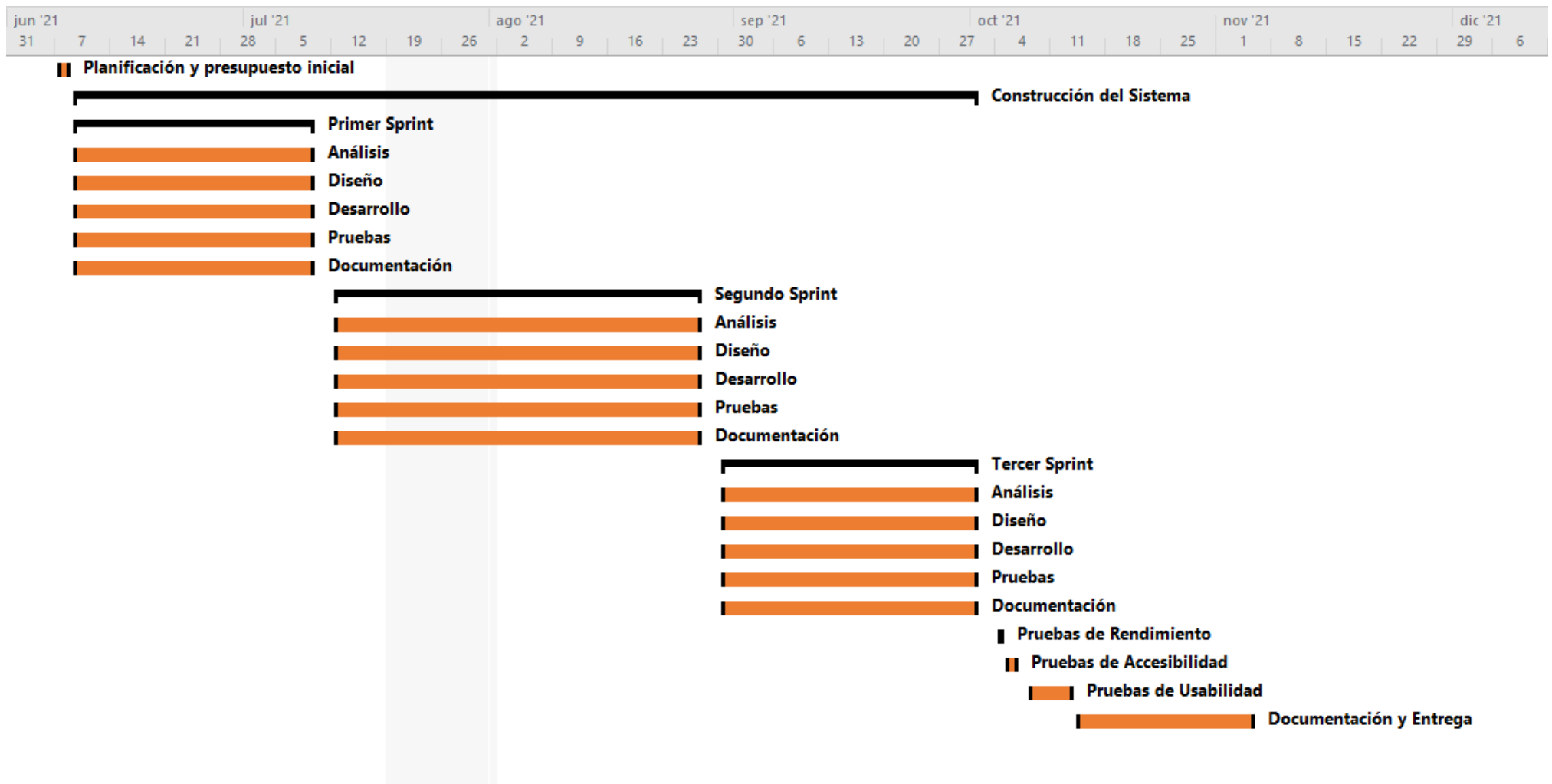


Figura 4 – Planificación Inicial del Proyecto

Ahora sí, vamos a ver la planificación. El Diagrama de Gantt de la Figura 4 muestra el orden de las tareas. Aquí se muestran las tareas con las dos partes bien claras. La tarea resumen que aquí se llama “Construcción del Sistema” está dividida en *sprints* y es la parte que realizaremos usando Scrum, entraremos posteriormente en detalle con esto.

Junto a la tarea que comentamos en el punto anterior, podemos ver que la planificación empieza con la redacción de este punto y del presupuesto que sigue. Una vez planteado, desarrollaremos el software. Tras ello, se realizarán las pruebas de accesibilidad y usabilidad, que, si bien se deberían hacer en cada *sprint*, por falta de recursos se realizarán al final de la construcción. Una vez validado el software pasaremos a redactar los puntos de la documentación que resten.

Este tema de la documentación se hará al final, pero no por ello estamos planteando redactar el análisis o el diseño una vez hecho el código. Los diferentes artefactos (historias de usuario, requisitos, diagramas...) se irán utilizando durante la fase de construcción, pero no se plasmarán en el documento final, si no que cada uno se mantendrá en las diferentes herramientas que se usen. Un gestor de tarjetas para las historias, otro tipo de herramientas para la gestión de los requisitos o un software que facilite la edición de diagramas pueden ser algunos ejemplos. Al final del desarrollo, se plasmará aquí el producto final, es decir, la lista completa de requisitos, los diagramas finalmente construidos, todas las historias ya refinadas... De esta manera se agilizará el uso de estos artefactos sin tener que pararse a maquetar elementos temporales que posteriormente serán modificados. Para todas las tareas de este párrafo y del anterior se ha estimado una duración de 81 horas, en función de la experiencia anterior usando la técnica conocida como “Juicio de Expertos”.

Ahora, entraremos en detalle en la parte de la construcción del software. Como se puede ver en el diagrama tendremos tres *sprints*. Para esta fase tendremos las 221 horas que restan, las de los días laborables entre el 9 de junio y el 1 de octubre. Los *sprints* serán de 75 horas (excepto el primero, que para coincidir con semanas completas hemos reducido a 69 horas). Esto son 25 días de trabajo. Esto sería muchísimo en un desarrollo normal, pero las características del TFG, con un único programador, nos obliga a alargar el tiempo del *sprint*, siendo aquí de 5 semanas, pero equivaliendo a 2 semanas en un equipo con jornada de 8 horas.

Esos tres *sprints* tendrán las fechas que se muestran en la Tabla 1:

	Comienzo	Fin	Horas
Sprint 1	9 de junio de 2021	9 de julio de 2021	69
Sprint 2	12 de julio 2021	27 de agosto de 2021	75
Sprint 3	30 de agosto 2021	1 de octubre de 2021	75

Tabla 1 – Fechas y duración de los ‘sprints’

En el diagrama, dentro de los *sprints*, se incluyen cinco tareas simultáneas: análisis, diseño, desarrollo, pruebas y documentación. Esto no quiere decir que se hagan todas a la vez. Se

realizará el ciclo completo por cada historia de usuario, cuya duración aún no tenemos estimada. Vamos a analizarlas una a una:

- **Análisis:** Contendrá toda la parte relativa a la gestión de requisitos, los casos de uso y el resto de tareas que se pueden ver en el Capítulo 5. Será el primer paso a la hora de afrontar una historia de usuario. También podemos considerar Análisis al refinamiento de la pila del producto y a su estimación, aunque no se haga de manera cronológica en una historia, si no en los momentos oportunos del *sprint*. Incluimos también aquí las tareas que se realizan al comienzo y al final del *sprint* para su planteamiento y para la reflexión acerca del trabajo realizado, respectivamente.
- **Diseño:** Dentro de esta fase se partirá de lo visto en el análisis y se diseñarán los cambios que se van a realizar. También se crean aquí las pruebas.
- **Desarrollo:** Se refiere a la implementación del código.
- **Pruebas:** Se realizarán varios tipos de prueba en esta fase. Aunque luego se detallará el calendario de pruebas, dentro de un *sprint* tendremos la ejecución de las pruebas unitarias, de integración y funcionales, que se irán manteniendo e incrementando a lo largo de todo el proyecto. Además, habrá unas pruebas específicas para cada historia de usuario.
- **Documentación:** Se refiere a la documentación del código y en general a dejar constancia escrita de aquellas cosas que pueden ser útiles en el futuro, por ejemplo, los problemas encontrados para el punto 7.4.1.

Dentro de esos *sprints* desarrollaremos una serie de funcionalidades. En este punto no podemos enumerarlas, pues es una tarea incremental. Sin embargo, si podemos definir una pila inicial de tareas, que enumeraremos a continuación. Aunque seguramente surjan más que añadir, al final de cada *sprint*. Por ahora serán EPICs (es decir, historias de usuario de muy alto nivel). La aplicación tiene que permitir al usuario autenticarse con su proveedor de correo electrónico.

- La aplicación debe permitir recuperar los correos recibidos y enviados por el usuario.
- La aplicación debe permitir generar archivos que contengan los correos que el usuario seleccione.
- La aplicación debe permitir el acceso con varios proveedores de correo electrónico.
- La aplicación debe estar internacionalizada a los idiomas asturiano, castellano e inglés y también las actas generadas.
- La aplicación debe tener unos logos propios que mejoren su identificación y apariencia.
- La aplicación debe mantener un log con las acciones realizadas sobre el sistema.
- La aplicación deberá poder desplegarse usando un contenedor de Docker.

Por último, está bien definir una planificación temporal de las pruebas. En función del tipo y para un mejor entendimiento de los puntos relacionados con ellas (empezando por el 5.8), se incluye este calendario de pruebas. Es un resumen, si se desea una explicación más avanzada pueden verse los apartados relacionados.

N.º	Tipo	Área bajo prueba	Diseño	Mantenimiento	Ejecución
1	Pruebas Unitarias	Un componente	En la fase de diseño de cada historia.	Incremental	Como parte del desarrollo de la historia.
2	Pruebas de Integración	Múltiples componentes	En la fase de diseño de cada historia.	Incremental	Como parte del desarrollo de la historia.
3	Pruebas de las Historias de Usuario	Una funcionalidad	En la fase de diseño de cada historia.	Cada historia tiene una lista de pruebas	De manera manual, una vez desarrollada la historia, antes de darla por completada.
4	Pruebas Funcionales	Aplicación completa	En el diseño del <i>sprint</i> .	Incremental	Al final del <i>sprint</i> .
5	Pruebas de Rendimiento	Funcionalidades que lo requieran	Sobre el producto final.	Una ejecución	Al final del desarrollo.
6	Pruebas de Accesibilidad	Aplicación completa	Cuestionario estandarizado.	Una ejecución	Al final del desarrollo.
7	Pruebas de Usabilidad	Aplicación completa	Sobre el producto final.	Una ejecución	Al final del desarrollo.

Tabla 2 – Calendario de Pruebas

4.2 Presupuesto Inicial

En este punto, plantearemos un presupuesto inicial para el proyecto. Este capítulo estará claramente dividido en dos partes. Primero calcularemos el presupuesto general de una empresa ficticia, a lo largo de un año. Una vez obtenido el coste real de cada empleado pasaremos al segundo punto, detallar el presupuesto de este proyecto en el tiempo correspondiente.

4.2.1 Presupuesto Anual de la Empresa

Empezaremos planteando la empresa, consideraremos una pequeña *startup*, con tres trabajadores. Estos empleados tendrán tres perfiles diferentes: un desarrollador, un *tester* y un Jefe de Negocio. La empresa contará con una oficina donde se realicen los trabajos.

4.2.1.1 Costes Indirectos

Vamos a desgarnar primero los costes indirectos relativos a los servicios de la oficina. Se han usado los precios medios anuales obtenidos de diferentes estadísticas, precios de mercado y la experiencia propia.

Servicios	Coste/año
Agua	140.64 €
Electricidad	703.80 €
Limpieza	720.00 €
Catering	1,728.00 €
Mantenimiento	300.00 €
Material de Oficina	100.00 €
Calefacción	1,400.00 €
Internet	358.80 €
TOTAL	4,299.24 €

Tabla 3 – Costes anuales de los servicios de la empresa

4.2.1.2 Costes de Equipos/Licencias

También debemos considerar los costes de los equipos que utilizan los empleados en el día a día, son tres empleados, por tanto, tres equipos. Como puede verse, son pequeños, pues no hace falta gran cantidad de material.

Equipo	Cantidad	Coste unitario	Años amortización	Coste/año
Ordenador	3	800.00 €	6	400.00 €
Pantalla	3	200.00 €	10	60.00 €
Periféricos	3	100.00 €	5	60.00 €
TOTAL				520.00 €

Tabla 4 – Costes de los equipos de la empresa

Consideramos también las licencias de software necesarias para el desarrollo de la actividad empresarial. Las licencias del sistema operativo y de Microsoft Office, valen para un solo equipo, por tanto, han de amortizarse en los mismos años que se estiman como periodo de amortización de este. La licencia de JetBrains ha de renovarse anualmente.

Licencias	Cantidad	Coste	Años	Coste/año
SO Microsoft Windows	3	439.00 €	6	219.50 €
Paquete Microsoft Office	3	579.00 €	6	289.50 €
JetBrains (Todos)	3	389.00 €	1	1,167.00 €
TOTAL				1,676.00 €

Tabla 5 – Costes de las licencias de la empresa

4.2.1.3 Costes de Personal

Como hemos dicho, esta empresa cuenta con tres trabajadores, y para calcular el gasto que nos suponen al año mostraremos una serie de cifras.

Dato	Desarrollador	Tester	Jefe de Negocio
Sueldo bruto	20 €/h	20 €/h	25 €/h

Horas/año/persona³	860	860	860
Sueldo bruto/año	17.200 €	17.200 €	21.500 €
Coste Salarial Año	22.600,80 €	22.600,80 €	28.251,00 €
Productividad⁴	90%	80%	20%
Coste Directo	20.340,72 €	18.080,64 €	5.650,20 €
Coste Indirecto	2.260,08 €	4.520,16 €	22.600,80 €

Tabla 6 – Cálculo de los costes salariales de los empleados de la empresa

Hemos partido de un sueldo medio obtenido de diferentes páginas de empleo. Posteriormente obtenemos el sueldo al año y le sumamos un 31%, correspondiente a las tarifas de cotización marcadas por ley⁵. En esta tabla cogemos el sueldo bruto de los diferentes puestos y obtenemos el coste salarial anual. Ese coste se divide en función de la productividad en costes directos o indirectos.

4.2.1.4 Resumen

Los cálculos anteriores se resumen en esta tabla, donde mostramos la suma de los costes directos e indirectos calculados anteriormente. Tomamos un 25% del total como beneficios deseados.

Costes Directos	44,071.56 €
Costes Indirectos	37,028.28 €
Beneficios (25%)	20,274.96 €
TOTAL	101,374.80 €

Tabla 7 – Costes y beneficios anuales de la empresa divididos por tipo

De estos cálculos sacamos que la empresa ha de obtener unos ingresos totales de 101,374.80 € al año.

Con estos ingresos deseados, podemos ver cuánto nos costaría cada puesto del personal al año (con y sin beneficios). Este valor será el que se use posteriormente en los presupuestos del proyecto, tanto en el de costes para la empresa (sin beneficios) como para el cliente (con beneficios).

Dato	Desarrollador	Tester	Jefe de Negocio	TOTAL
Sueldo bruto (hora)	20 €	20 €	25 €	
Precio / hora (final)	62 €	62 €	77,50 €	
Facturación deseada (anual)	47.988,00 €	42.656,00 €	13.330,00 €	103.974,00 €
Precio / hora (sin beneficios)	49,60 €	49,60 €	62,00 €	

³ Se usa la mitad de las horas laborables en el año 2021 ya que en esta empresa se trabaja 4h/día.

⁴ Se toman unos valores aproximados que determinan el porcentaje de tiempo que dedica a un proyecto, esto son costes directos. La cantidad restante son costes indirectos y se corresponden con el tiempo dedicado a labores de gestión, formación...

⁵ <https://cdn.cuestioneslaborales.es/wp-content/uploads/2020/10/Tarifas-primas-AT-y-EP.pdf>

Tabla 8 – Resumen de los costes salariales del personal

Se han redondeado los precios/hora de los diferentes puestos, dejando un margen de un 2,56 % entre el total necesario y la facturación deseada que calculamos.

En definitiva, para los siguientes apartados consideraremos los siguientes precios:

Rol	Precio/hora (sin beneficios)	Precio/hora (con beneficios)
Desarrollador	49,60 €	62,00 €
Tester	49,60 €	62,00 €
Jefe de Negocio	62,00 €	77,50 €

Tabla 9 – Resumen de costes finales de los empleados.

4.2.2 Presupuesto del Proyecto

Hasta aquí hemos calculado todo lo relativo a nuestra empresa. Ahora, en función del coste de cada empleado obtenido anteriormente pasaremos a detallar los presupuestos del proyecto. Primero habrá un presupuesto de costes, que usará los precios sin beneficios y determinará el coste real del proyecto para la empresa. Posteriormente mostraremos el presupuesto del cliente, que sería el que se le entregaría y que incluye los beneficios.

4.2.2.1 Presupuesto de Costes

En base a la planificación sacamos lo que nos costaría realizar cada uno de los *items*, definiendo las horas de trabajo necesario para cada uno de ellos y el precio hora obtenido en el punto anterior. En función de la tarea se ha asignado un recurso u otro. Los temas de gestión como la planificación y el presupuesto es tarea del jefe de negocio, las pruebas del *tester*, y la documentación técnica es tarea del desarrollador. En los *sprints* trabajan los tres puestos, aunque la dedicación del jefe de negocio es un tercio de la del resto.

Part.	Item	Recurso	Horas	Precio/h	Total Item	
1	Construcción del Sistema				26,250.80 €	
	1	Primer Sprint				
		1	Jefe de Negocio	23	62.00 €	1,426.00 €
		2	Desarrollador	69	49.60 €	3,422.40 €
		3	Tester	69	49.60 €	3,422.40 €
	2	Segundo Sprint				
		1	Jefe de Negocio	25	62.00 €	1,550.00 €
		2	Desarrollador	75	49.60 €	3,720.00 €
		3	Tester	75	49.60 €	3,720.00 €
	3	Tercer Sprint				
		1	Jefe de Negocio	25	62.00 €	1,550.00 €
		2	Desarrollador	75	49.60 €	3,720.00 €
		3	Tester	75	49.60 €	3,720.00 €

2		Documentación			2,976.00 €
1		Planificación y presupuesto			
	1	<i>Jefe de Negocio</i>	6	62.00 €	297.60 €
2		Redactar documentación			
	1	<i>Desarrollador</i>	54	49.60 €	2,678.40 €
3		Pruebas del sistema			1,041.60 €
1		Pruebas de accesibilidad			
	1	<i>Tester</i>	9	49.60 €	446.40 €
2		Pruebas de usabilidad			
	1	<i>Tester</i>	12	49.60 €	595.20 €
		TOTAL			30,268.40 €

Tabla 10 – Presupuesto de Costes del Proyecto

4.2.2.2 Presupuesto del Cliente

Aquí detallaremos los diferentes *items* como se mostrarían al cliente. Es decir, sumándole el 25% de beneficio a cada uno de los del presupuesto de costes. Si tuviésemos materiales entregables (Ej. hardware) el valor aumentaría, pero aquí no es necesario.

Part.	Item	Recurso	Horas	Precio/h	Total Item
1		Construcción del Sistema			32,813.50 €
	1	Primer Sprint			
		1 <i>Jefe de Negocio</i>	23	77.50 €	1,782.50 €
		2 <i>Desarrollador</i>	69	62.00 €	4,278.00 €
		3 <i>Tester</i>	69	62.00 €	4,278.00 €
	2	Segundo Sprint			
		1 <i>Jefe de Negocio</i>	25	77.50 €	1,937.50 €
		2 <i>Desarrollador</i>	75	62.00 €	4,650.00 €
		3 <i>Tester</i>	75	62.00 €	4,650.00 €
	3	Tercer Sprint			
		1 <i>Jefe de Negocio</i>	25	77.50 €	1,937.50 €
		2 <i>Desarrollador</i>	75	62.00 €	4,650.00 €
		3 <i>Tester</i>	75	62.00 €	4,650.00 €
2		Documentación			3,720.00 €
	1	Planificación y presupuesto			
		1 <i>Jefe de Negocio</i>	6	77.50 €	372.00 €
	2	Redactar documentación			
		1 <i>Desarrollador</i>	54	62.00 €	3,348.00 €
3		Pruebas del sistema			1,302.00 €
	1	Pruebas de accesibilidad			
		1 <i>Tester</i>	9	62.00 €	558.00 €

2	Pruebas de usabilidad			
1	Tester	12	62.00 €	744.00 €
			TOTAL	37,835.50 €

Tabla 11 – Presupuesto de Cliente del Proyecto

Y resumiendo aún más las partidas, los costes que se le aplicarían al cliente serían estos:

Dato	Precio
Construcción del Sistema	32,813.50 €
Documentación	3,720.00 €
Pruebas del sistema	1,302.00 €
TOTAL	37,835.50 €

Tabla 12 – Resumen por partidas del presupuesto de cliente

Capítulo 5. Análisis

En este apartado incluiremos toda aquella documentación relativa al análisis de la aplicación. Entre otras cosas se hablará del alcance, de los requisitos, de los casos de uso y del plan de pruebas.

5.1 Definición del Sistema

En las próximas líneas delimitaremos a grandes rasgos aquello que se incluirá en el producto desarrollado. Pese a que sigamos la metodología Scrum, es importante fijar un alcance inicial para dejar claro lo que se quiere hacer, aunque luego pueda sufrir algunas modificaciones a lo largo de los diferentes *sprints*.

5.1.1 Determinación del Alcance del Sistema

Este producto constará de dos partes bien definidas. Una SPA (aplicación de una sola página) que actuará como *front-end* y una API (interfaz de programación de aplicaciones) que actuará como *back-end*. Deberán trabajar en conjunto, pues, aunque nosotros propongamos ese *front-end*, un usuario con habilidades de programador podría crear el suyo propio para responder a sus necesidades.

Independientemente de la interfaz usada, el sistema deberá proporcionar al usuario una serie de funcionalidades. La idea es que pueda conectarse a su cuenta de correo electrónico sea cual sea el proveedor que use. Esta conexión se hará de manera automática, aunque además el usuario debe poder especificar el servidor de manera manual.

Una vez conectado, deberá poder listar los correos recibidos (y los enviados) en la cuenta de correo especificada. Los mensajes podrán ser seleccionados para generar un documento con ellos, que llamaremos *acta*.

Esa *acta*, será un documento PDF que contendrá el cuerpo de los correos, así como algunos de sus metadatos (direcciones de emisor y receptor, fechas...) Además de esos datos, deberá poder añadirse manualmente cualquier información que estime oportuna (títulos, fecha del documento o cualquier otro campo personalizado...) y se le proporcionará la posibilidad de mostrar otra pre-programada a partir de los datos procesados de los correos. Un ejemplo de esto puede ser el incluir una enumeración de los participantes en la conversación o calcular las fechas de inicio y fin de la misma. Aunque con menos sofisticación, también se incluirá la exportación de los correos en otros formatos.

La aplicación estará internacionalizada, y localizada a tres idiomas (asturiano, castellano e inglés). Esto será aplicable tanto al *front-end* como a las *actas* generadas. Evidentemente estarán localizadas las etiquetas, formatos numéricos y fechas, no el contenido de los correos.

5.2 Historias de Usuario

Tras esta definición del alcance, es hora de precisar qué construir. Antes de pasar a obtener los requisitos directamente del cliente, tal y como se haría en un desarrollo clásico, aquí tenemos un paso intermedio que son las historias de usuario. En nuestro día a día trabajaremos principalmente con ellas, aunque sin olvidar los requisitos, como bien hablaremos luego.

Las historias de usuario son uno de los artefactos más típicos de Scrum. Estas se describen como “una explicación general e informal de una función de software escrita desde la perspectiva del usuario final” y su objetivo es “articular cómo proporcionará una función de software valor al cliente” [11].

Para el desarrollo de este sistema hemos usado varias y en las próximas líneas las iremos mostrando, divididas en función del *sprint* en que se realizan. Para todas ellas mostraremos la estimación prevista en horas. Han de tomarse como nuestra principal herramienta a la hora de definir el trabajo a realizar.

Las historias se agrupan bajo el paraguas de una EPIC (una historia con muy poco detalle que aún no ha sido refinada) y que representa una parte del sistema.

En las próximas líneas detallaremos las EPIC, que tendrán un identificador que posteriormente se indicará en las historias de usuario propias de cada *sprint*.

5.2.1 EPICs

Tenemos seis EPICs principales de las que partiremos, son las siguientes.

Como usuario quiero poder conectarme a mi correo electrónico para poder acceder a los correos enviados/recibidos

Identificador	CONEXIÓN
---------------	----------

Como usuario quiero poder ver los correos que he enviado/recibido para poder escoger cuáles quiero generar

Identificador	LISTADO
---------------	---------

Como usuario quiero poder generar un acta que contenga los correos que me interesen para poder usarlos en el futuro de acuerdo a los objetivos de la aplicación

Identificador	GENERACIÓN
---------------	------------

Como usuario programador quiero usar las funcionalidades de la aplicación vía API para tener la libertad de usar la interfaz que yo quiera

Identificador	API
---------------	-----

Como Jefe de Proyecto quiero que la aplicación esté desplegada de manera segura, eficiente

y mantenible para evitar sobrecostos futuros	
Identificador	OPS

Como Jefe de Proyecto quiero que la aplicación tenga una interfaz moderna y amigable para facilitar la labor de marketing	
Identificador	INTERFAZ

5.2.2 Historias del Primer Sprint

Aquí se incluyen las historias que se desarrollan en el primer *sprint*, comprendido entre los días 21 de junio y 9 de julio de 2021, que cuenta con un total de 15 días laborables y 45 horas de trabajo.

5.2.2.1 Historia H1

<i>Como usuario quiero poder conectarme a mi cuenta de correo para poder realizar operaciones posteriormente sobre ella.</i>	
Título Corto	Conexión con correo
Identificador	H1
EPIC	CONEXIÓN
Tiempo estimado (horas)	9
Tiempo necesario para completar (horas) ⁶	9
Criterios de aceptación	
<ul style="list-style-type: none"> • Al entrar en la aplicación el usuario debe poder especificar la dirección y la contraseña de su correo. • Al especificar la dirección y la contraseña, el sistema debe conectarse al servidor de correo de esa cuenta. • Las direcciones que debemos reconocer deben ser de Outlook o de Gmail. 	

5.2.2.2 Historia H2

<i>Como usuario quiero poder desconectarme de mi cuenta de correo para evitar que esta se bloquee.</i>
--

⁶ Aunque este dato se tomó con posterioridad a la fase de análisis se incluye aquí para facilitar la comparativa entre el tiempo previsto y el definitivo.

Título Corto	Desconectarse
Identificador	H2
EPIC	CONEXIÓN
Tiempo estimado (horas)	3
Tiempo necesario para completar (horas)	3
Criterios de aceptación	
<ul style="list-style-type: none"> Teniendo una conexión abierta debe poder cerrarse con una acción de usuario. 	

5.2.2.3 Historia H3

<i>Como usuario quiero poder ver el asunto de los correos que me han enviado para poder seleccionar de cuáles debo generar un informe</i>	
Título Corto	Listar correos
Identificador	H3
EPIC	LISTADO
Tiempo estimado (horas)	18
Tiempo necesario para completar (horas)	18
Criterios de aceptación	
<ul style="list-style-type: none"> Una vez abierta una conexión deben mostrársele al usuario los mensajes de su cuenta de correo. Para cada mensaje debe ver el asunto del mismo. 	

5.2.2.4 Historia H4

<i>Como usuario quiero poder añadir campos extra a los archivos generados.</i>	
Título Corto	Añadir campos extra
Identificador	H4
EPIC	GENERACIÓN

Tiempo estimado (horas)	6
Tiempo necesario para completar (horas)	5
Criterios de aceptación	
<ul style="list-style-type: none"> • El usuario debe poder añadir un título al acta. • El usuario debe poder añadir una fecha al acta, aunque se le debe facilitar poner la actual. 	

5.2.2.5 Historia H5

<i>Como usuario quiero poder generar un PDF con una serie de correos seleccionados para que me sirvan de acta de reunión</i>	
Título Corto	Generar PDF Simple
Identificador	H5
EPIC	GENERACIÓN
Tiempo estimado (horas)	9
Tiempo necesario para completar (horas)	10
Criterios de aceptación	
<ul style="list-style-type: none"> • El usuario debe poder elegir qué correos exportar. • El sistema debe generar un PDF que contenga esos correos, así como los campos extra (título y fecha). 	

5.2.3 Historias del Segundo Sprint

En el segundo *sprint* se incluyen las siguientes historias. Se desarrolla entre el 9 de agosto de 2021 y el 10 de septiembre, con 25 días de trabajo y, por tanto, 75 horas a invertir.

5.2.3.1 Historia H6

Como usuario quiero poder ver más información que el asunto, sobre los correos que voy a generar para saber de manera más eficiente que correos estoy viendo.

Título Corto	Previsualizar contenido correos
Identificador	H6
EPIC	LISTADO
Tiempo estimado (horas)	6
Tiempo necesario para completar (horas)	12
Criterios de aceptación	
<ul style="list-style-type: none"> • Deberá mostrarse el cuerpo, la fecha, remitente y destinatarios de los correos que aparecen en la lista de correos. 	

5.2.3.2 Historia H7

<i>Como usuario quiero poder añadir campos extra con la etiqueta y el valor que yo quiera para poder personalizar aún más las actas generadas.</i>	
Título Corto	Añadir campos extra personalizados
Identificador	H7
EPIC	GENERACIÓN
Tiempo estimado (horas)	12
Tiempo necesario para completar (horas)	6
Criterios de aceptación	
<ul style="list-style-type: none"> • Debe poder añadirse un número "ilimitado" de campos extra (etiqueta y valor) a las actas. • Deberá poder modificarse el nombre de los campos por defecto. • Deberán poder desactivarse los campos para que no se añadan al acta. 	

5.2.3.3 Historia H8

<i>Como Jefe de Proyecto quiero que se añadan logos a la aplicación para mejorar su apariencia y cumplir con las obligaciones.</i>	
Título Corto	Logos de la aplicación
Identificador	H8

EPIC	INTERFAZ
Tiempo estimado (horas)	9
Tiempo necesario para completar (horas)	3
Criterios de aceptación	
<ul style="list-style-type: none"> • Debe crearse un logo para la aplicación que se mostrará junto al título. • Debe crearse una versión simplificada del logo para usar como favicon. • Debe añadirse el logo de la Universidad de Oviedo para identificar el proyecto, así como el logo de la Escuela de Ingeniería Informática. 	

5.2.3.4 Historia H9

<i>Como usuario quiero que los correos carguen más rápido para poder tener una experiencia más fluida.</i>	
Título Corto	Cachear correos
Identificador	H9
EPIC	LISTADO
Tiempo estimado (horas)	12
Tiempo necesario para completar (horas)	9
Criterios de aceptación	
<ul style="list-style-type: none"> • No debe haber tiempos de espera superiores a medio segundo al cambiar entre las diferentes páginas del listado de correos, si estas ya se han cargado una vez en esta conexión. 	

5.2.3.5 Historia H10

<i>Como usuario quiero poder navegar entre páginas de manera no secuencial para evitar perder tiempo navegando por páginas que no me interesan.</i>	
Título Corto	Navegación por páginas no secuencial
Identificador	H10
EPIC	LISTADO

Tiempo estimado (horas)	6
Tiempo necesario para completar (horas)	4,5
Criterios de aceptación	
<ul style="list-style-type: none"> • Debe poder escribirse el número de la página a la que se quiere ir. • Debe poder navegarse a la última página con un sólo clic. • Debe poder navegarse a la primera página con un sólo clic. 	

5.2.3.6 Historia H11

<i>Como usuario quiero que mis sesiones sean independientes de las del resto de usuarios para evitar filtraciones de mis datos personales.</i>	
Título Corto	Independencia de sesiones
Identificador	H11
EPIC	API
Tiempo estimado (horas)	12
Tiempo necesario para completar (horas)	10,5
Criterios de aceptación	
<ul style="list-style-type: none"> • Ninguno de los datos almacenados en la sesión debe ser compartido con ninguno de los usuarios de otras sesiones. • Los correos cacheados deben ser independientes de unas a otras sesiones, así como las conexiones abiertas. 	

5.2.3.7 Historia H12

<i>Como usuario quiero ver la lista de correos en orden de fecha descendente para tener una experiencia más similar a un cliente de correo convencional.</i>	
Título Corto	Orden de correos
Identificador	H12
EPIC	LISTADO
Tiempo estimado (horas)	12

Tiempo necesario para completar (horas)	15
Criterios de aceptación	
<ul style="list-style-type: none"> La lista de correos debe aparecer de más recientes (página 1) a menos recientes (última página). 	

5.2.3.8 Historia H13

<i>Como usuario quiero poder generar las actas en mi idioma para poder distribuirlas posteriormente a público no multilingüe.</i>	
Título Corto	i18n back-end
Identificador	H13
EPIC	GENERACIÓN
Tiempo estimado (horas)	6
Tiempo necesario para completar (horas)	15
Criterios de aceptación	
<ul style="list-style-type: none"> Los textos de las actas generadas deben estar en el mismo idioma que el usado en la GUI. Por supuesto, esto sólo se aplicará a las etiquetas, no al contenido de los correos. 	

5.2.4 Historias del Tercer Sprint

En el tercer *sprint* se seguirá con el trabajo desarrollando las siguientes historias entre los días 14 de septiembre de 2021 y 15 de octubre de 2021, lo que hace un total de 24 días y 72 horas.

5.2.4.1 Historia H14

<i>Como usuario quiero poder exportar a formato CSV mis correos para poder procesarlos masivamente en una herramienta externa.</i>	
Título Corto	Exportar a CSV
Identificador	H14

EPIC	GENERACIÓN
Tiempo estimado (horas)	7
Tiempo necesario para completar (horas)	7,5
Criterios de aceptación	
<ul style="list-style-type: none"> • El usuario debe poder seleccionar varios correos con los que generar un archivo de tipo MIME "text/csv". • El usuario debe poder elegir las mismas opciones de exportación que con los PDF. • Los archivos CSV exportados no contendrán información extra personalizada (como sí hacen las actas PDF). • Se generará un archivo CSV que se descargará en el navegador. • El archivo CSV tendrá cabeceras internacionalizadas para cada una de las columnas. • Los valores irán entre comillas dobles (") y se separarán entre sí por medio de ",". Una fila sería de la forma: "<e1>";"<e2>";"<e3>". 	

5.2.4.2 Historia H15

<i>Como Jefe de Proyecto quiero que la aplicación permita el acceso a múltiples proveedores de correo para ampliar su público objetivo.</i>	
Título Corto	Más proveedores de correo
Identificador	H15
EPIC	CONEXIÓN
Tiempo estimado (horas)	5
Tiempo necesario para completar (horas)	4,5
Criterios de aceptación	
<ul style="list-style-type: none"> • Debe permitirse el acceso con todos los proveedores posibles especificando manualmente su host. • Deberá haber también una lista que relacione dominios de correo con sus correspondientes hosts, de manera que cuando se ponga una dirección de correo que esté en la lista, el host se configure de forma automática. • La lista anterior deberá poder ser modificada por el administrador de la aplicación. 	

5.2.4.3 Historia H16

<i>Como usuario quiero ver los correos de todas mis carpetas (recibidos, enviados, spam...) para poder exportar las conversaciones completas</i>	
Título Corto	Gestionar múltiples carpetas
Identificador	H16
EPIC	LISTADO
Tiempo estimado (horas)	10
Tiempo necesario para completar (horas)	9,5
Criterios de aceptación	
<ul style="list-style-type: none"> • Deben mostrarse todos los correos de todas las carpetas del usuario (Solo se mostrarán las carpetas del nivel más bajo, deshaciendo la jerarquía de carpetas). • Debe poder seleccionarse cualquiera de los correos para exportación. • Pueden exportarse en el mismo acta correos de varias carpetas. • No se mostrarán las carpetas vacías. 	

5.2.4.4 Historia H17

<i>Como Jefe de Proyecto quiero que se muestre un aviso de cookies adecuado para cumplir con la legalidad vigente.</i>	
Título Corto	Aviso de Cookies
Identificador	H17
EPIC	OPS
Tiempo estimado (horas)	4
Tiempo necesario para completar (horas)	3,5
Criterios de aceptación	
<ul style="list-style-type: none"> • Debe mostrarse al usuario un aviso de cookies que cumpla con los requisitos legales. • El aviso debe mostrarse al entrar en la aplicación. • Una vez que el usuario lo oculte ya no se le mostrará más al inicio. • Si está oculto deberá poder mostrarlo manualmente. 	

5.2.4.5 Historia H18

Como Jefe de Proyecto quiero que la comunicación se haga a través del protocolo HTTPS para evitar la interceptación de comunicaciones.

Título Corto	Certificado digital HTTPS
Identificador	H18
EPIC	OPS
Tiempo estimado (horas)	6
Tiempo necesario para completar (horas)	3
Criterios de aceptación	
<ul style="list-style-type: none"> Las comunicaciones entre el navegador del usuario y el <i>front-end</i> deberán hacerse por medio del protocolo HTTPS. Las comunicaciones entre el <i>front-end</i> y el <i>back-end</i> deberán hacerse por medio del protocolo HTTPS. Se deberá usar un certificado digital auto firmado. 	

5.2.4.6 Historia H19

Como usuario inexperto quiero ver en algún lado instrucciones de cómo configurar Gmail para evitar confusiones por las medidas extra de seguridad que tiene este proveedor.

Título Corto	Advertencia correos Google
Identificador	H19
EPIC	CONEXION
Tiempo estimado (horas)	5
Tiempo necesario para completar (horas)	3,5
Criterios de aceptación	
<ul style="list-style-type: none"> Si falla la autenticación en Gmail, debe mostrarse un mensaje de error que advierta que puede ser por la falta de una contraseña de aplicación. El mensaje debe contener un enlace a la página de Google que explica cómo conseguirla. 	

5.2.4.7 Historia H20

<i>Como usuario quiero descargar los archivos adjuntos de mis conversaciones de correo para poder dejar constancia completa de los envíos.</i>	
Título Corto	Mostrar adjuntos
Identificador	H20
EPIC	GENERACIÓN
Tiempo estimado (horas)	6
Tiempo necesario para completar (horas)	9,5
Criterios de aceptación	
<ul style="list-style-type: none"> • Se añadirá un botón que indique si se quieren descargar los archivos adjuntos. • Si se marca la opción de exportación correspondiente, los archivos adjuntos de todos los correos se empaquetarán en formato ZIP y se descargarán junto al acta. • En el mismo paquete, se incluirá también el cuerpo de los correos, como un archivo independiente y en formato HTML. • En la lista de correos se mostrará una lista con los nombres de los archivos adjuntos de cada uno. 	

5.2.4.8 Historia H21

<i>Como usuario quiero tener una opción de exportación que me indique en el acta que participantes hay en la conversación, para poder ver de manera sencilla quién formó parte de ella.</i>	
Título Corto	Resumir participantes conversación
Identificador	H21
EPIC	LISTADO
Tiempo estimado (horas)	5
Tiempo necesario para completar (horas)	3,5
Criterios de aceptación	

- Debe incluirse una nueva opción en la pantalla de exportación.
- Si la opción se marca, debería mostrarse en el acta una lista de correos, con una entrada para cada uno de los participantes.
- Un participante será aquella persona que envía o recibe un correo.

5.2.4.9 Historia H22

Como usuario quiero poder volver desde la pantalla de exportación a la de selección de correos para poder subsanar posibles errores.

Título Corto	Botón atrás en la exportación
Identificador	H22
EPIC	GENERACIÓN
Tiempo estimado (horas)	3
Tiempo necesario para completar (horas)	3,5
Crterios de aceptación	
<ul style="list-style-type: none"> • Una vez en la pantalla de exportación, debe incluirse alguna manera para poder volver a la pantalla anterior. • No debe cerrarse la sesión al retornar. • Deben mantenerse seleccionados los correos previamente seleccionados. • Los posibles campos personalizados y opciones marcadas deben mantenerse en el mismo estado que cuando se retornó. 	

5.2.4.10 Historia H23

Como usuario quiero ver mis correos renderizados como HTML, tanto en el listado como en el acta generado para no perder datos que no sean texto plano (enlaces, estilos...).

Título Corto	Mostrar correos en formato HTML
Identificador	H23
EPIC	GENERACIÓN
Tiempo estimado (horas)	9

Tiempo necesario para completar (horas)	9,5
Criterios de aceptación	
<ul style="list-style-type: none"> • En el listado, el cuerpo de los correos deberá mostrarse como HTML en lugar de en texto plano, si el correo se envió en este formato. • Los correos cuyo cuerpo esté en formato HTML deberán incluirse en el acta en este mismo formato. • Deberán mantenerse estilos, pero se eliminarán los scripts, por motivos de seguridad. 	

5.2.4.11 Historia H24

<i>Como usuario quiero que se muestre a quién se le envió copia de los correos (CC) en las actas para tener una exportación más exhaustiva.</i>	
Título Corto	Añadir campo CC
Identificador	H24
EPIC	GENERACIÓN
Tiempo estimado (horas)	3
Tiempo necesario para completar (horas)	3,5
Criterios de aceptación	
<ul style="list-style-type: none"> • Deberá añadirse el campo CC, tratándolo de la misma manera que los campos enviado y recibido. • Deberá mostrarse el campo CC en el cuerpo y añadirse al acta. • Habrá una opción de exportación para activar/desactivar la muestra de ese campo. 	

5.2.4.12 Historia H25

<i>Como usuario quiero que se pueda ver en el acta diferentes estadísticas de los participantes en la conversación para tener una idea de lo que contiene el acta.</i>	
Título Corto	Añadir estadísticas participantes
Identificador	H25

EPIC	GENERACIÓN
Tiempo estimado (horas)	3
Tiempo necesario para completar (horas)	3,5
Criterios de aceptación	
<ul style="list-style-type: none"> • La lista de participantes del acta será una tabla con cuatro columnas: "Correo del participante", "Mensajes enviados", "Mensajes recibidos", "Mensajes CC". • Para cada participante habrá una fila. • Esta funcionalidad se activará/desactivará a través de una opción de exportación. 	

5.2.4.13 Historia H26

<i>Como desarrollador quiero que la aplicación se pueda desplegar usando Docker para facilitar el mantenimiento y las pruebas.</i>	
Título Corto	Desplegar en Docker
Identificador	H26
EPIC	OPS
Tiempo estimado (horas)	6
Tiempo necesario para completar (horas)	7,5
Criterios de aceptación	
<ul style="list-style-type: none"> • El <i>front-end</i> y el <i>back-end</i> deberán desplegarse como una aplicación multi-contenedor usando Docker-Compose. • El <i>front-end</i> se desplegará usando Nginx. • El <i>back-end</i> se desplegará como un .jar. 	

5.3 Requisitos del Sistema

En este apartado se incluirán los requisitos del sistema correspondientes a cada una de las historias de usuario anteriores. Junto a ellos, se podrán ver los diferentes actores del sistema y los casos de uso, analizados a lo largo del desarrollo.

5.3.1 Obtención de los Requisitos del Sistema

Si usásemos una metodología clásica, esto sería una lista de requisitos funcionales y no funcionales obtenidos al principio del desarrollo a partir de los diferentes interesados. Estos formarían la línea base de requisitos y posteriormente, cuando se requiriese modificar alguno o añadir nuevos, se seguiría un procedimiento de gestión. Estaría compuesto por una serie de pasos donde se solicitaría lo que se quiere cambiar y tras analizar detalladamente su viabilidad, se modificaría la línea base.

Pero en este proyecto, usamos Scrum, y por tanto no tenemos esa lista inicial, que “nos dice lo que hay que hacer”. Sin embargo, no podemos obviar los requisitos y seguiremos un sistema similar al que se usaría en ese procedimiento de gestión de requisitos de un desarrollo clásico. Para esto, cada historia funcionará como un cambio solicitado, que se irá añadiendo a los que ya tenemos.

Quizás se podría pensar en ignorar la gestión de requisitos y trabajar sólo con las historias de usuario, pero el uso de estos nos ayuda a crear productos más completos y fiables. Seguir el proceso de identificación, validación, verificación y análisis nos ayuda a pensar en toda la casuística que envuelve a los diferentes componentes de la aplicación y nos proporciona una base más sólida de la que partir en la fase de diseño.

En las próximas líneas incluiremos los requisitos obtenidos a lo largo de todo el desarrollo y la historia con la que se corresponden. Algunos requisitos no funcionales no se corresponden con ninguna historia de usuario en concreto, pues se refieren a capacidades que debe tener el sistema en conjunto, sin ser de ninguna parte en concreto y afectando a varias de ellas.

A la hora de incluir aquí los requisitos funcionales hemos decidido mostrar primero los del primer *sprint* y después la lista completa actualizada al final de cada uno de los siguientes, destacando los cambios realizados, para ver de esta manera la evolución de nuestra línea base. Los requisitos no funcionales se incluyen al final de estos, y para ellos no mostraremos la evolución, pues no aporta gran cosa al lector de este documento.

5.3.1.1 Requisitos funcionales del Primer Sprint

En este apartado se incluye la lista de requisitos tal y como se encontraba al final del primer *sprint*.

Código	Historia	Requisito
Con1	H1	El sistema deberá abrir una conexión con el servidor de correo del usuario
Con1.1	H1	El usuario deberá especificar sus credenciales para la conexión
Con1.1.1	H1	El usuario deberá especificar su correo electrónico
Con1.1.2	H1	El usuario deberá especificar la contraseña de su correo electrónico
Con1.2	H1	Se abrirá una conexión con el servidor adecuado

Con1.2.1	H1	Si la cuenta es del dominio "uniovi.es" se conectará al servidor de correo outlook.office365.com
Con1.2.2	H1	Si la cuenta es del dominio "gmail.com" se conectará al servidor de correo smtp.gmail.com
Con1.3	H1	Si la conexión abierta en Con1.2 es exitosa se informará al usuario
Con1.3.1	H1	Si se producen errores al intentar conectarse a Con1.2 se informará al usuario
Con2	H2	El sistema deberá permitir al usuario desconectarse del servidor de correo
Con2.1	H2	El sistema deberá cerrar todas las conexiones abiertas que tenga con el servidor de correo del proveedor del usuario
Con2.2	H2	Se deberá notificar al usuario si el cierre de la conexión fue satisfactorio o no
List1	H3	El sistema deberá listar los correos recibidos en la cuenta de correo cuya conexión fue abierta en Con1
List1.1	H3	Para cada correo se deberá mostrar el asunto del mismo
List1.1.1	H3	Si el correo no tuviera asunto se mostrará el texto "<sin asunto>"
List1.2	H3	Si se produjesen errores obteniendo los correos se notificará el error
Gen1	H5	El sistema deberá permitir generar un acta con una serie de correos seleccionados
Gen1.1	H5	El sistema deberá permitir seleccionar un número ilimitado de correos entre los listados en List1
Gen1.2	H5	El sistema deberá generar un documento PDF que será el acta
Gen1.2.1	H5	El documento PDF generado deberá poder ser descargado por el usuario a su sistema local
Gen1.2.2	H5	El acta contendrá los correos seleccionados en Gen1.1
Gen1.2.2.1	H5	Para cada correo se incluirá en el acta el asunto del mismo
Gen1.2.2.2	H5	Para cada correo se incluirá en el acta el cuerpo del mismo
Gen1.2.2.2.1	H5	Sólo se mostrará el texto contenido en el cuerpo del correo
Gen1.2.2.2.2	H5	El cuerpo del correo no incluirá imágenes ni otros elementos multimedia
Gen1.2.2.3	H5	Para cada correo se incluirá en el acta la fecha de envío del mismo
Gen1.2.2.3.1	H5	El usuario deberá poder elegir si las fechas de envío se incluyen o no
Gen1.2.2.4	H4	El acta podrá contener un título
Gen1.2.2.4.1	H4	El título deberá ser especificado por el usuario
Gen1.2.2.4.2	H4	El título es opcional, pudiendo no incluirse si así lo especifica el usuario
Gen1.2.2.5	H4	El acta deberá contener una fecha de acta
Gen1.2.2.5.1	H4	La fecha es un campo opcional
Gen1.2.2.5.2	H4	La fecha predeterminada será la de generación del acta

Gen1.2.2.5.3	H4	El usuario podrá modificar esta fecha incluyendo cualquier otro valor de texto
Gen1.2.2.5.4	H4	El formato predeterminado de fecha será: "dd/mm/aaaa hh:mm:ss"

Tabla 13 – Lista de requisitos funcionales del primer sprint

5.3.1.2 Requisitos funcionales del Segundo Sprint

La implementación de las Historias del Segundo *Sprint* provocó cambios en la lista de requisitos. Siguiendo el procedimiento definido, se añadieron algunos y se modificaron otros.

A continuación, se incluye la lista completa, tal y como estaba al final del segundo *sprint*. En verde, se indica la información añadida, en rojo y tachado, la información eliminada. Además, la columna “Modificaciones” aclara los cambios realizados.

Código	Historia	Requisito	Modificaciones
Con1	H1	El sistema deberá abrir una conexión con el servidor de correo del usuario	
Con1.1	H1	El usuario deberá especificar sus credenciales para la conexión	
Con1.1.1	H1	El usuario deberá especificar su correo electrónico	
Con1.1.2	H1	El usuario deberá especificar la contraseña de su correo electrónico	
Con1.2	H1	Se abrirá una conexión con el servidor adecuado	
Con1.2.1	H1	Si la cuenta es del domino "uniovi.es" se conectará al servidor de correo outlook.office365.com	
Con1.2.2	H1	Si la cuenta es del domino "gmail.com" se conectará al servidor de correo smtp.gmail.com	
Con1.3	H1	Si la conexión abierta en Con1.2 es exitosa se informará al usuario	
Con1.3.1	H1	Si se producen errores al intentar conectarse a Con1.2 se informará al usuario	
Con2	H2	El sistema deberá permitir al usuario desconectarse del servidor de correo	
Con2.1	H2	El sistema deberá cerrar todas las conexiones abiertas que tenga con el servidor de correo del proveedor del usuario	
Con2.2	H2	Se deberá notificar al usuario si el cierre de la conexión fue satisfactorio o no	
List1	H3	El sistema deberá listar los correos recibidos en la cuenta de correo cuya conexión fue abierta en Con1	
List1.1	H3	Para cada correo se deberá mostrar el asunto del mismo	Antes se mostraba sólo el asunto del correo, ahora se muestran más datos
List1.1	H6	Para cada correo se deberá mostrar una serie de datos	
List1.1.1	H6	Se mostrará el asunto de los correos	
List1.1.1 List.1.1.1.1	H3	Si el correo no tuviera asunto se mostrará el texto "<sin asunto>"	

List1.1.2	H6	Se mostrará la fecha y hora de envío de los correos	
List1.1.2.1	H6	La fecha y hora tendrán precisión de minutos	
List1.1.3	H6	Se mostrará el remitente de los correos	
List1.1.3.1	H6	Se indicará nombre y dirección del remitente	
List1.1.4	H6	Se mostrará el destinatario o destinatarios de los correos	
List1.1.4.1	H6	Se indicará nombre y dirección de los destinatarios	
List1.1.5	H6	Se mostrará el cuerpo de los correos	
List1.1.5.1	H6	Se incluirá solo el texto del cuerpo, sin ningún tipo de elemento multimedia (fotos, enlaces...)	
List1.2	H3	Si se produjesen errores obteniendo los correos se notificará el error	
List1.3	H12	Los correos se mostrarán en orden descendiente de fecha y hora (más recientes primero)	Se especifica el orden de los correos, antes se mostraban como venían del proveedor
Gen1	H5	El sistema deberá permitir generar un acta con una serie de correos seleccionados	
Gen1.1	H5	El sistema deberá permitir seleccionar un número ilimitado de correos entre los listados en List1	
Gen1.2	H5	El sistema deberá generar un documento PDF que será el acta	
Gen1.2.1	H5	El documento PDF generado deberá poder ser descargado por el usuario a su sistema local	
Gen1.2.2	H5	El acta contendrá los correos seleccionados en Gen1.1	
Gen1.2.2.1	H5	Para cada correo se incluirá en el acta el asunto del mismo	
Gen1.2.2.2	H5	Para cada correo se incluirá en el acta el cuerpo del mismo	
Gen1.2.2.2.1	H5	Sólo se mostrará el texto contenido en el cuerpo del correo	
Gen1.2.2.2.2	H5	El cuerpo del correo no incluirá imágenes ni otros elementos multimedia	
Gen1.2.2.3	H5	Para cada correo se incluirá en el acta la fecha de envío del mismo	
Gen1.2.2.3.1	H5	El usuario deberá poder elegir si las fechas de envío se incluyen o no	
Gen1.2.2.4	H7	El acta podrá contener una serie de campos personalizados	Se sustituyen los campos personalizados estáticos que había antes, por unos nuevos
Gen1.2.2.4.1	H7	Los campos personalizados estarán formados por un par compuesto por una etiqueta y un valor	
Gen1.2.2.4.2	H7	Una etiqueta será un texto, que puede estar vacío	

Gen1.2.2. 4.3	H7	Un valor será un texto, que puede estar vacío	que son dinámicos
Gen1.2.2. 4.4	H7	El usuario podrá añadir tantos campos personalizados como desee	
Gen1.2.2. 4.5	H7	Los campos personalizados se mostrarán en el acta generado y serán la etiqueta y el valor.	
Gen1.2.2. 4	H4	El acta podrá contener un título	
Gen1.2.2. 4.1	H4	El título deberá ser especificado por el usuario	
Gen1.2.2. 4.2	H4	El título es opcional, pudiendo no incluirse si así lo especifica el usuario	
Gen1.2.2. 5	H4	El acta deberá contener una fecha de acta	
Gen1.2.2. 5.1	H4	La fecha es un campo opcional	
Gen1.2.2. 5.2	H4	La fecha predeterminada será la de generación del acta	
Gen1.2.2. 5.3	H4	El usuario podrá modificar esta fecha incluyendo cualquier otro valor de texto	
Gen1.2.2. 5.4	H4	El formato predeterminado de fecha será: "dd/mm/aaaa-hh:mm:ss"	

Tabla 14 – Requisitos funcionales del segundo sprint

5.3.1.3 Requisitos funcionales del Tercer Sprint

De la misma manera que en el segundo *sprint*, en este tercero se volvió a modificar la lista de requisitos. Siguiendo el código de colores descrito entonces se procede a especificar como queda la lista, que es la definitiva al final de este proyecto.

Código	Historia	Requisito	Modificaciones
Con1	H1	El sistema deberá abrir una conexión con el servidor de correo del usuario	
Con1.1	H1	El usuario deberá especificar sus credenciales para la conexión	
Con1.1.1	H1	El usuario deberá especificar su correo electrónico	
Con1.1.2	H1	El usuario deberá especificar la contraseña de su correo electrónico	
Con1.1.3	H15	El usuario podrá especificar el host de su proveedor de correo electrónico	Se elimina la resolución de hosts estáticos y se añade un campo host dinámico que puede personalizar el usuario, junto a un sistema que automatiza la resolución de
Con1.2	H1	Se abrirá una conexión con el servidor adecuado	
Con1.2.1	H1	Si la cuenta es del dominio "uniovi.es" se conectará al servidor de correo outlook.office365.com	
Con1.2.2	H1	Si la cuenta es del dominio "gmail.com" se conectará al servidor de correo smtp.gmail.com	
Con1.2.1	H15	Si se especifica un host en Con1.1.3 se intentará abrir la conexión con el host especificado	
Con1.2.2	H15	Si no se especifica un host en Con1.1.3 se intentará abrir la conexión con el proveedor adecuado de la lista de proveedores para el dominio de la	

		dirección de correo especificada en Con.1.1.1.	algunos nombres
Con1.2.2.1	H15	La lista de proveedores contendrá la equivalencia entre el dominio de correo y el host.	
Con1.2.2.2	H15	La lista de proveedores podrá ser modificada por el administrador del sistema.	
Con1.3	H1	Si la conexión abierta en Con1.2 es exitosa se informará al usuario	
Con1.3.1	H1	Si se producen errores al intentar conectarse a Con1.2 se informará al usuario	
Con2	H2	El sistema deberá permitir al usuario desconectarse del servidor de correo	
Con2.1	H2	El sistema deberá cerrar todas las conexiones abiertas que tenga con el servidor de correo del proveedor del usuario	
Con2.2	H2	Se deberá notificar al usuario si el cierre de la conexión fue satisfactorio o no	
Con3	H18	Se indicará como conseguir una contraseña de aplicación para Gmail.	
List1	H3	El sistema deberá listar los correos recibidos en la cuenta de correo cuya conexión fue abierta en Con1	
List1.1	H6	Para cada correo se deberá mostrar una serie de datos	
List1.1.1	H6	Se mostrará el asunto de los correos	
List1.1.1.1	H3	Si el correo no tuviera asunto se mostrará el texto "<sin asunto>"	
List1.1.2	H6	Se mostrará la fecha y hora de envío de los correos	
List1.1.2.1	H6	La fecha y hora tendrán precisión de minutos	
List1.1.3	H6	Se mostrará el remitente de los correos	
List1.1.3.1	H6	Se indicará nombre y dirección del remitente	
List1.1.4	H6	Se mostrará el destinatario o destinatarios de los correos	
List1.1.4.1	H6	Se indicará nombre y dirección de los destinatarios	
List1.1.5	H6	Se mostrará el cuerpo de los correos	
List1.1.5.1	H6	Se incluirá solo el texto del cuerpo, sin ningún tipo de elemento multimedia (fotos, enlaces...)	Antes se mostraba sólo el texto de los correos, ahora se muestran el HTML con todos sus elementos multimedia
List1.1.5.1	H23	Se incluirá el cuerpo de los correos como HTML	
List1.1.6	H20	Se mostrará una lista con el nombre de los archivos adjuntos de cada correo	Se añade campo en el listado de correos
List1.1.7	H24	Se mostrará la copia de carbón de los correos	Se añade campo en el listado de
List1.1.7.1	H24	Se indicará el nombre y la dirección de los destinatarios de la copia de carbón	

			correos
List1.2	H3	Si se produjesen errores obteniendo los correos se notificará el error	
List1.3	H12	Los correos se mostrarán en orden descendiente de fecha y hora (más recientes primero)	
List1.4	H16	Se mostrarán los correos de todas las carpetas	Se añade la división de los correos por carpetas y la lectura de aquellas que no sean INBOX
List1.4.1	H16	No se mostrarán aquellas carpetas que no contengan mensajes	
Gen1	H5	El sistema deberá permitir generar un acta con una serie de correos seleccionados	
Gen1.1	H5	El sistema deberá permitir seleccionar un número ilimitado de correos entre los listados en List1	
Gen1.2	H5	El sistema deberá generar un documento PDF que será el acta	
Gen1.2.1	H5	El documento PDF generado deberá poder ser descargado por el usuario a su sistema local	
Gen1.2.2	H5	El acta contendrá los correos seleccionados en Gen1.1	
Gen1.2.2.1	H5	Para cada correo se incluirá en el acta el asunto del mismo	
Gen1.2.2.2	H5	Para cada correo se incluirá en el acta el cuerpo del mismo	
Gen1.2.2.2.1	H5	Sólo se mostrará el texto contenido en el cuerpo del correo	Antes se generaba el acta con el cuerpo de los correos como texto, ahora se añade como HTML
Gen1.2.2.2.2	H5	El cuerpo del correo no incluirá imágenes ni otros elementos multimedia	
Gen1.2.2.2.1	H23	Se mostrará el cuerpo en formato HTML	
Gen1.2.2.2.2	H23	Si se produjesen problemas incluyendo el HTML en el archivo, se indicará que se puede consultar en los archivos adjuntos.	
Gen1.2.2.3	H5	Para cada correo se incluirá en el acta la fecha de envío del mismo	
Gen1.2.2.3.1	H5	El usuario deberá poder elegir si las fechas de envío se incluyen o no	
Gen1.2.2.4	H7	El acta podrá contener una serie de campos personalizados	
Gen1.2.2.4.1	H7	Los campos personalizados estarán formados por un par compuesto por una etiqueta y un valor	
Gen1.2.2.4.2	H7	Una etiqueta será un texto, que puede estar vacío	
Gen1.2.2.4.3	H7	Un valor será un texto, que puede estar vacío	
Gen1.2.2.4.4	H7	El usuario podrá añadir tantos campos personalizados como desee	
Gen1.2.2.4.5	H7	Los campos personalizados se mostrarán en el acta generado y serán la etiqueta y el valor.	

Gen1.2.2.5	H20	El acta contendrá el nombre de los archivos adjuntos de cada correo	Se añade un campo nuevo
Gen1.2.2.6	H25	El acta contendrá una tabla que resuma los participantes de los correos del acta.	Se añade un elemento al acta con la H21 y se amplía su funcionalidad con la H25
Gen1.2.2.6.1	H21	Será un participante cada email que salga en el remitente, el destinatario o la copia de carbón de alguno de los correos del acta.	
Gen1.2.2.6.2	H21	Para cada participante se mostrará el email	
Gen1.2.2.6.3	H21	Para cada participante se mostrarán todos los nombres que aparezcan junto al email.	
Gen1.2.2.6.4	H25	Para cada participante se mostrará el número de mensajes enviados	
Gen1.2.2.6.5	H25	Para cada participante se mostrará el número de mensajes recibidos	
Gen1.2.2.6.6	H25	Para cada participante se mostrará el número de mensajes en los que es destinatario de copia de carbón	
Gen1.2.2.7	H24	El acta contendrá los destinatarios de la copia de carbón	Nueva opción de exportación
Gen1.2.2.7.1	H24	El usuario podrá indicar si los destinatarios de la copia de carbón se incluyen o no	
Gen1.3	H14	El usuario podrá generar un archivo CSV	Se añade la generación de archivos CSV
Gen1.3.1	H14	El archivo CSV generado deberá poder ser descargado por el usuario a su sistema local.	
Gen1.3.2	H14	El archivo contendrá los correos seleccionados en Gen1.1	
Gen1.3.3	H14	Para cada correo se incluirán los siguientes datos	
Gen1.3.3.1	H14	El asunto del correo	
Gen1.3.3.2	H14	La fecha de envío del correo	
Gen1.3.3.2.1	H14	El usuario deberá poder elegir si las fechas de envío se incluyen o no	
Gen1.3.3.3	H14	Los archivos adjuntos del correo	
Gen1.3.3.3.1	H14	El usuario deberá poder elegir si los archivos adjuntos del correo se incluyen o no	
Gen1.3.3.4	H14	Las direcciones del remitente, destinatario y copia de carbón	
Gen1.3.3.4.1	H14	El usuario deberá poder elegir si las direcciones se incluyen o no	
Gen1.3.3.5	H14	El cuerpo del correo	
Gen1.4	H20	El usuario podrá descargar una serie de archivos adjuntos	Se añade la funcionalidad de exportación de archivos adjuntos
Gen1.4.1	H20	Se generará un archivo ZIP que podrá ser descargado por el usuario a su sistema local.	
Gen1.4.1.1	H20	Los archivos adjuntos de los correos se incluirán en el ZIP generado	
Gen1.4.1.2	H20	Cada uno de los cuerpos de los correos se incluirá en el ZIP generado como un archivo diferente.	

Tabla 15 – Requisitos funcionales del tercer sprint

5.3.1.4 Requisitos No Funcionales

A continuación, se muestra la lista de requisitos no funcionales considerados para el desarrollo de este producto.

Código	Requisito
RNF1	La aplicación web deberá funcionar correctamente en Mozilla Firefox 93.0, Google Chrome 95.0 y Microsoft Edge 94.0
RNF2	Las acciones de usuario deberán tener una respuesta en menos de 0,5 segundos.
RNF3	Las sesiones deberán ser independientes para cada usuario, no pudiendo acceder uno a la información de otro
RNF4	El sistema no podrá almacenar datos personales de usuario
RNF5	El sistema deberá llevar un registro (log) de las acciones realizadas escribiendo al menos una línea por cada llamada a un controlador
RNF6	Los errores deberán registrarse en un log
RNF7	El sistema deberá cumplir con las pautas de accesibilidad WCAG 2.1, con el nivel AA.
RNF8	La aplicación deberá estar internacionalizada, y localizada a los idiomas asturiano, español e inglés, tanto para el <i>front-end</i> como para el <i>back-end</i> , aplicándose tanto a los textos como a las fechas y valores numéricos.
RNF9	Los tiempos de espera no pueden ser superiores a 0,5 segundos al cargar páginas previamente cargadas en la lista de correos.
RNF10	La interfaz deberá incluir el logo de la Universidad de Oviedo/Uviéu y de la Escuela de Ingeniería Informática
RNF11	Deberá poder navegarse por toda la aplicación usando el teclado, sin necesidad de ratón.
RNF12	Debe mostrarse un aviso de cookies que cumpla con los requisitos legales
RNF13	La comunicación entre el <i>front-end</i> y el <i>back-end</i> se hará a través de HTTPS con un certificado auto firmado.
RNF14	La aplicación se desplegará usando contenerizada Docker
RNF14.1	El <i>front-end</i> se desplegará usando Nginx
RNF14.2	El <i>back-end</i> se desplegará como un JAR

Tabla 16 – Lista de requisitos no funcionales

5.3.2 Identificación de Actores del Sistema

Una vez vistos los requisitos, es el momento de extraer, a partir de ellos, quienes son los actores que interactúan con el sistema, que en nuestro caso es muy simple.

De manera activa, sólo actúan con el sistema los usuarios, que dividimos en dos tipos.

- Usuario no conectado: Este es el caso trivial. Los usuarios no conectados son aquellos que entran a la aplicación y aún no han introducido sus credenciales de correo, y, por tanto, no se ha establecido conexión con su proveedor.
- Usuario conectado: Es aquel que ya ha abierto una conexión con su proveedor de correo y por tanto puede acceder a sus conversaciones.

Los usuarios conectados no tienen roles específicos. Pues todos aquellos que se hayan conectado a su proveedor podrán realizar las mismas acciones.

5.3.3 Especificación de Casos de Uso

Estos actores podrán realizar una serie de acciones sobre la aplicación, que describiremos en los siguientes casos de uso.

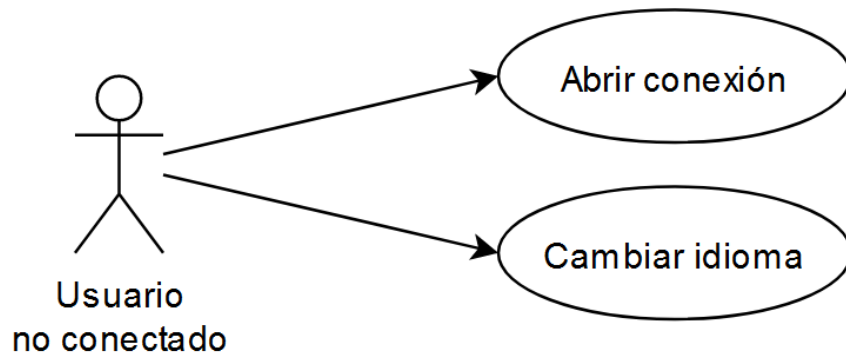


Figura 5 – Casos de Uso del Usuario no conectado

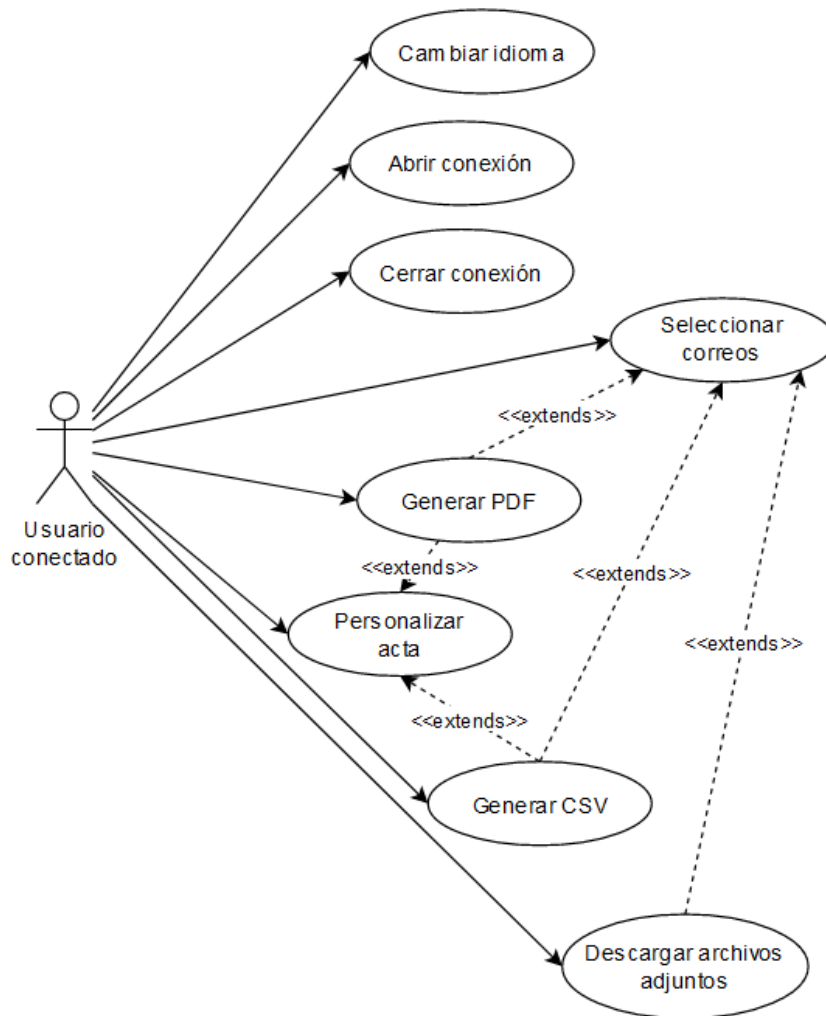


Figura 6 – Casos de Uso del usuario conectado

En la Figura 5 se muestran los casos de uso para un usuario no conectado y en la Figura 6 para aquel usuario que sí lo haya hecho.

Las siguientes tablas, describen los diferentes casos de uso de ambos actores.

Nombre del Caso de Uso	
Abrir conexión	
Descripción	
El usuario creará una conexión con su proveedor de correo, especificando:	
<ul style="list-style-type: none"> • Su correo electrónico • Su contraseña • El host de su proveedor de correo 	

Nombre del Caso de Uso	
Cambiar idioma	
Descripción	

Se mostrará una opción para cada uno de los idiomas soportados (asturiano, español e inglés) que cambiará el idioma en que se muestra la aplicación.

Nombre del Caso de Uso

Cerrar conexión

Descripción

Se cerrará la conexión con el proveedor de correo, no pudiendo a partir de este punto recuperar más correos.
--

Nombre del Caso de Uso

Ver correos

Descripción

El usuario podrá ver el asunto de los correos electrónicos que haya recibido. También podrá ver la fecha de envío, el remitente y los destinatarios, así como el cuerpo del correo.

Nombre del Caso de Uso

Seleccionar correos

Descripción

El usuario podrá seleccionar una serie de correos que le interesen.

Nombre del Caso de Uso

Personalizar acta

Descripción

El usuario podrá especificar una serie de opciones para generar un acta. Estas opciones serán:
--

- | |
|--|
| <ul style="list-style-type: none"> • Determinar si quiere que se incluyan las fechas de envío • Determinar si quiere que se muestre el nombre de los archivos adjuntos • Determinar si quiere que se incluya el resumen de los participantes • Determinar si quiere que se incluyan las direcciones de los correos (remitente, destinatario y CC). |
|--|

También podrá añadir un número ilimitado de campos de texto, compuestos por una etiqueta y un valor (dos textos planos), que podrán añadirse o eliminarse a voluntad.

Nombre del Caso de Uso

Generar PDF

Descripción

El usuario podrá descargar un acta en formato PDF que contenga el texto de los correos seleccionados en el caso de uso "Seleccionar correos".

La generación del acta se hará de acuerdo con las opciones especificadas en el caso de uso "Personalizar acta".

Nombre del Caso de Uso
Generar CSV
Descripción
El usuario podrá descargar en formato CSV los datos de los correos seleccionados en el caso de uso "Seleccionar correos". La generación del archivo se hará de acuerdo con las opciones especificadas en el caso de uso "Personalizar acta".

Nombre del Caso de Uso
Descargar archivos adjuntos
Descripción
El usuario podrá descargar los archivos adjuntos de los correos seleccionados, todos juntos en un archivo ZIP.

5.4 Identificación de los Subsistemas en la Fase de Análisis

Una vez vistos los casos de uso, nos tocaría subdividir el sistema completo en otros más pequeños. Además de basarnos en los requisitos obtenidos, también estamos influenciados por alguna de las decisiones tomadas en la Evaluación de Alternativas.

5.4.1 Descripción de los Subsistemas

Los dos subsistemas quedan claros, como decíamos, ya se han decidido en la Evaluación de Alternativas. Son los siguientes:

- Por un lado, el *front-end*. El cliente web estará construido usando Angular, siendo una SPA. Este subsistema contendrá todas aquellas clases encargadas de la presentación final al usuario y se comunicará con el servicio web que actuará como *back-end*.
- El segundo subsistema será ese *back-end*, un servicio web REST a construir con SpringBoot. Implementará las clases relacionadas con la gestión de los correos electrónicos, comunicándose con el proveedor de correo, y, además, se encargará de la generación de las actas.

No tenemos más subsistemas, ya que, por ejemplo, en este punto no tenemos base de datos. Podríamos pensar en el proveedor de correo como subsistema, pero esto sería incorrecto, pues está fuera de nuestro sistema. Por último, podríamos considerar una mayor subdivisión de nuestros dos subsistemas, pero no lo creemos necesario. Según nuestro planteamiento poseen una buena cohesión interna y una buena independencia externa. Cualquier subdivisión dentro de cada uno de ellos daría lugar a una excesiva complejidad y a un incremento en el número de interfaces y llamadas entre ellas.

5.4.2 Descripción de los Interfaces entre Subsistemas

La comunicación es muy básica entre los dos subsistemas. Se hará por medio de llamadas desde el *front-end* al *back-end*. Para ello se usará el protocolo HTTP, que realizará peticiones al servicio web REST, expuesto en una IP concreta, que podría o no estar en la misma máquina, al fin y al cabo, el anfitrión en el que se desplieguen ambos no es relevante para la fase de desarrollo.

5.5 Diagrama de Clases Preliminar del Análisis

En este apartado vamos a seguir desgranando lo que será nuestro sistema por medio de un diagrama de clases. El objetivo en este punto no es el excesivo detalle, pero sí hacer una muestra precisa de lo que vamos a construir.

En el primer subapartado se incluye el diagrama de clases general y en el segundo van a irse desgranando todas ellas, una a una. Hemos decidido suprimir los atributos y operaciones de los diagramas, de esta manera serán más simples de comprender. Sin embargo, sí se incluyen en las tablas del segundo apartado, que es más propicio para dar detalles.

5.5.1 Diagrama de Clases

Tal y como dijimos cuando hablábamos de los subsistemas, intentamos que ambos sean independientes. Por este motivo, es coherente que ambos diagramas de clases sean independientes entre sí y así lo hemos hecho. Primero se muestra el diagrama de clases para la *front-end* y posteriormente para el *back-end*.

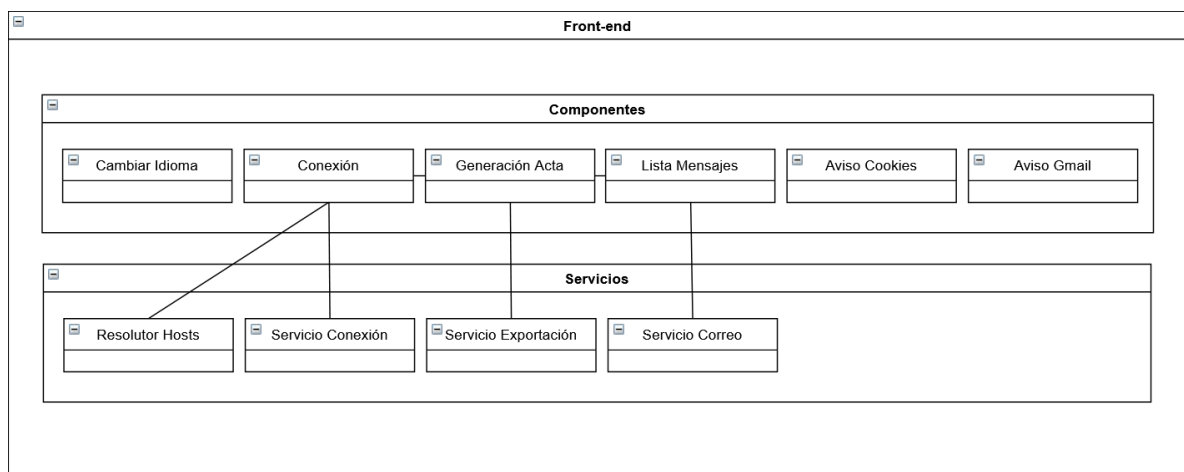


Figura 7 – Diagrama de Clases del front-end

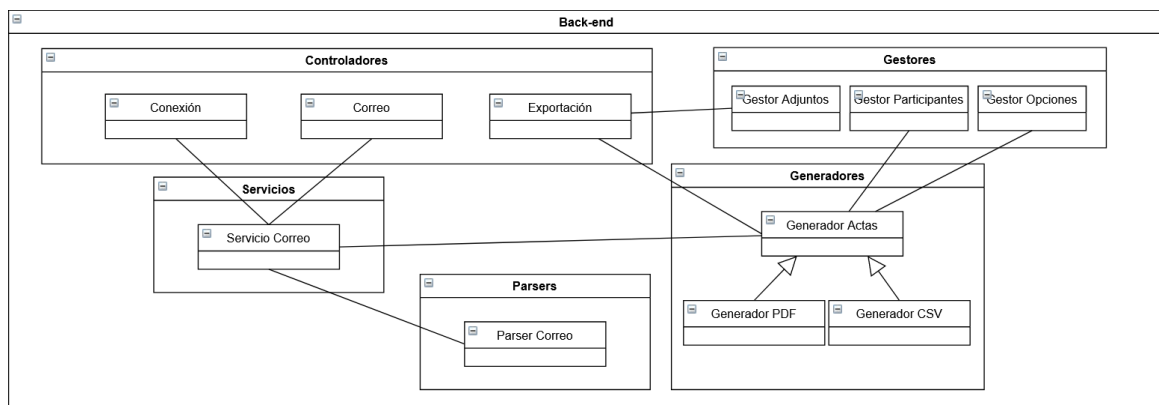


Figura 8 – Diagrama de Clases del back-end

En la Figura 7 podemos ver que el frontal se divide en dos capas. La de los componentes y la de los servicios. La primera de ellas contiene aquellas clases que se encargan principalmente de renderizar los datos y de mostrárselos al usuario, manejando todas aquellas interacciones que este haga. Pueden ser cosas del estilo, ocultar o mostrar botones, responder a diferentes eventos, en definitiva, aquellas que tratan aspectos relacionados con la interfaz de usuario.

Por otro lado, también deben comunicarse con los servicios para pedirles que obtengan los datos del *back-end*. Precisamente esa es la funcionalidad de la segunda capa, que contiene las clases que se encargan de realizar llamadas. En general, el frontal se encarga de obtener los datos y pintarlos en pantalla, sin embargo, hay una pequeña lógica que sí se computará en el frontal, la resolución de hosts en base a la dirección de correo especificada en las credenciales de conexión. Se realiza aquí para, además de independizar el *back-end* de los posibles proveedores, incrementar el rendimiento. También se extrae del servicio de conexión o del componente de conexión en el *front-end*, con el objetivo de facilitar una futura extensión de esta funcionalidad, hacia métodos como la resolución automática de dominios.

La Figura 8 nos muestra que el *back-end* se divide en cinco partes. Por un lado, los controladores se encargan de recibir las llamadas del exterior, procesar los parámetros y componer las respuestas. Sin embargo, el corazón del sistema es el servicio de correo. Este se encarga de comunicarse con el proveedor y manejar los datos de la sesión (correos, carpetas...). Los controladores de conexión y correo llaman directamente a este servicio, para abrir y cerrar la conexión o para obtener esos datos. El propio servicio al comunicarse con el proveedor necesita obtener el cuerpo de los correos, que puede estar en varios formatos o tener archivos adjuntos y también debe obtener los metadatos que nos sean de utilidad. Para todo ello, está el *parser* de correos, un módulo externo a este servicio. De manera diferente el controlador de exportación se encarga de las llamadas relacionadas con la generación de actas, de archivos de otro tipo o de los adjuntos. Para las actas, hay una serie de clases, con una estructura que permite la reutilización de código. La clase que domina la jerarquía es la que se comunica con el servicio de correos para obtener aquellos a exportar. Por último, tenemos una serie de gestores, que modelan pequeñas funcionalidades de tratamiento de datos, haciendo funciones simples que se pueden ejecutar aisladas y se llaman desde partes específicas del código. Un ejemplo de esto, es el gestor de participantes, que procesa los correos seleccionados y realiza un resumen de los participantes.

La comunicación entre los dos subsistemas se realizaría desde los servicios del primero, a los controladores del segundo.

5.5.2 Descripción de las Clases

En este punto describiremos con más detalle, aunque como decíamos antes, sin entrar en aspectos de bajo nivel, las diferentes clases contenidas en los diagramas anteriores. Para mostrarlas aquí hemos seguido la división por subsistemas establecida varios apartados atrás.

5.5.2.1 Front-end

Nombre de la Clase
Cambiar Idioma
Descripción
Componente encargado de la gestión de los diferentes idiomas de la aplicación.
Responsabilidades
- Devolver los textos en el idioma correcto, cuando se pidan. - Cambiar el idioma cuando el usuario lo solicite
Atributos Propuestos
Idioma Actual: Guarda el identificador del idioma actual en que se muestra la aplicación.
Métodos Propuestos
Cambiar Idioma: Dado un nuevo idioma lo establece como actual Traducir: Dado un texto clave, devuelve el texto en el idioma que esté establecido como actual

Nombre de la Clase
Conexión
Descripción
Componente encargado de la gestión de las conexiones, de cara al usuario.
Responsabilidades
- Iniciar el proceso de abrir una conexión - Iniciar el proceso de cerrar una conexión - Renderizar los textos correspondientes al éxito o al fracaso de los procesos - Manejar la interfaz de usuario correspondiente a la conexión (Manejar botones y elementos de la interfaz, manejar los eventos de usuario...)
Atributos Propuestos
n/a
Métodos Propuestos
Abrir conexión: Al recibir un correo y una contraseña, llama al servicio correspondiente para abrir una conexión y una vez que recibe respuesta imprime los mensajes de éxito o error adecuados. Cerrar conexión: Llama al servicio correspondiente para cerrar una conexión y una vez que recibe respuesta imprime los mensajes de éxito o error adecuados.

Nombre de la Clase
Generación Acta
Descripción
Componente encargado de la gestión de la generación del acta, de cara al usuario.
Responsabilidades
- Iniciar el proceso de generación del acta - Manejar las opciones de personalización del acta - Manejar la interfaz de usuario correspondiente a la generación del acta (Manejar elementos de la interfaz, manejar los eventos de usuario...)
Atributos Propuestos
datos: Mapa clave-valor donde se almacenan los textos personalizados, por ejemplo, "título" y valor del título o "fecha" y valor de la fecha. opciones: Lista de opciones disponibles a la hora de generar el acta (p. ej. añadir fecha de

envío a los correos) opcionesSeleccionadas: Lista de opciones seleccionadas por el usuario de entre las disponibles. formatoSeleccionado: El formato en que se quiere generar el acta (PDF/CSV).
Métodos Propuestos
<< constructor >>: Recibe la lista de los correos seleccionados en la lista de correos. generarActa: Inicia el proceso de generar el acta, pasando al servicio adecuado los correos y formato seleccionados y las opciones y títulos contenidos.

Nombre de la Clase
Lista Mensajes
Descripción
Componente encargado de pintar en pantalla los correos obtenidos y de gestionar los seleccionados.
Responsabilidades
- Pintar los correos del usuario - Manejar los correos seleccionados - Manejar la interfaz de usuario correspondiente a los correos (Manejar elementos de la interfaz, manejar los eventos de usuario...)
Atributos Propuestos
correos: Lista de los correos recibidos por el usuario correosSeleccionados: Lista de los correos seleccionados por el usuario para su exportación
Métodos Propuestos
obtenerCorreos: Llama al servicio correspondiente para obtener los correos recibidos por el usuario avanzarAExportación: Crea el componente de Generación de Actas pasándole como parámetro los correos seleccionados.

Nombre de la Clase
Aviso Cookies
Descripción
Componente encargado de pintar en pantalla el aviso de cookies.
Responsabilidades
- Mostrar en la pantalla el aviso de cookies. - Ocultar el aviso de cookies cuando el usuario lo pida. - Comprobar que el aviso sólo se muestra la primera vez que el usuario entra en la página.
Atributos Propuestos
N/A
Métodos Propuestos
mostrarAviso: Muestra el aviso de cookies. ocultarAviso: Oculta el aviso de cookies.

Nombre de la Clase
Aviso Gmail
Descripción
Componente encargado de pintar en pantalla las instrucciones para configurar Gmail.
Responsabilidades
- Mostrar en la pantalla el aviso de Gmail.

Atributos Propuestos
N/A
Métodos Propuestos
cerrarAviso: Cierra el <i>pop-up</i> en que se muestra este aviso.

Nombre de la Clase
Resolutor de Hosts
Descripción
Servicio encargado de obtener el <i>host</i> de las direcciones de correo.
Responsabilidades
- Dada una dirección de correo, obtener su <i>host</i> si está en la lista de hosts de la aplicación. - Manejar la lectura de la lista de <i>hosts</i> de la aplicación.
Atributos Propuestos
n/a
Métodos Propuestos
resolverHost: Dada una dirección de correo devuelve su <i>host</i> .

Nombre de la Clase
Servicio Conexión
Descripción
Servicio encargado de las llamadas al <i>back-end</i> relativas a la conexión.
Responsabilidades
- Realizar las llamadas al <i>back-end</i> para abrir y cerrar una conexión.
Atributos Propuestos
n/a
Métodos Propuestos
Abrir conexión: Realiza una petición HTTP al <i>back-end</i> , con los datos necesarios para abrir una conexión. Cerrar conexión: Realiza una petición al <i>back-end</i> , con los datos necesarios para cerrar una conexión.

Nombre de la Clase
Servicio Exportación
Descripción
Servicio encargado de las llamadas al <i>back-end</i> relativas a la exportación.
Responsabilidades
- Realizar las llamadas al <i>back-end</i> para generar un acta. - Realizar las llamadas al <i>back-end</i> para obtener los archivos adjuntos.
Atributos Propuestos
n/a
Métodos Propuestos
generarActa: Realiza una petición HTTP al <i>back-end</i> , con los datos necesarios para generar un acta. obtenerAdjuntos: Obtiene los archivos adjuntos de los correos desde el <i>back-end</i> .

Nombre de la Clase
Servicio Correo

Descripción
Servicio encargado de las llamadas al <i>back-end</i> relativas a la obtención de datos de correo.
Responsabilidades
- Realizar las llamadas al <i>back-end</i> para obtener los correos y carpetas del usuario.
Atributos Propuestos
n/a
Métodos Propuestos
obtenerCorreos: Realiza una petición HTTP al <i>back-end</i> , para obtener los correos del usuario. obtenerCarpetas: Realiza una petición HTTP al <i>back-end</i> , para obtener las carpetas del usuario.

5.5.2.2 *Back-end*

Nombre de la Clase
Controlador Conexión
Descripción
Controlador que recibe las peticiones correspondientes a una conexión
Responsabilidades
- Recibir las peticiones GET y POST correspondientes a abrir y cerrar una conexión. - Formatear los datos a devolver o los correspondientes errores
Atributos Propuestos
n/a
Métodos Propuestos
abrirConexión: Llama al servicio de correo con los datos correspondientes para abrir una conexión. cerrarConexión: Llama al servicio de correo solicitando cerrar una conexión.

Nombre de la Clase
Controlador Correo
Descripción
Controlador que recibe las peticiones correspondientes a la obtención de datos de correos
Responsabilidades
- Recibir las peticiones GET correspondientes a la obtención de datos de correos. - Formatear los datos a devolver o los correspondientes errores
Atributos Propuestos
n/a
Métodos Propuestos
obtenerCorreos: Llama al servicio de correo para obtener los correos de la sesión actual obtenerCarpetas: Llama al servicio de carpetas para obtener las carpetas de la sesión actual

Nombre de la Clase
Controlador Exportación
Descripción
Controlador que recibe las peticiones correspondientes a la exportación
Responsabilidades
- Recibir las peticiones GET correspondientes a la generación de actas. - Devolver los archivos que se generen listos para su descarga o los correspondientes errores.

- Devolver los archivos adjuntos a los correos que se pidan.
Atributos Propuestos
n/a
Métodos Propuestos
generarActa: Llama al generador correspondiente dados los datos adecuados y el tipo de archivo que se quiere generar.
obtenerAdjuntos: Obtiene los datos de los archivos adjuntos, llama al gestor de adjuntos y devuelve el archivo ZIP pedido.

Nombre de la Clase
Generador de Actas
Descripción
Clase que se encarga de la generación de actas en el formato adecuado
Responsabilidades
- Generar los archivos que se soliciten en el formato adecuado y con las opciones y correos pedidos.
Atributos Propuestos
n/a
Métodos Propuestos
generar: Llama al Servicio de Correo para obtener los datos necesarios (cuerpo de los correos, fechas...) y los imprime en el formato pedido (según esté implementado en sus subclases) siguiendo las opciones especificadas.

Nombre de la Clase
Generador de Actas PDF
Descripción
Clase que se encarga de la generación de actas en PDF
Responsabilidades
- Implementar las funcionalidades propias de la generación de acta en PDF
Atributos Propuestos
n/a
Métodos Propuestos
generarPDF: implementa la generación de un acta en PDF con los datos obtenidos de su clase padre.

Nombre de la Clase
Generador de Actas CSV
Descripción
Clase que se encarga de la generación de actas en CSV
Responsabilidades
- Implementar las funcionalidades propias de la generación de acta en CSV
Atributos Propuestos
n/a
Métodos Propuestos
generarCSV: implementa la generación de un acta en CSV con los datos obtenidos de su clase padre.

Nombre de la Clase
Servicio Correo
Descripción
Clase que se encarga de la conexión con el proveedor de servicios
Responsabilidades
- Conectarse con el proveedor de servicios para manejar las conexiones y/u obtener los correos.
Atributos Propuestos
datosDeConexión: Todos aquellos datos de conexión necesarios, tales como correo, contraseña, host...
Métodos Propuestos
abrirConexión: Abre una conexión con el proveedor de correos. obtenerCorreos: Llama al proveedor de servicios para obtener los correos pedidos y obtiene los datos que sean necesarios para nuestro producto.

Nombre de la Clase
Parser Correos
Descripción
Clase que se encarga de <i>parsear</i> los correos electrónicos recibidos del proveedor.
Responsabilidades
- Obtener los metadatos del proveedor y establecerlos en la entidad correo. - Obtener el cuerpo del correo independientemente de su formato
Atributos Propuestos
N/A
Métodos Propuestos
parse: Dado un correo del proveedor, devuelve un objeto de la entidad Correo, listo para ser tratado en el sistema.

Nombre de la Clase
Gestor de Adjuntos
Descripción
Clase que se encarga del empaquetado de los archivos adjuntos.
Responsabilidades
- Generar un ZIP con los archivos adjuntos de los correos.
Atributos Propuestos
N/A
Métodos Propuestos
generarZIP: Dada una lista de correos, genera un ZIP con los archivos adjuntos de este y el cuerpo de estos correos si es HTML o texto plano.

Nombre de la Clase
Gestor de Participantes
Descripción
Clase que se encarga de obtener estadísticas de los participantes.
Responsabilidades
- Identificar los participantes de los correos seleccionados.

- Calcular estadísticas de los participantes, p. ej., el número de correos enviados.
Atributos Propuestos
N/A
Métodos Propuestos
calcularParticipantes: Dada una lista de correos, devuelve las estadísticas de todos ellos.

Nombre de la Clase
Gestor de Opciones
Descripción
Clase que se encarga de la identificación de las diferentes opciones de exportación.
Responsabilidades
- Dada una petición, obtener que opciones de exportación se desean usar. - Indicar a los generadores de actas que opciones de exportación están activas.
Atributos Propuestos
opcionesEstablecidas: Almacena aquellas opciones que estén seleccionadas.
Métodos Propuestos
<< constructor >>: Dados los parámetros de una petición crea un gestor de opciones con las opciones pasadas como <i>String</i> directamente desde la petición. opcionEstablecida: Permite saber si una opción está establecida. Esto es de utilidad en la generación de actas.

5.6 Análisis de Casos de Uso y Escenarios

Una vez que hemos analizado las diferentes clases que tenemos en nuestra aplicación debemos volver atrás y recordar los casos de uso que especificamos en el apartado 5.3.3. Partiendo de ellos, deberemos especificar los pasos que se seguirían para completar el escenario principal, así como todos aquellos escenarios alternativos que pudiesen darse.

El objetivo de esta sección es dejar claro, de cara a la fase de diseño, cuáles son los pasos para una correcta ejecución los casos de uso y cómo deben responder a las diferentes casuísticas.

5.6.1 Caso de Uso 1 – Abrir conexión

Abrir Conexión	
Precondiciones	n/a
Poscondiciones	Habrà una conexión abierta con el proveedor de correo
Actores	Usuario conectado / Usuario no conectado
Descripción	El usuario: <ol style="list-style-type: none"> 1. Introducirà su email 2. Introducirà la contraseña de su email 3. Indicarà que quiere abrir una conexión 4. Recibirà una notificación de que la conexión se ha abierto
Variaciones (escenarios secundarios)	Podría introducirse un paso opcional entre el 2. y 3.: especificar el host al que se quiere conectar. Si no se especifica el sistema intentará seleccionar el adecuado automáticamente.
Excepciones	<ul style="list-style-type: none"> • El email es incorrecto: Puede no ser de uno de los proveedores soportados o quizás sí se soporta el proveedor, pero la cuenta no existe. <ul style="list-style-type: none"> ○ Notificar un error asociado al problema encontrado • La contraseña es incorrecta: Siendo el email válido, la contraseña puede ser incorrecta. <ul style="list-style-type: none"> ○ Notificar un error asociado al problema encontrado • La cuenta está bloqueada: No se puede acceder a la cuenta desde la aplicación, por estar bloqueado el acceso por parte del proveedor de correo. <ul style="list-style-type: none"> ○ Notificar un error asociado al problema encontrado
Notas	Todas las excepciones pueden producirse en el punto 3, volviendo todas al punto 1 en caso de producirse.

5.6.2 Caso de Uso 2 – Cambiar idioma

Cambiar idioma	
Precondiciones	n/a
Poscondiciones	Los textos de la aplicación aparecerán en el idioma seleccionado
Actores	Usuario conectado / Usuario no conectado
Descripción	El usuario:

	<ol style="list-style-type: none"> 1. Seleccionará el idioma en el que desee ver los textos 2. Los textos cambiarán del idioma actual al idioma seleccionado
Variaciones (escenarios secundarios)	<p>Escenario Alternativo 1: No se ha seleccionado ningún idioma previamente.</p> <ol style="list-style-type: none"> 1. Se usará el idioma por defecto del navegador o español si el idioma por defecto no está entre los soportados por la aplicación. 2. Se continuará por el paso 2 del escenario principal.
Excepciones	n/a
Notas	Los idiomas soportados serán español, asturiano e inglés

5.6.3 Caso de Uso 3 – Cerrar conexión

Reconectar	
Precondiciones	Debe haber una conexión abierta
Poscondiciones	No habrá ninguna conexión abierta
Actores	Usuario conectado
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 1. Realizará una acción que indique que desea cerrar la conexión 2. Recibirá una notificación de que la conexión se ha cerrado con éxito
Variaciones (escenarios secundarios)	n/a
Excepciones	<ul style="list-style-type: none"> • La conexión no pudo cerrarse: Por algún motivo, por ejemplo, desconexión con internet, la conexión con el proveedor de correo no puede cerrarse. <ul style="list-style-type: none"> ○ Notificar un error asociado al problema encontrado
Notas	

5.6.4 Caso de Uso 4 – Seleccionar correos

Seleccionar Correos	
Precondiciones	Debe haber una conexión abierta
Poscondiciones	
Actores	Usuario conectado
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 1. Irá a la parte de la interfaz en que se muestren los correos 2. Seleccionará aquellos con los que le interese generar un acta 3. Avanzará a la pantalla de exportación
Variaciones (escenarios secundarios)	n/a

Excepciones	<ul style="list-style-type: none"> • No se han seleccionado correos: El usuario desea realizar el paso 3, pero no hay correos seleccionados en el paso 2 <ul style="list-style-type: none"> ○ No se permitirá realizar el paso 3
Notas	

5.6.5 Caso de Uso 5 – Personalizar acta

Personalizar acta	
Precondiciones	Debe haberse avanzado a la pantalla de exportación
Poscondiciones	n/a
Actores	Usuario conectado
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 1. Especificará el formato en que quiere generar el acta (PDF/CSV) 2. Añadirá el número de campos personalizados que estime necesario. 3. Especificará los valores que le interesen para los campos personalizados añadidos (etiqueta y valor). 4. Seleccionará las opciones de exportación que le interesen (p. ej. mostrar fecha de los correos).
Variaciones (escenarios secundarios)	<p>Escenario alternativo 1: El usuario no añade campos personalizados en el paso 2.</p> <ul style="list-style-type: none"> • Se omite el paso 3 del escenario principal <p>Escenario alternativo 2: El usuario añade campos personalizados en el paso 2, pero no especifica valores para esos campos en el paso 3.</p> <ul style="list-style-type: none"> • Se imprimirán en el acta sólo los campos que estén activados, y si estuvieran activados y sin valor, se añadirá un campo en blanco.
Excepciones	n/a
Notas	Si el formato especificado en el paso 1 es CSV, se omiten los pasos 2 y 3.

5.6.6 Caso de Uso 6 – Generar PDF

Generar PDF	
Precondiciones	Debe haber una serie de correos seleccionados y una serie de opciones de exportación especificadas. También debe seleccionarse el formato PDF.
Poscondiciones	n/a
Actores	Usuario conectado
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 1. El usuario realizará una acción para especificar que desea generar el acta 2. El sistema devolverá el acta en formato PDF y se descargará desde el navegador al sistema

Variaciones (escenarios secundarios)	n/a
Excepciones	<ul style="list-style-type: none"> • Errores en la generación: Se producen errores generando el acta. <ul style="list-style-type: none"> ○ Se notificará un error asociado al problema encontrado
Notas	n/a

5.6.7 Caso de Uso 7 – Generar CSV

Generar CSV	
Precondiciones	Debe haber una serie de correos seleccionados y una serie de opciones de exportación especificadas. También debe especificarse que la generación ha de ser en formato CSV.
Poscondiciones	n/a
Actores	Usuario conectado
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 1. El usuario realizará una acción para especificar que desea generar el acta 2. El sistema devolverá el acta en formato CSV y se descargará desde el navegador al sistema
Variaciones (escenarios secundarios)	n/a
Excepciones	<ul style="list-style-type: none"> • Errores en la generación: Se producen errores generando el acta. <ul style="list-style-type: none"> ○ Se notificará un error asociado al problema encontrado
Notas	n/a

5.6.8 Caso de Uso 8 – Exportar Archivos Adjuntos

Exportar Archivos Adjuntos	
Precondiciones	Debe haber una serie de correos seleccionados y una serie de opciones de exportación especificadas. También debe especificarse que la generación ha de ser en formato CSV.
Poscondiciones	n/a
Actores	Usuario conectado
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 1. El usuario realizará una acción para especificar que desea exportar los archivos adjuntos. 2. El sistema devolverá los archivos adjuntos en formato ZIP y se descargará desde el navegador al sistema
Variaciones (escenarios secundarios)	n/a

Excepciones	<ul style="list-style-type: none"> • Errores en la generación: Se producen errores generando el archivo ZIP. <ul style="list-style-type: none"> ○ Se notificará un error asociado al problema encontrado
Notas	n/a

5.7 Análisis de Interfaces de Usuario

Si siguiésemos una metodología clásica, este sería el momento de empezar a crear las interfaces de usuario. Veríamos que elementos necesitamos, cuál es la mejor manera de distribuirlos para facilitar la experiencia del usuario y como debería comportarse en base a los diferentes eventos. Sin embargo, como estamos usando Scrum, no podemos afrontar el diseño de la interfaz completa al principio del desarrollo. En los apartados anteriores hemos incluido la que sería la “versión final” de los mismos, pues según hemos ido avanzando por los diferentes *sprints*, hemos ido modificando los diagramas, subapartados, requisitos... Sin embargo, para esta parte de las interfaces de usuario, tenemos que hacerlo de forma diferente. En Scrum, lo que se hace es un boceto por cada historia, que debería aprobar el *Product Owner*. De esta manera, iremos mostrando los diferentes bocetos para cada pantalla, divididos cada uno por su historia. Por tanto, pueden aparecer partes de la interfaz en las primeras historias, que luego no estén en el producto final y que sean modificadas por historias siguientes.

Otro aspecto a tener en cuenta es que al ser una SPA (Single-Page Application), es más correcto que hablemos de componentes que de pantallas. La idea es que estos interactúen entre sí, pero que la funcionalidad esté encapsulada dentro de ellos mismos.

5.7.1 Descripción de la Interfaz

Cada uno de los subapartados siguientes se corresponde con una historia de usuario. En ellos, se mostrará la interfaz planteada junto a una pequeña descripción de los elementos que aparecen y como interactuarían.

5.7.1.1 Historia H1 (Conexión con correo)

Este boceto se corresponde con el componente para la Historia H1.

The image shows a web browser window titled 'UNA'. Inside the window, there is a login form enclosed in a black border. The form contains two input fields: 'Correo' and 'Contraseña'. The 'Contraseña' field is masked with asterisks. Below the fields is a green button labeled 'Conectar'. To the right of the button, the text 'Éxito en la conexión' is displayed, indicating a successful login.

Figura 9 – Interfaz del componente de conexión para la historia H1

Este componente de la interfaz que se encarga de la conexión. La idea es que el usuario introduzca sus datos de conexión y clique conectar. Si se establece la conexión correctamente se mostrará a la derecha un mensaje de éxito y si no, uno de error.

5.7.1.2 Historia H2 (Desconectarse)

Ahora vamos con el boceto para la Historia H2.



UNA

Correo ejemplo@gmail.com

Contraseña *****

Conectar

Cerrar Conexión

Éxito en la conexión

Figura 10 – Interfaz del componente de conexión para la historia H2

Como puede verse es simplemente añadirle un botón a la anterior. La idea es que, una vez establecida la conexión, con ese botón pueda cerrarse. También se mostrarán los posibles mensajes de error o de éxito que se produzcan.

5.7.1.3 Historia H3 (Listar correos)

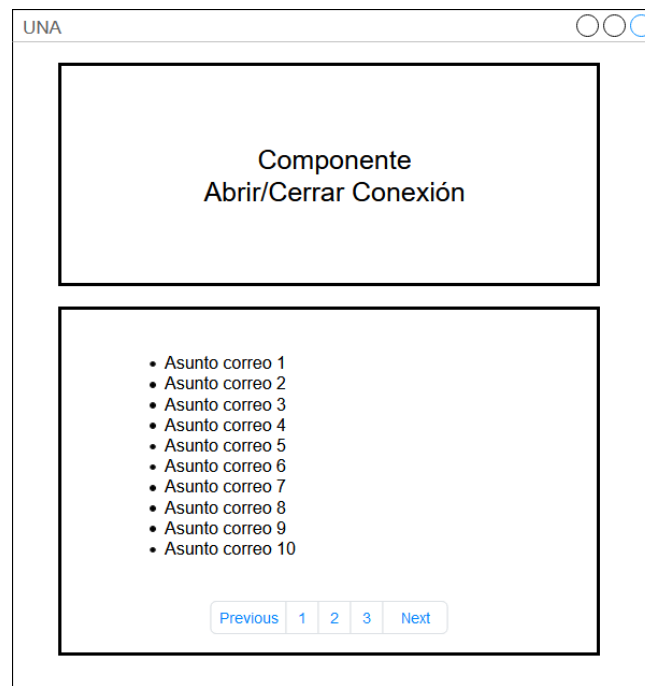
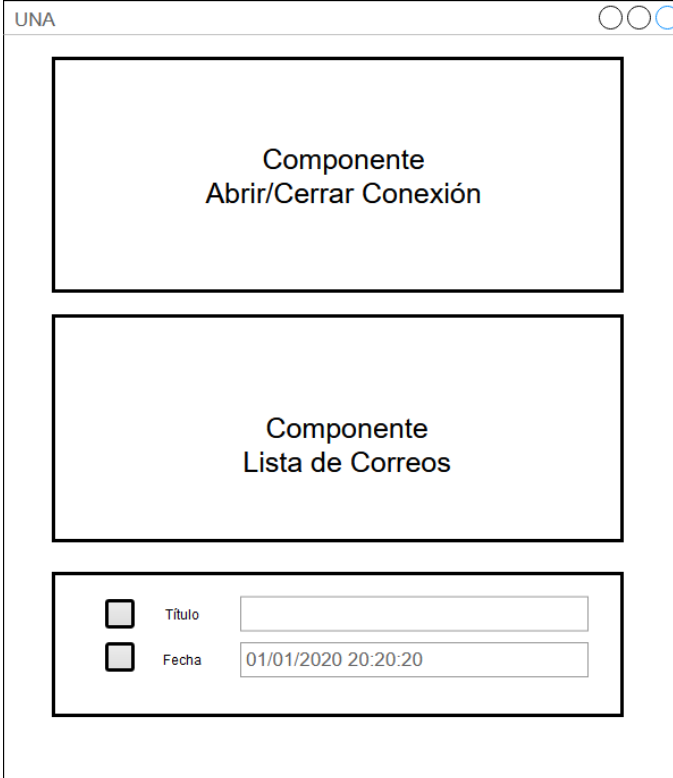


Figura 11 – Pantalla principal de la conexión con los cambios de la historia H3.

Para la Historia H3 se propone la siguiente interfaz de usuario. Se mostrará este componente debajo del de Abrir/Cerrar Conexión descrito en los dos apartados anteriores. Aquí se mostrará la lista de los correos paginada. Para cada uno de ellos se indicará el asunto del mismo.

5.7.1.4 Historia H4 (Añadir campos extra)



The screenshot shows a window titled "UNA" with three distinct components stacked vertically. The top component is a box labeled "Componente Abrir/Cerrar Conexión". The middle component is a box labeled "Componente Lista de Correos". The bottom component is a form with two rows. The first row has a checkbox, the label "Título", and an empty text input field. The second row has a checkbox, the label "Fecha", and a text input field containing the value "01/01/2020 20:20:20".

Figura 12 – Pantalla principal de la aplicación, tras la historia H4.

La Historia H4 se resuelve con una interfaz como esta. Un nuevo componente se añade debajo, con los campos título y fecha. Ambos pueden rellenarse y deberá marcarse la casilla de la izquierda si se quieren añadir.

5.7.1.5 Historia H5 (Generar PDF Simple)

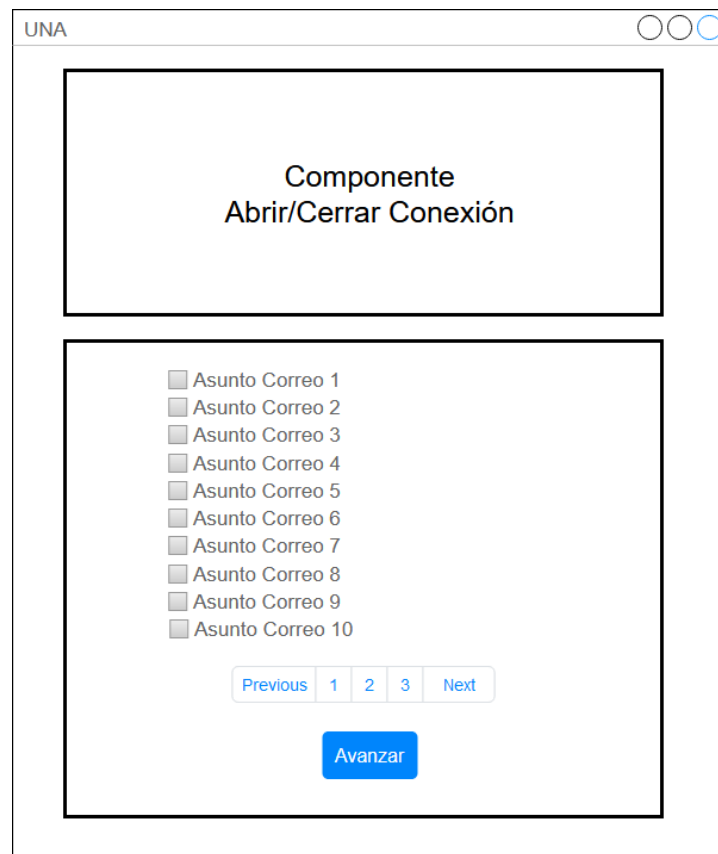


Figura 13 – Propuesta de interfaz para la pantalla principal en la historia H5.

UNA

Componente
Abrir/Cerrar Conexión

Título

Fecha

Añadir fechas de los correos al acta

Se generará un acta con los siguientes correos:

- Asunto correo 2
- Asunto correo 3
- Asunto correo 4

Figura 14 – Propuesta de interfaz inicial para el componente de exportación.

La Historia H5 es un poco más compleja que las anteriores, pues implica hacer cambios en las interfaces ya existentes. Primero se han añadido casillas a la lista de correos para poder seleccionar aquellos que se quieran exportar. También se ha añadido un botón que cambiará el componente de la lista de correos por otro nuevo, el que se ve en la Figura 14. Aquí movemos los campos de título y fecha, añadimos opciones de exportación y mostramos los correos seleccionados. Al clicar en el botón generar acta se descargará el PDF.

5.7.1.6 Historia H6 (Previsualizar contenido correos)

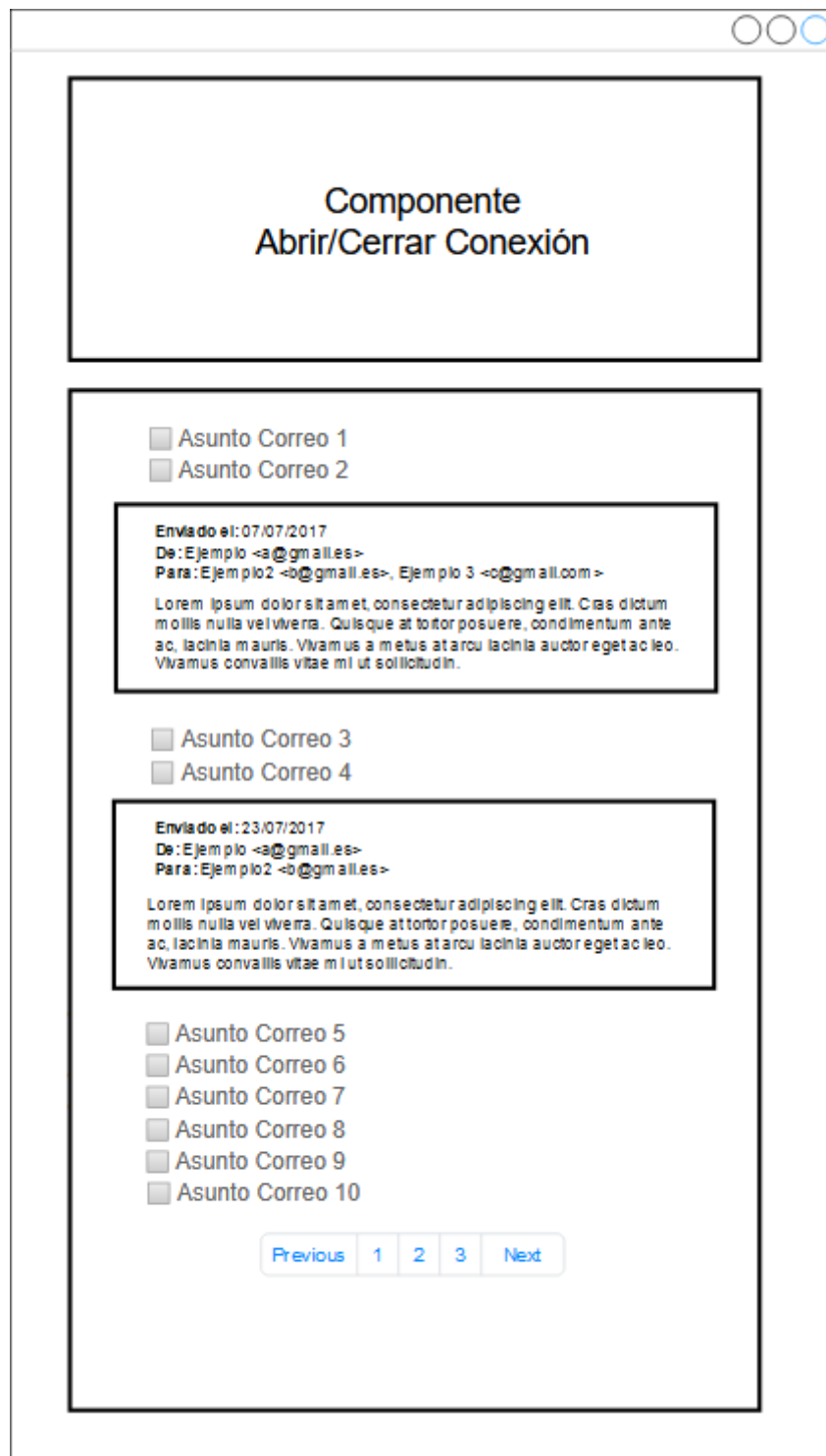


Figura 15 – Interfaz del componente de correos en la historia H6.

Para la Historia H7 modificamos el componente que lista los correos. Para ello tenemos que mostrar el cuerpo de los correos, la fecha de envío y las direcciones del destinatario y del remitente. Se hace de esta manera, mostrando una tarjeta con los datos, que se desplegará al clicar sobre el asunto del correo. Al volver a clicar sobre el asunto, la tarjeta se cerraría.

5.7.1.7 Historia H7 (Añadir campos extra personalizados)

Figura 16 – Modificaciones sobre el componente de exportación (Historia H7)

Aquí toca modificar la interfaz del componente de generación de actas. Para cumplir con la Historia H7 se propone la siguiente interfaz. Se han añadido una lista de *inputs* para poder incluir los nuevos campos personalizados. Cada fila contendrá dos de ellos, uno para la etiqueta y otro para el valor. Se incluirán también *checkboxes* que permitirán activar o desactivar el campo. De los campos que estén desactivados no podrán modificarse los datos, pero tampoco se borrarán, una vez activados podrá escribirse en ellos. Por defecto aparecerán dos campos activados, uno con la etiqueta “Título” y el valor vacío y otro con la etiqueta “Fecha” y la marca de tiempo del momento actual en formato “dd/mm/aaaa hh:mm:ss. Las etiquetas estarán internacionalizadas y se traducirán al idioma seleccionado por el usuario. Para añadir más campos deberá tocarse el botón de “Añadir Otro” y para eliminarlos el botón rojo que hay al final de cada fila. El botón rojo tendrá un icono de una papelera, que por restricciones de la herramienta no se ha podido incluir en el boceto. Los campos nuevos aparecerán debajo de los anteriores.

5.7.1.8 Historia H8 (Logos de la aplicación)

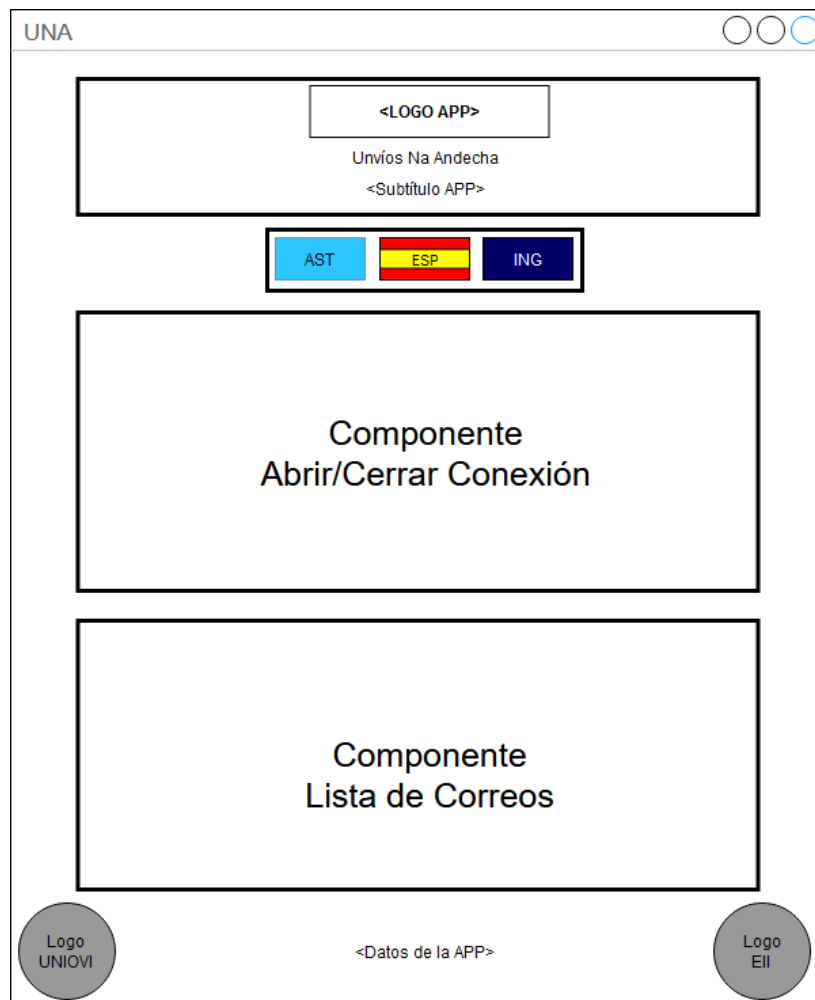


Figura 17 – Pantalla principal de la aplicación UNA (Historia H8)

La Historia H8 no afecta a ningún componente en concreto, si no al conjunto de la aplicación. La decisión principal que afecta a esta interfaz es el posicionamiento de los logos. Deben incluirse tres: el de la aplicación, el de la Universidad de Oviedo y el de la Escuela de Ingeniería Informática. La distribución es colocar arriba el logo de la aplicación, junto al título de la misma y un breve subtítulo que la describa. En la parte inferior deberán ir los otros dos logos corporativos, y entre ambos los datos de la aplicación (autor, Trabajo de Fin de Grado...). Para dispositivos móviles se eliminarán los logos corporativos del pie, dejando sólo los datos de la app. En el encabezado será todo lo contrario, eliminando los textos y dejando sólo el logo.

5.7.1.9 Historia H9 (Cachear correos)

Para la Historia H9 no se han realizado cambios en la interfaz.

5.7.1.10 Historia H10 (Navegación por páginas no secuencial)

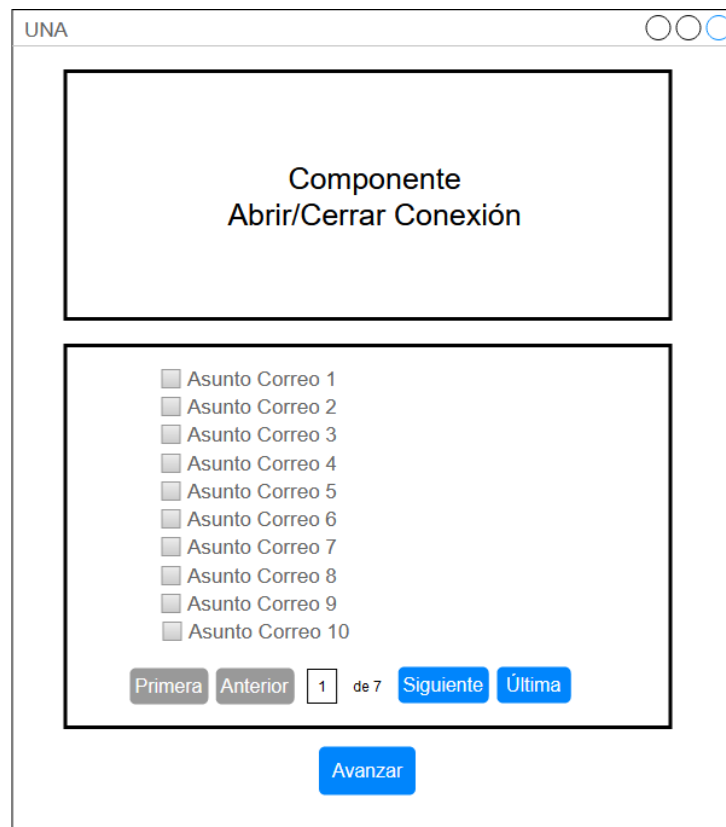


Figura 18 – Propuesta aceptada de interfaz para la paginación.

Hasta ahora la paginación de la lista de correos se hacía de manera rudimentaria, con un navegador básico. Sin embargo, la Historia H10 trae consigo la modificación de este. Para ello se añaden varios botones para la navegación y un cajetín para texto plano. En este *input* se podrá especificar el número de la página a la que se desea ir y al modificarlo automáticamente se cambiará de página. De manera auxiliar se añaden los botones “Siguiente” y “Anterior” que avanzarán o retrocederán una página y los botones “Primera” y “Última” que llevarán a ambos extremos de la lista de correos. Aquellos botones que correspondan a acciones no válidas, por ejemplo, estando en la página 1 ir a una página anterior, se encontrarán desactivados.

5.7.1.11 Historia H11 (Independencia de sesiones)

La Historia H11 no comporta cambios en la interfaz de usuario.

5.7.1.12 Historia H12 (Orden de correos)

La Historia H12 no implica modificaciones en la interfaz de usuario.

5.7.1.13 Historia H13 (i18n back-end)

La Historia H13 carece de modificaciones sobre la interfaz de usuario.

5.7.1.14 Historia H14 (Exportar a CSV)

Componente
Abrir/Cerrar Conexión

Seleccionar Formato:
PDF

Título

Fecha 01/01/2020 20:20:20

Nuevo Otros datos

Añadir Otro

Añadir fechas de los correos al acta

Se generará un acta con los siguientes correos:

- Asunto correo 2
- Asunto correo 3
- Asunto correo 4

Generar Acta

Figura 19 – Interfaz para la generación de actas PDF

Componente
Abrir/Cerrar Conexión

Seleccionar Formato:
CSV

Añadir fechas de los correos al acta

Se generará un acta con los siguientes correos:

- Asunto correo 2
- Asunto correo 3
- Asunto correo 4

Generar Acta

Figura 20 – Interfaz para la generación de actas CSV

Para esta historia se introduce un selector en la pantalla de generación de actas. Nos permitirá elegir el formato en el que generar el acta. En caso de que sea en PDF se mostrará como hasta ahora. Sin embargo, si es en CSV, no se mostrará la parte de crear los campos personalizados.

5.7.1.15 Historia H15 (Más proveedores de correo)



UNA

Correo

Contraseña

Host

Éxito en la conexión

Figura 21 – Componente de conexión renovado para la historia H15.

Aquí debemos introducir un nuevo botón que permita desplegar el campo “Host”. Este campo tenemos que añadirlo para permitir introducir el valor. Si la conexión es errónea, este campo debe de desplegarse automáticamente sin necesidad de tocar el botón.

5.7.1.16 Historia H16 (Gestionar múltiples carpetas)

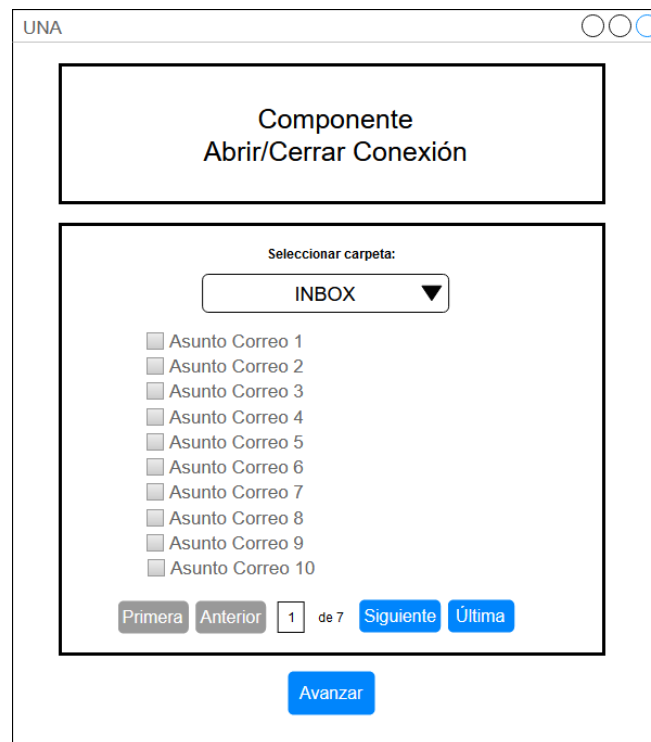


Figura 22 – Lista de correos con las modificaciones de la historia H16.

La modificación aquí es sencilla. Simplemente un nuevo selector de carpetas, que nos permita navegar entre las diferentes que haya. Cuando se detecte un cambio se procederá a la actualización de los correos con los procedentes de la nueva carpeta seleccionada.

5.7.1.17 Historia H17 (Aviso de cookies)

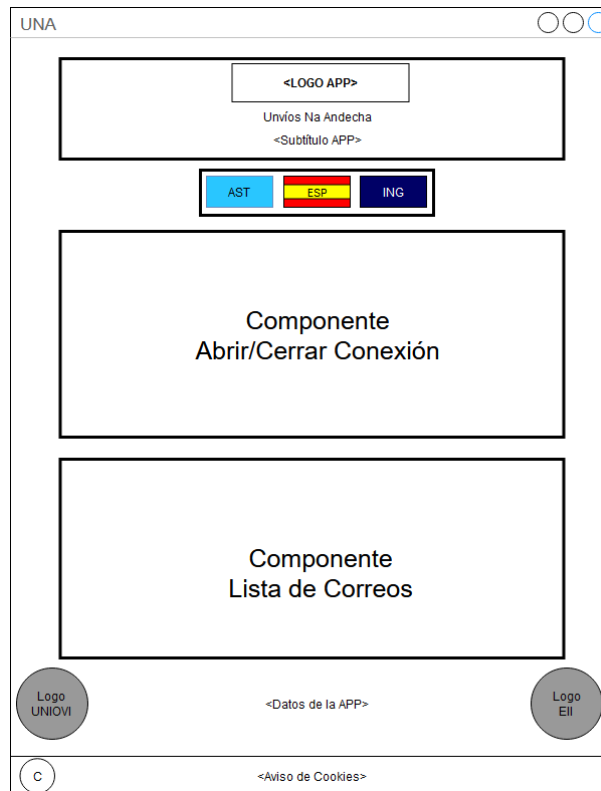


Figura 23 – Propuesta de interfaz para el aviso de cookies.

El aviso de cookies se mostrará en la parte de debajo de la página. Contendrá un botón que permita ocultarlo o mostrarlo (En el dibujo el círculo con la letra C). Este botón aparecerá en todas las páginas de la aplicación.

5.7.1.18 Historia H18 (Certificado digital HTTPS)

La Historia H18 no modifica la interfaz de usuario.

5.7.1.19 Historia H19 (Advertencia correos Google)



Figura 24 – Componente de conexión con los iconos de la historia H19.

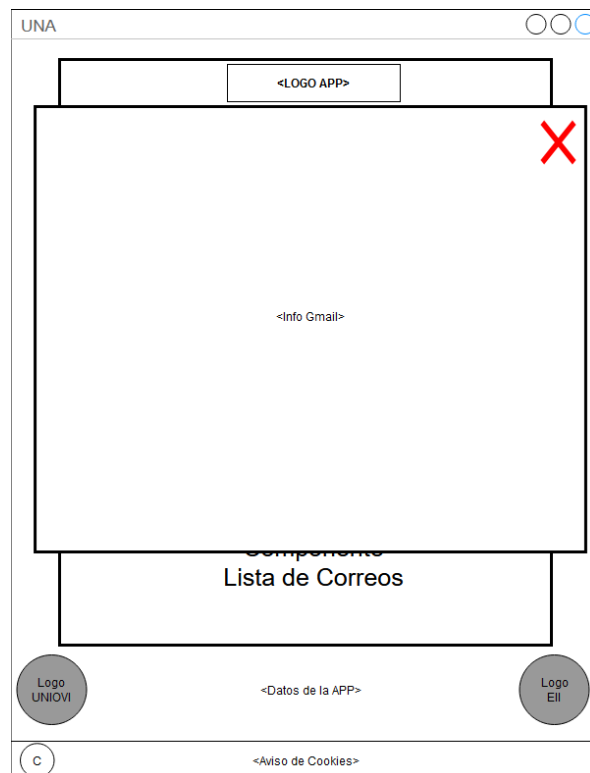


Figura 25 – Posicionamiento del aviso de la historia H19.

Aquí se introduce un nuevo icono en el componente de conexión. Al clicar sobre él se mostrará un *popup* modal sobre la aplicación con la información indicada.

5.7.1.20 Historia H20 (Mostrar adjuntos)

UNA

Componente
Abrir/Cerrar Conexión

Seleccionar Formato:
PDF

Título

Fecha 01/01/2020 20:20:20

Nuevo Otros datos

Añadir fechas de los correos al acta

Se generará un acta con los siguientes correos:

- Asunto correo 2
- Asunto correo 3
- Asunto correo 4

Figura 26 – Distribución de botones para la historia H20.

Para descargar los archivos adjuntos se incluye un nuevo botón que comenzará el proceso de descarga.

5.7.1.21 Historia H21 (Resumir participantes conversación)

La Historia H21 no modifica la interfaz de usuario.

5.7.1.22 Historia H22 (Botón atrás en la exportación)

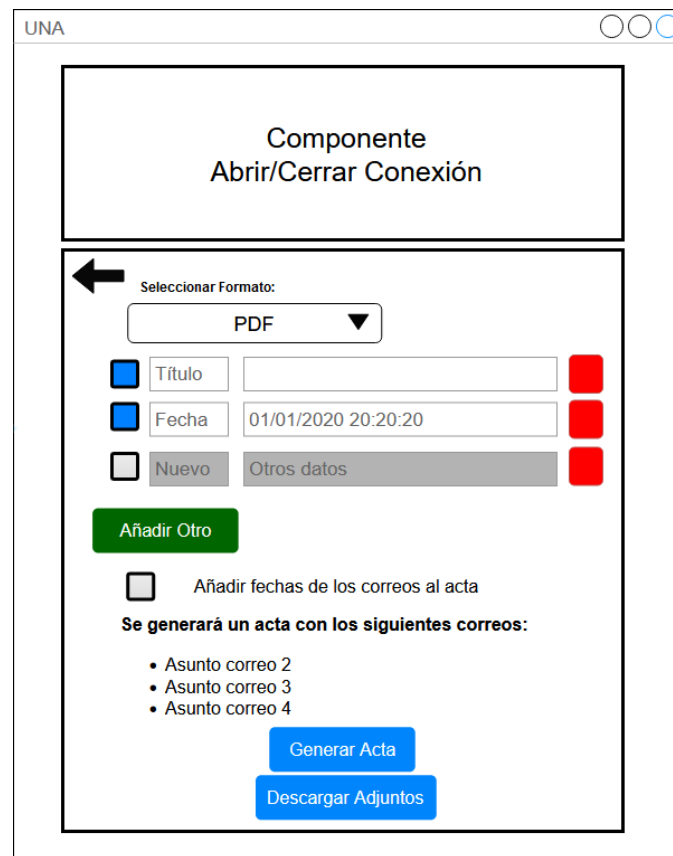


Figura 27 – Posición del botón de “vuelta atrás” para la historia H22.

La Historia H22 simplemente añade un botón de retroceso en el componente de generación de actas.

5.7.1.23 Historia H23 (Mostrar correos en formato HTML)

Esta historia no produce cambios en la interfaz de usuario.

5.7.1.24 Historia H24 (Añadir campo CC)

Esta historia no produce cambios en la interfaz de usuario.

5.7.1.25 Historia H25 (Añadir estadísticas participantes)

Esta historia no produce cambios en la interfaz de usuario.

5.7.1.26 Historia H26 (Desplegar en Docker)

Esta historia no produce cambios en la interfaz de usuario.

5.7.2 Descripción del Comportamiento de la Interfaz

Una vez vistas las pantallas que se realizarán, es momento de dejar claras algunas convenciones y establecer algunos principios de cara al diseño. Aquí se trata de definir cómo se va a dar solución a diferentes aspectos generales, que se nos presentarán y aparecerán en las interfaces correspondientes a diferentes historias de usuario.

Una de las cosas que se tienen que definir es como se realizará la validación de los datos que introduzca el usuario. Aquí una aproximación clásica es lo mejor. Se hará en el *front-end*, para mejorar la experiencia de usuario, y en el *back-end* por seguridad, como recomiendan las buenas prácticas.

Los posibles mensajes de error se mostrarán de dos maneras. Si son cosas que un usuario pueda entender, por pantalla, internacionalizados y en el idioma propio de este. Si son cosas más técnicas se debería mostrar un mensaje genérico, y en tal caso, mostrar en la consola del navegador los detalles técnicos. Un elemento que se suele tener en cuenta es la pantalla de error que se produce cuando se introduce una ruta inexistente dentro de la aplicación. En nuestro caso es poco probable que esto suceda, pues no manejamos rutas, y si alguien la introdujese sería seguramente por motivos maliciosos. En este caso, la mejor estrategia es dar la menor información posible. Una pantalla sencilla, que indique “Error 404” es la mejor solución.

La ayuda que se proporcionará al usuario intentaremos que sea mínima, en cuanto a manuales o grandes textos detallados. Si tuviésemos que proporcionarlos estaríamos fracasando en uno de los objetivos del proyecto, que es tener una interfaz intuitiva. Para la navegación principal habrá que poner especial cuidado en los textos elegidos, deben quedar muy claros y estar en el idioma del usuario. Son preferibles los textos a los símbolos, pues estos pueden tener interpretaciones múltiples (W3C, Images). En el caso de que se tengan que utilizar, por exigencias de tamaño u otras causas, se incluirán siempre *tooltips* indicando su función. Se podría tener que llegar a proporcionar instrucciones detalladas, pero en principio sería para funcionalidades externas, por ejemplo: ¿cómo configurar tu correo electrónico para usar nuestra aplicación?

Otro de los aspectos a tener en cuenta entre interfaces es que todas ellas deben verse adecuadamente, aunque haya reducciones de pantalla, lo que se conoce como diseño web adaptable. Aunque es una aplicación que suele utilizarse más desde un ordenador, por tratar con aspectos de ofimática y profesionales, habrá usuarios que la utilicen desde teléfono móvil.

Por último, un principio menos concreto que debemos tener en cuenta es que el objetivo de esta aplicación es proporcionar una solución flexible al usuario, por tanto, se preferirá optar por una mayor personalización antes que comprobaciones de tipos de dato. Por ejemplo, si el usuario quiere introducir una fecha para su acta, le ofreceremos la posibilidad de introducir, la fecha “22/07/2020”, aunque no deberíamos prohibirle poner el literal “22 de julio por la mañana”.

5.8 Especificación del Plan de Pruebas

Una de las partes más importantes, y a la vez más olvidadas, del desarrollo del software son las pruebas. En ellas tenemos que asegurarnos que hemos construido el producto de forma correcta, sin errores, y que además cumple la función para la que está diseñado. Para esta delicada tarea, donde nos toca no tener piedad con nuestro código y buscarle el mayor número de fallos posible, es importante seguir un orden, siendo metódicos y consistentes. Para ese proceso, hemos dividido esta documentación en varias partes. En este capítulo, que es la parte de análisis, describiremos a grandes rasgos todo aquello que queremos probar y cómo pretendemos hacerlo.

5.8.1 Pruebas Unitarias

En el mundo del *testing* hay un concepto que es el de la Pirámide de Pruebas. No está clara su autoría, pero es una explicación ideal de cómo deberían distribuirse las pruebas en los proyectos de software, basándose en el grado de automatización y su tipo. La susodicha pirámide es la de la Figura 28 e intenta transmitir la idea de que en un proyecto software debemos tener una gran cantidad de pruebas unitarias automatizadas, abundantes pruebas de integración automatizadas y algunas que otras de interfaz gráfica, prefiriendo que estas sean automáticas también [13]. Esta distribución es ideal... y teórica, pues muy frecuentemente se invierte y acabamos haciendo las pruebas a mano. En algunas ocasiones, tampoco se destinan los recursos necesarios para realizar las pruebas de manera exhaustiva.

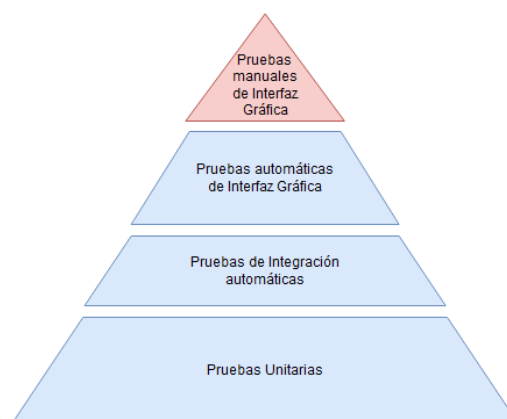


Figura 28. Pirámide de 'Testing'

Sin embargo, aplicar estos conceptos nos pone en un buen camino hacia la detección de errores. Para aplicar todo esto de manera eficiente a nuestro proyecto, debemos considerar que, al usar una metodología ágil, las pruebas unitarias aumentan más si cabe su importancia. Por esto, deberemos hacer pruebas unitarias tanto en el *back-end* como en el *front-end*, intentando lograr la mayor cobertura posible. Para ello se usarán *Mocks* y *Spies* que permitan aislar los métodos del resto de clases con las que interactúan.

5.8.2 Pruebas de Integración

Con este tipo de pruebas intentaremos encontrar defectos en las llamadas a la API. Estas pruebas tendrán un componente que simule un proveedor de correo, y realizaremos llamadas de alto nivel directamente a los controladores, que trabajarán con el resto de componentes. De esta manera probaremos cómo funciona toda la estructura en conjunto. No destinaremos a ellas tantos recursos como a las unitarias, pero aun así automatizaremos un buen número de funcionalidades clave, aquellas que tienen muchas entradas posibles, por ejemplo, la llamada a “abrir conexión”. Las que no estén automatizadas se realizarán manualmente utilizando una herramienta que nos permita realizar peticiones GET y POST.

Para estas pruebas debemos comprobar los mensajes de error, además de las respuestas aceptables, pues no podemos olvidarnos que la API será *front-end* de cierto tipo de usuarios.

5.8.3 Pruebas de las Historias de Usuario

Si las pruebas unitarias buscan probar que los componentes funcionan correctamente y las de integración que funcionan bien entre sí, estas otras pruebas, a partir de las historias de usuario pretenden ver si a muy alto nivel, es decir, una vez frente a la interfaz, los incrementos realizados cumplen con lo que se espera de ellos.

Estas pruebas se enmarcan dentro de los *sprints* y se diseñan justo después de plantear las historias de usuario, de manera que no estén influenciadas por el código desarrollado. Se ejecutan antes de dar por completada la historia. Esto se hace de manera manual sobre la interfaz final desarrollada y nos permite probar de primera mano no sólo la funcionalidad, sino también la experiencia, la apariencia y ese tipo de aspectos difíciles de evaluar por medio de las máquinas.

5.8.4 Pruebas Funcionales

Estas pruebas las realizaremos sobre la interfaz gráfica de la aplicación web. Se simularán acciones de usuario utilizando Selenium, una herramienta que permite interactuar programáticamente con un navegador web.

El código de estas pruebas será independiente de la aplicación y se irá incrementando durante el desarrollo. El objetivo es tener unos *tests* que puedan ejecutarse rápida y automáticamente y nos permitan validar de manera ágil que la aplicación funciona correctamente.

5.8.5 Pruebas de Rendimiento

En algunas funcionalidades clave es necesario medir el tiempo que se tardan en realizar una serie de acciones. Por ejemplo, el tiempo en recuperar una información de un sistema externo o en realizar alguna operación. Realizar pruebas de rendimiento nos permite detectar en que puntos es asumible y en cuales debemos pensar en mejorar el funcionamiento de los algoritmos o en su defecto facilitar información más detallada al usuario. Estas pruebas se realizarán sobre el producto final, como forma de validación de los tiempos de carga.

5.8.6 Pruebas de Accesibilidad

Las pruebas de accesibilidad permiten comprobar que la aplicación sigue unos principios que facilitan su uso, tanto para el público habitual como para personas que presenten algún tipo de discapacidad. La ejecución de estas pruebas es sencilla, y además de ser altamente recomendable para las webs que se diseñan, la obligatoriedad de realizarlas se recoge en los
Tabla 15 – Requisitos funcionales del tercer sprint

Requisitos No Funcionales de la aplicación. La validación se hará por medio de la combinación de herramientas automáticas, como las que contienen los navegadores y de la revisión manual de *checklists* oficiales que especifican los principios que se deben seguir.

5.8.7 Pruebas de Usabilidad

Uno de los objetivos del proyecto definidos en los capítulos de esta documentación es la facilidad de uso de las interfaces, como punto clave en la elección de UNA frente a sus posibles competidores. Para probar esto, nada mejor que los usuarios finales. Por tanto, deberemos organizar pruebas de usabilidad, en las que estos realicen una serie de acciones y nos cuenten su experiencia de uso.

Lo ideal sería realizar unas pruebas de usabilidad al final de cada *sprint*, pero otra vez por motivos de tiempo y recursos nos resulta imposible. Las realizaremos únicamente al final del desarrollo, y en base a los resultados que obtengamos, aplicaremos las mejoras que sean oportunas, para mejorar la usabilidad del producto.

Capítulo 6. Diseño del Sistema

Una vez hecho el análisis y con las ideas bien claras de qué se va a construir, pasamos a la parte de diseño. Aquí ya se pueden ver cosas muy parecidas al producto software.

6.1 Arquitectura del Sistema

El primer punto de este capítulo será incluir algunos diagramas que muestren desde varias perspectivas como interactúan los elementos de la aplicación.

6.1.1 Diagramas de Paquetes

Partiendo del diagrama de clases, podemos crear la estructura de paquetes de la aplicación. Incluye cosas que no estaban en el diagrama de clases, como aquellos paquetes específicos de implementación, que no se consideran a alto nivel, pero sí una vez que profundizamos. Vamos a hablar primero del *front-end* cuyo diagrama de paquetes es este.

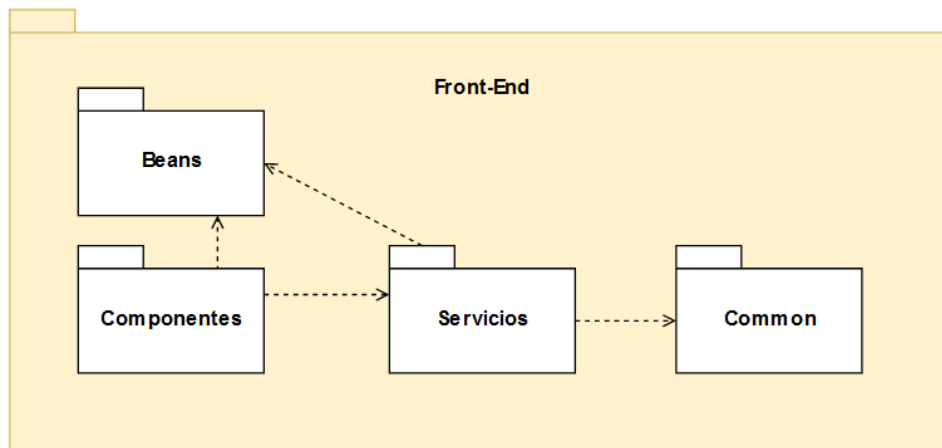


Figura 29 – Diagrama de paquetes del front-end

Y después pasaremos a explicar los componentes del *back-end* que se representan en este otro esquema:

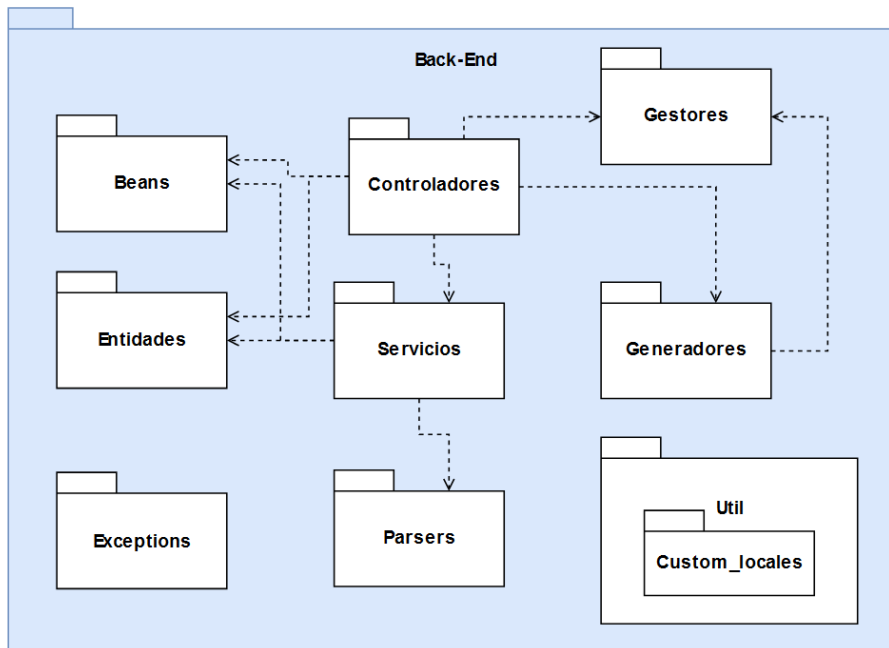


Figura 30 – Diagrama de paquetes del back-end

Beans (Front)

Este paquete contiene una serie de clases que nos permiten transmitir los datos entre diferentes partes de la aplicación, así como realizar llamadas HTTP con ellos. Siguen el clásico patrón de diseño DTO.

Componentes (Front)

Incluye las clases que se encargan de manejar las acciones del usuario y el funcionamiento de los diferentes elementos de la interfaz. Tienen poca carga lógica y realizan muchas llamadas a los servicios.

Servicios (Front)

Contiene las clases que se encargan de realizar las llamadas al *back-end* vía HTTP y el resolutor de dominios. Aquí también irían las clases que tuvieran carga lógica, si las tuviéramos en algún desarrollo futuro.

Common (Front)

Contiene aquellas clases comunes a los servicios y componentes. Sería como el típico paquete `Util`, métodos muy específicos, para funcionalidades fuera de la lógica de negocio de la aplicación. En esta aplicación se usa poco, simplemente permite la centralización de la gestión de constantes comunes a los servicios.

Beans (Back)

Las clases de este paquete, al igual que en su homónimo del *front-end*, contienen una serie de clases que siguen el patrón DTO. Su objetivo es proporcionar unas clases sencillas para su traspaso a través de la red. La mayoría de los atributos son tipos simples (int, string...).

Entidades (Back)

Las entidades son más complejas que los Beans. Estas sí contienen carga lógica, que facilita los procesos de la aplicación. Se transforman en Beans para su envío por la red.

Controladores (Back)

Son la entrada a la API y reciben las llamadas a ella. Se encargan de recibir las peticiones, llamar a los métodos adecuados del resto de componentes y devolver las respuestas.

Servicios (Back)

Aquí se incluirían los servicios, las clases principales que contendrían la lógica de negocio. Sólo se usa una, el Servicio de Correos, que realiza la conexión con el proveedor de correo, mantiene los datos que de allí obtiene y realiza los procesos llamando a los métodos oportunos de otras clases cuando sea necesario.

Parsers (Back)

Contiene clases que usan los servicios y que se encargan de transformar datos del proveedor para su uso en la aplicación. En esta versión sólo hay una, la que trabaja con los correos y sus metadatos.

Generadores (Back)

Se incluyen aquí los generadores de actas, que se encargan de componerlas y devolver los archivos pedidos. Hay un generador por formato.

Gestores (Back)

Las clases de este paquete no tienen una temática común, pero sí un funcionamiento similar. Se encargan de recibir unos datos, procesarlos y ponerlos a disposición de la clase que los llame. Por ejemplo, el gestor de participantes recibe una serie de correos, obtiene de ellos los participantes y sus estadísticas y ofrece unos métodos sencillos con los que otras clases pueden consultar los resultados sin dificultad.

Util (Back)

Contiene una serie de clases, que pueden ser útiles en algunos momentos. Por ejemplo, un *wrapper* para las respuestas HTTP o una utilidad que permite facilitar el proceso de generación de un CSV, colocando los separadores en los lugares correctos.

Custom_locales (Back)

Localizado dentro del paquete `Util`, contiene las clases para resolver fechas del *locale* para el idioma asturiano, que no está incluido dentro del paquete por defecto de Java. Si hubiese que incluir otros idiomas no soportados irían aquí.

Excepciones (Back)

Aquí van las implementaciones de la clase `Exception` que nos permiten un tratamiento más fino de los errores que puedan producirse, devolviendo mensajes de error más significativos.

6.1.2 Diagramas de Componentes

Mientras que en el diagrama de paquetes mostrábamos una agrupación por temáticas/funcionalidad de las clases, en el diagrama de componentes que vamos a mostrar aquí, nos centraremos en qué interactúa con qué. Los componentes que aquí aparecen no tienen una traducción exacta desde el diagrama de paquetes, ya que algunos aportan más de un componente y otros se han suprimido para no ensuciar el diagrama con relaciones que poco aportan a la hora de comprender el sistema.

6.1.2.1 Front-end

Para el *front-end* se proporciona el siguiente diagrama:

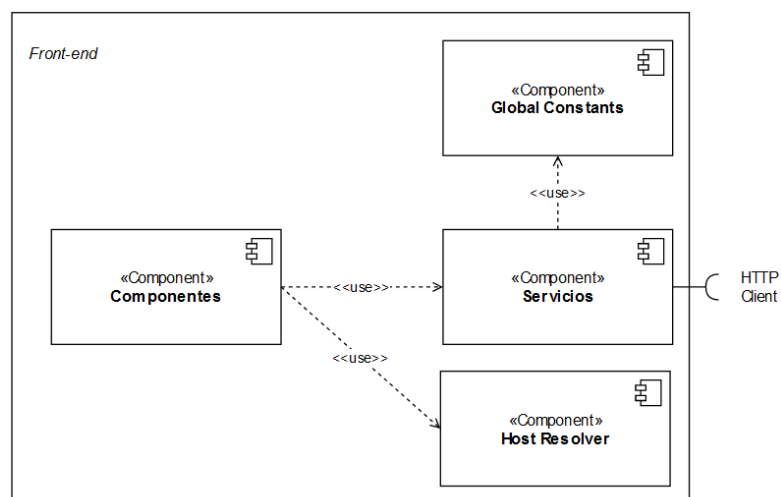


Figura 31 – Diagrama de componentes del front-end

El funcionamiento es sencillo. Los componentes usan los servicios y estos a su vez, realizan llamadas a través de un cliente HTTP. Como añadido, el `HostResolver`, traduce los nombres que le proporcionan los componentes. Además, el módulo `GlobalConstants`, es usado por los servicios para consultarlas.

6.1.2.2 Back-end

Para el *back-end* de la aplicación se propone el siguiente esquema.

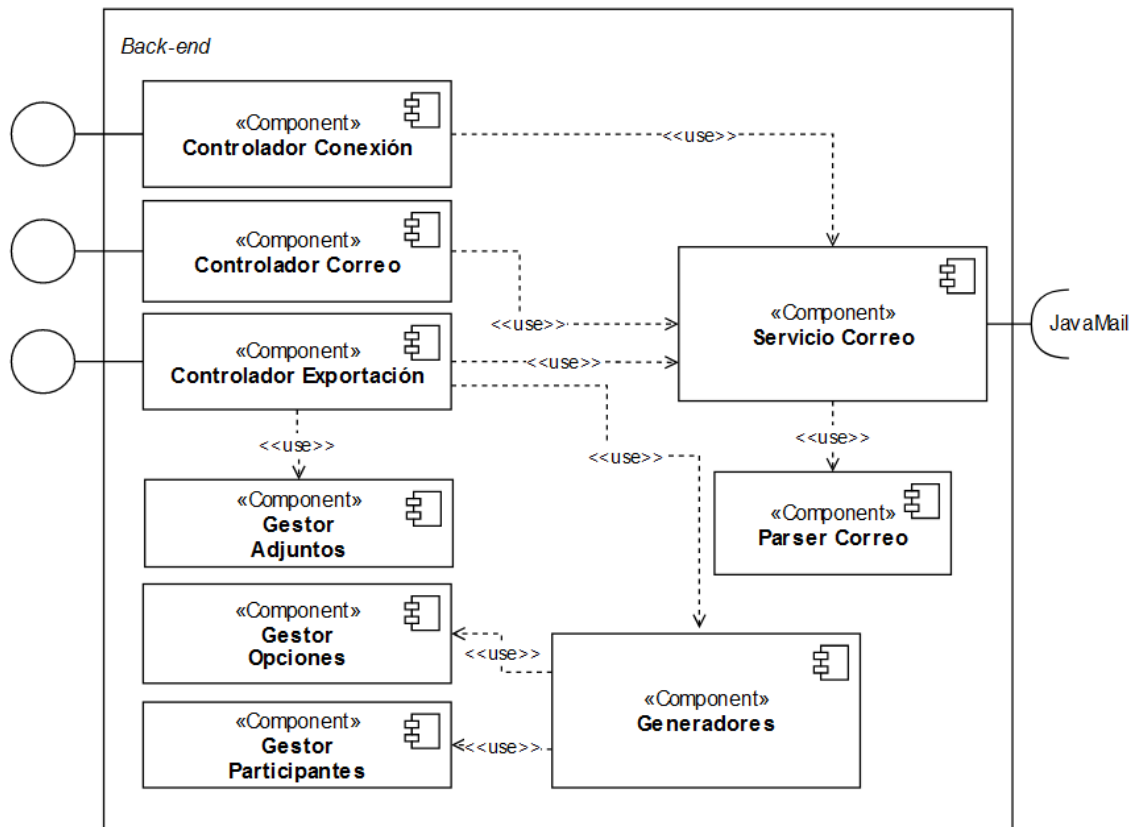


Figura 32 – Diagrama de componentes del back-end

Se han desgranado los controladores y los gestores, para una vista más precisa. Como vemos, los primeros proporcionan una interfaz, donde reciben las peticiones HTTP del exterior. El controlador de la conexión se comunica con el servicio de correo para abrirla y cerrarla. El controlador de correo consultará en el servicio los datos de la conexión abierta. El Servicio obtiene estos datos del proveedor usando JavaMail y los pasa a un formato manejable usando el MailParser. El último de los controladores, el de exportación, obtendrá los datos del Servicio y se los pasará al generador adecuado, para que este componga el acta, usando los gestores de opciones y participantes. Este controlador, llamará en su lugar al gestor de adjuntos si lo que se desea es exportar los archivos adjuntos, en lugar de un acta.

6.1.3 Diagramas de Despliegue

Una vez construida la aplicación es necesario desplegarla para el acceso público. Para explicar esto usaremos un diagrama de despliegue. Este que se puede ver a continuación:

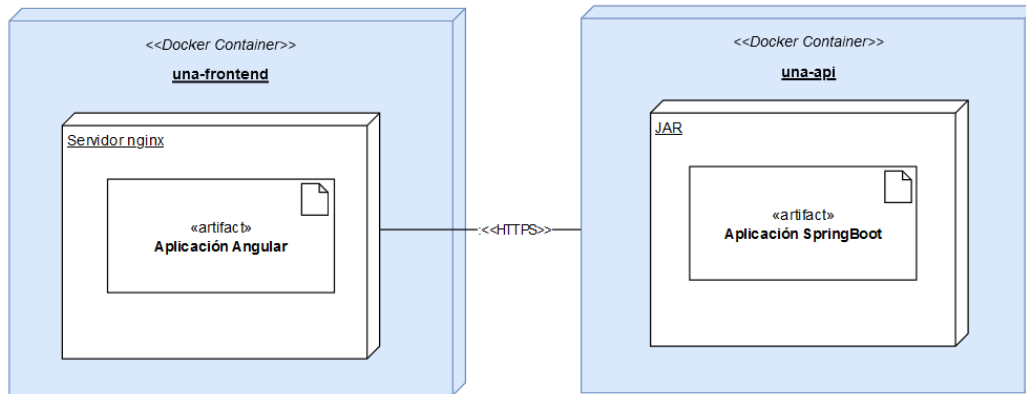


Figura 33 – Diagrama de despliegue de la aplicación

La aplicación se desplegará usando Docker Compose, que manejará dos contenedores, uno para el *front-end* y otro para la API. Vamos a ver las características de cada uno.

6.1.3.1 Front-end

La aplicación Angular se desplegará en un servidor *Nginx*, que expondrá el puerto 80 para el acceso público por HTTP.

6.1.3.2 Back-end

Por su parte, la aplicación SpringBoot se comprimirá en un JAR, que se ejecutará en un contenedor de Docker y que expondrá públicamente el puerto 8080. Aquí realizarán las peticiones tanto los usuarios finales que usen la API, como el *front-end* para la obtención de datos.

6.2 Diseño de Clases

En las próximas líneas será el momento de indicar qué clases forman parte del sistema final. Esta parte de la documentación se basa en su correspondiente en el análisis, en concreto, en el punto 5.5 - Diagrama de Clases Preliminar del Análisis.

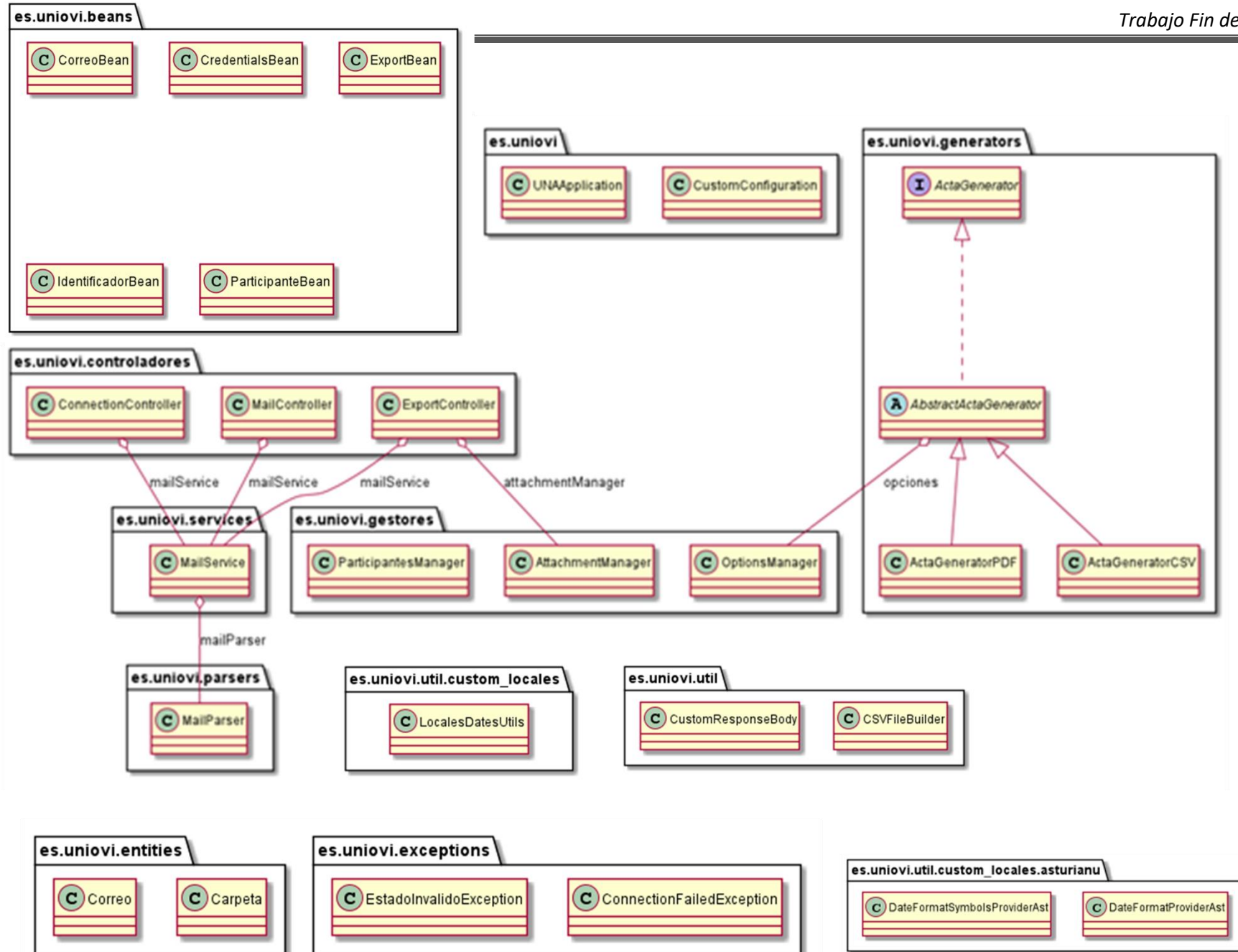
6.2.1 Diagrama de Clases

El diagrama que aquí se va a mostrar contiene todas las clases programadas en el desarrollo de la aplicación. No se incluyen aquellas clases propias de los *frameworks* usados, por ejemplo, las de configuración de SpringBoot. Tampoco se incluyen los *plugins* externos, ni las clases usadas para pruebas.

Siguiendo el mismo criterio que en el resto de la documentación, mostraremos por separado los diagramas de clases del *back-end* de los del *front-end*.

6.2.1.1 *Back-end*

Para esta parte del sistema mostraremos primero el diagrama de clases general y después iremos desgranando uno a uno los diferentes paquetes. Por motivos de legibilidad, el diagrama general no contiene los métodos ni los atributos de las clases representadas, estos se incluyen en los diagramas específicos de cada paquete. Pese a que sean específicos de un paquete, también incluyen las relaciones con clases de otros, aunque estas no mostrarán los métodos.



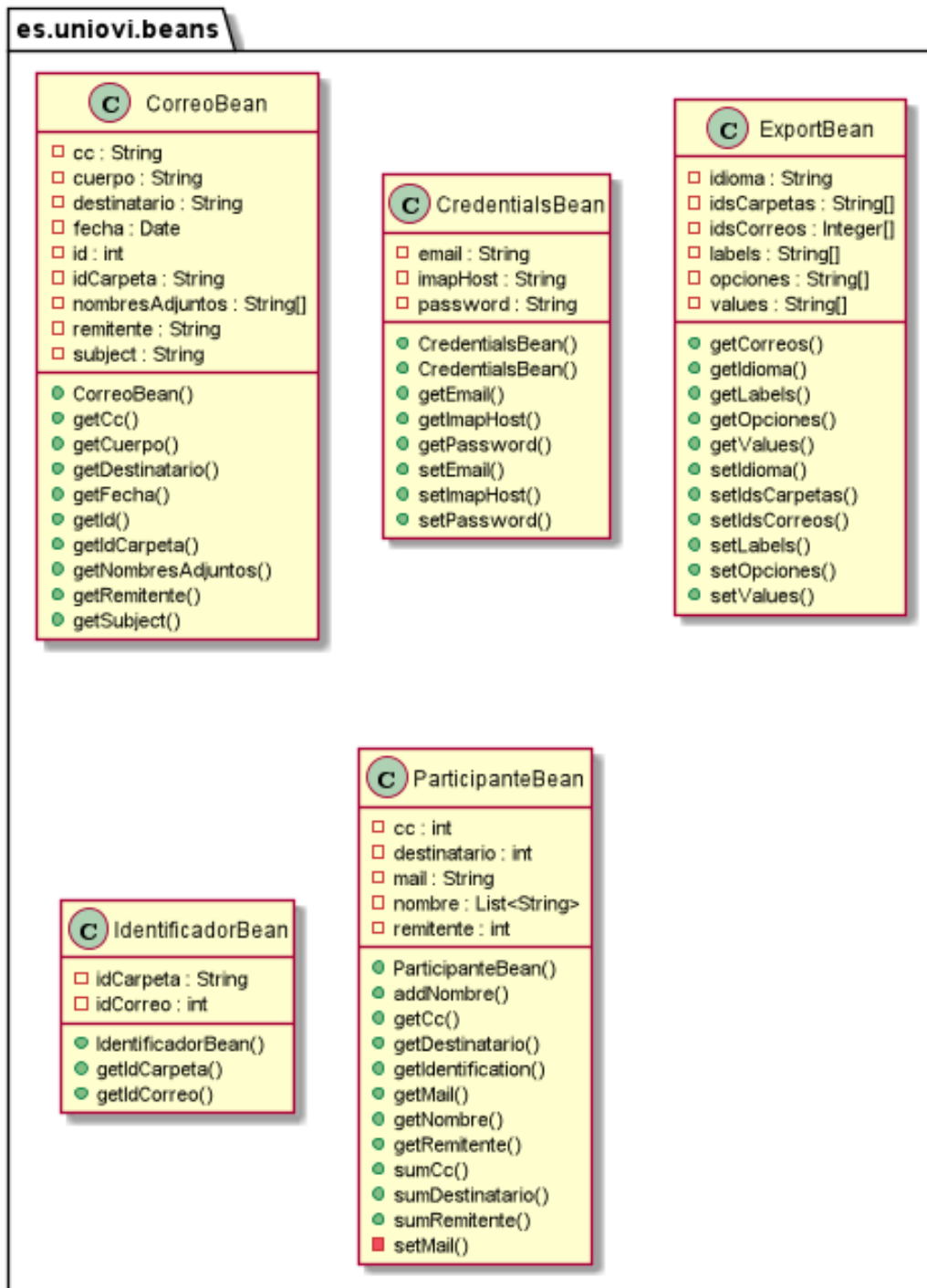


Figura 35 – Diagrama de clases (paquete beans)

El paquete `beans` contiene una serie de clases que almacenan datos. Todos tienen tipos de datos sencillos: `int`, `String`, `Array`, `Date` o `List`. Fácilmente serializables para su envío por la red. También contienen una serie de accesores (*getters*) y mutadores (*setters*).

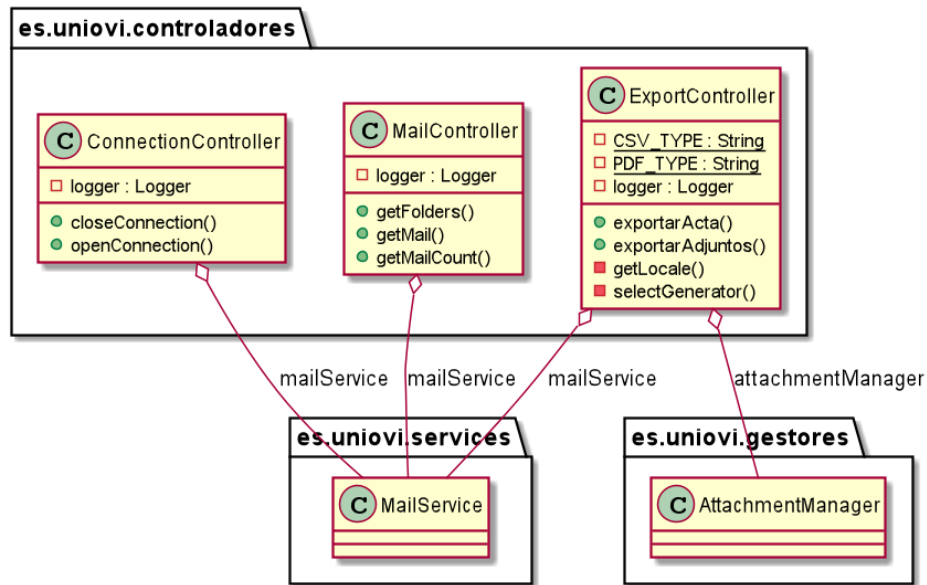


Figura 36 – Diagrama de clases (paquete controladores)

Los controladores tienen una serie de métodos que se corresponden con las llamadas permitidas a la API. También aparece algún método privado. Todos ellos incluyen un MailService al que realizan peticiones. Además, el ExportController exporta los archivos adjuntos usando el AttachmentManager.

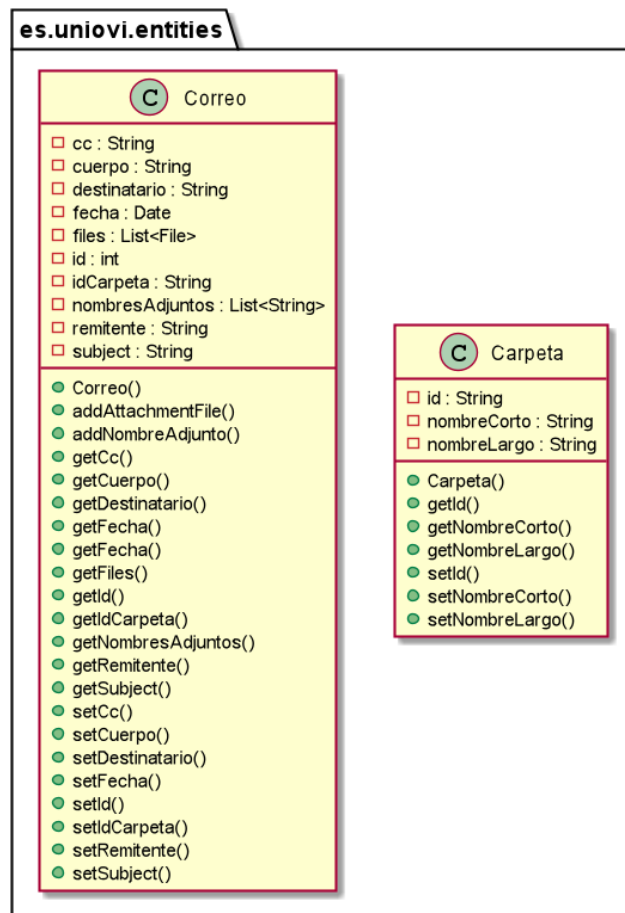


Figura 37 – Diagrama de clases (paquete entities)

Hay dos entidades en este sistema. Una de ellas es `Correo` y otra `Carpeta`. Estas se usan en el sistema para mapear objetos del proveedor de servicios. No tienen más que mutadores y accesoros a una serie de atributos.

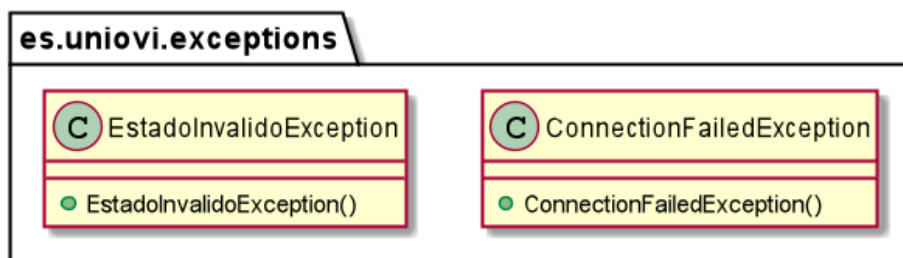


Figura 38 – Diagrama de clases (paquete exceptions)

Las excepciones personalizadas son de dos tipos. Uno de ellos es `EstadoInvalidoException`, que se mostrará principalmente cuando se intente realizar una acción no permitida por la situación actual de la conexión, por ejemplo, obtener una serie de correos con la conexión cerrada. El otro tipo es `ConnectionFailedException`, que saltará cuando falle el intento de abrir la conexión.

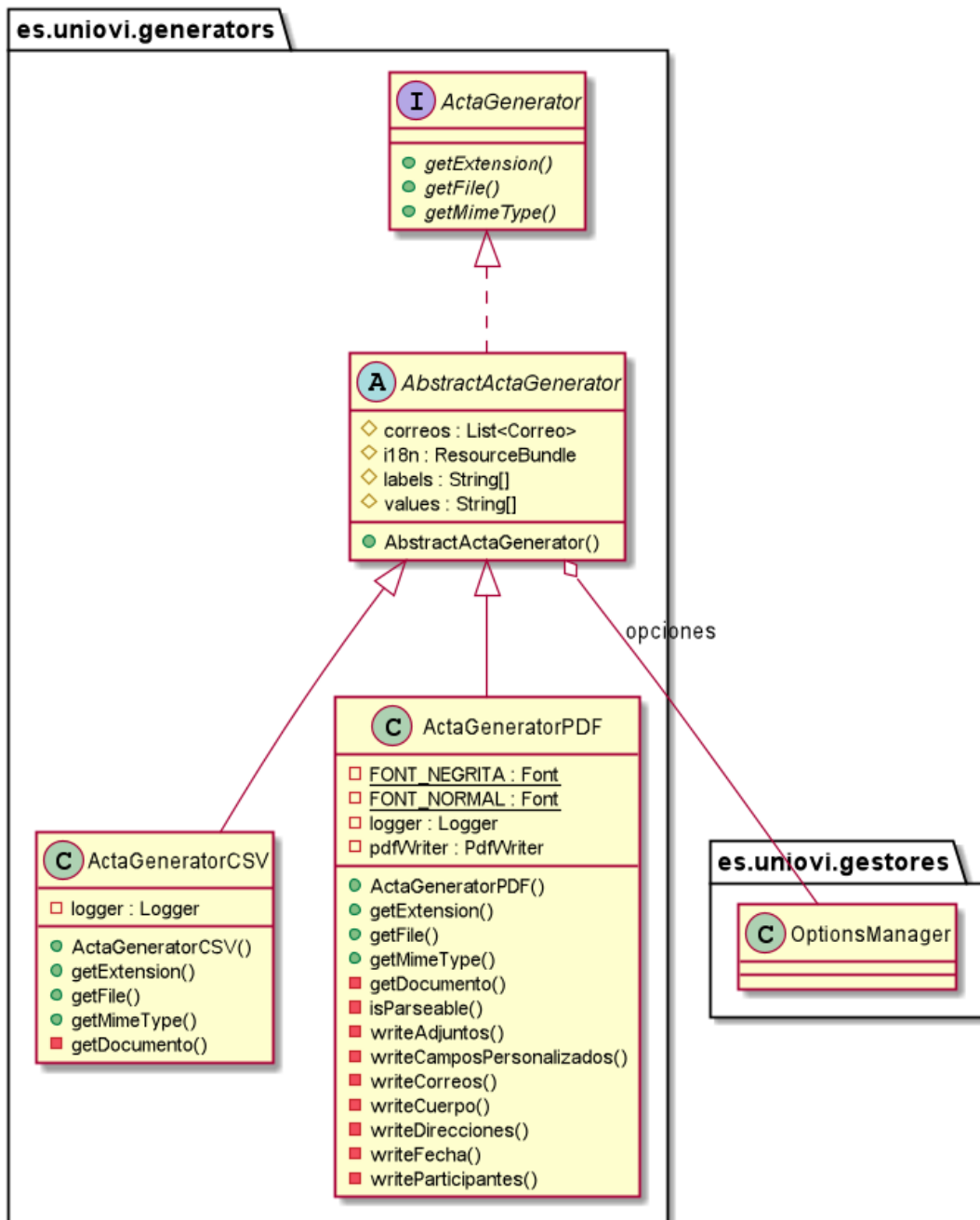


Figura 39 - Diagrama de clases (paquete generators)

La estructura de generadores es la siguiente. Contiene una interfaz para tratarlos de manera similar y una clase abstracta que permite implementar cierta funcionalidad común, esto es, el *parseo* de etiquetas y valores, la carga de los archivos de internacionalización y la obtención de la lista de correos. Esta clase abstracta también tiene un `OptionsManager` que almacenará la información *parseada* de la lista de opciones de exportación. Por último, hay un generador por tipo que implementa la funcionalidad restante, con sus propios métodos privados.

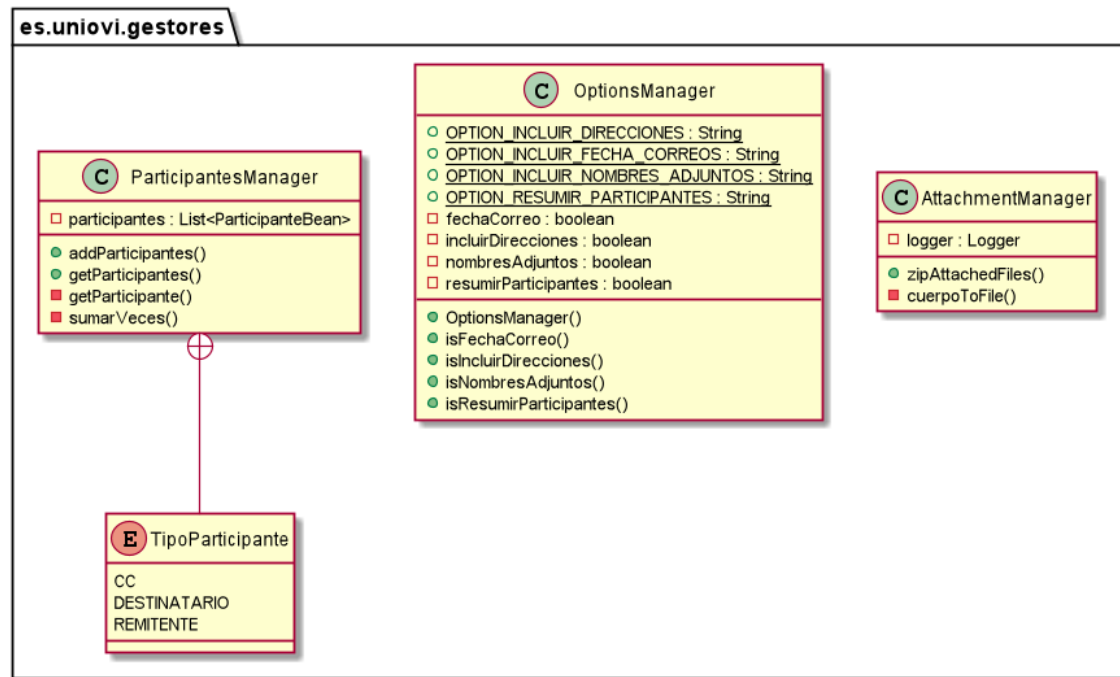


Figura 40 – Diagrama de clases (paquete gestores)

Los gestores se usan en diferentes partes e implementan una serie de funcionalidades. El `ParticipantesManager` se crea allí donde se necesiten computar datos de una serie de participantes. Sólo se usa como variable local, de ahí que no aparezca en ningún otro diagrama. Tiene una enumeración que define el tipo de participante. El `OptionsManager` tiene una estructura muy sencilla. Una serie de constantes que permiten *parsear* las diferentes opciones de exportación, un atributo por cada una de ellas y un *getter* para cada uno. El constructor *parsea* una serie de *strings* y establece el valor de los atributos. Además, el `AttachmentManager` crea un ZIP a través de su método `zipAttachedFiles()`.

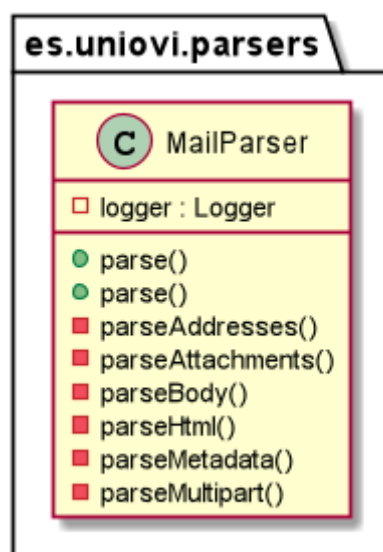


Figura 41 – Diagrama de clases (paquete parsers)

El único *parser* que tenemos es el `MailParser`. Contiene un método `parse`, que está sobrecargado para llamarlo con más o menos parámetros y una serie de métodos privados que *parsean* las diferentes partes del correo.

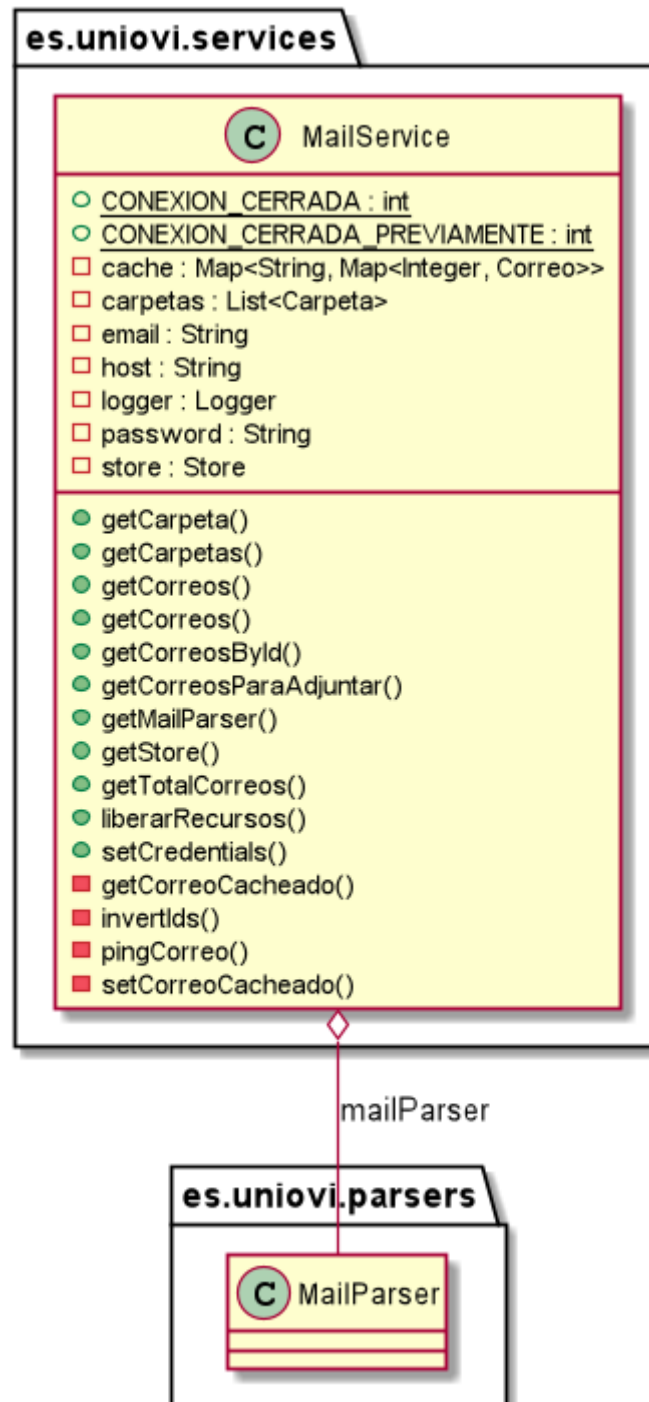


Figura 42 – Diagrama de clases (paquete parsers)

También tenemos un solo servicio, que es el que usa el `MailParser`. Este servicio contiene una serie de métodos que permiten obtener diferentes datos del proveedor (carpetas, correos,

número de correos...). También varios métodos privados que sirven de utilidad. Almacena las credenciales de la conexión y cachea los correos ya leídos para facilitar su carga.

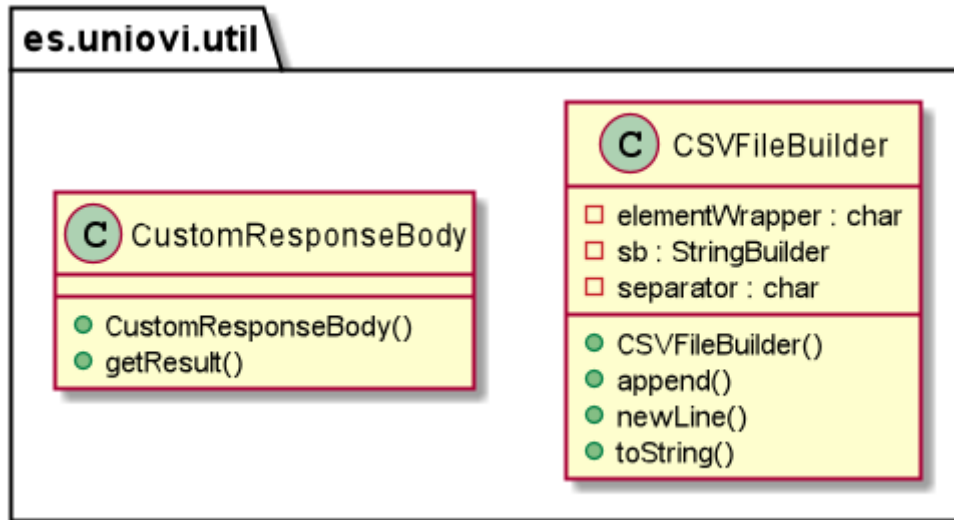


Figura 43 – Diagrama de clases (paquete util)

Por último, las clases del paquete Util. CustomResponseBody es un envoltorio sobre un tipo de dato que queremos devolver. Por ejemplo, cuando devolvemos el número de correos. De esta manera se devuelven objetos JSON uniformes, a la hora de retornar un valor simple. Por su parte, el CSVFileBuilder, nos facilita la creación de archivos CSV.

6.2.1.2 Front-end

Esta parte del sistema tiene menos clases, por tanto, podemos incluir el diagrama entero en la misma página. Sin embargo, para poder incluirlo hemos tenido que renunciar a incluir los paquetes. Aun así, estos son fáciles de identificar basándose en el nombre de las clases.

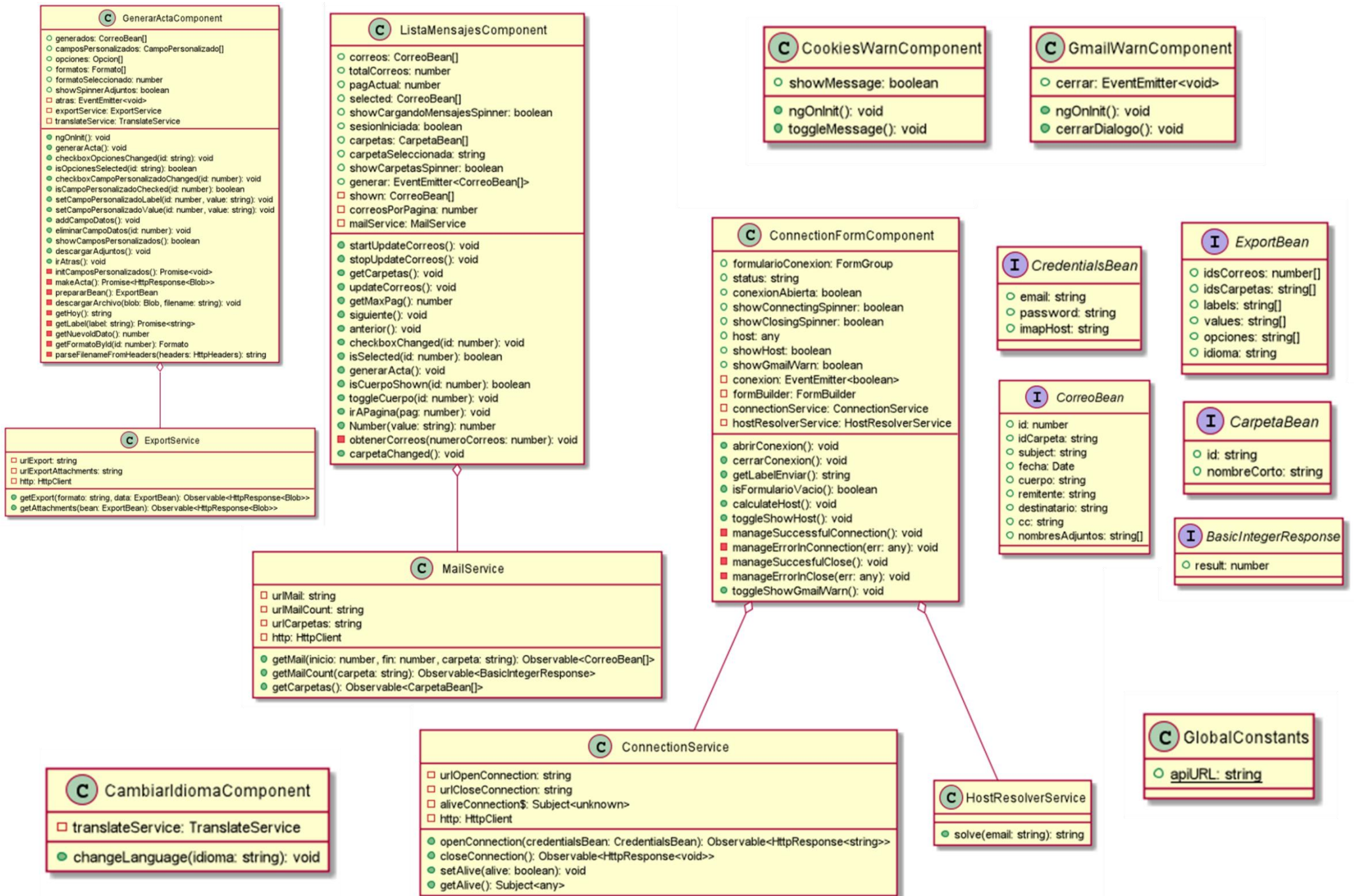


Figura 44 – Diagrama de clases del front-end

El diagrama de la página anterior podemos ver cómo hay varios tipos de clases. Por un lado, están los componentes (aquellos cuyo nombre termina en “component”). Parece que todos ellos tienen muchísimos métodos y atributos, sin embargo, todos estos métodos son muy simples y son usados por el *framework* de Angular para manejar las interacciones con la pantalla. Los componentes llaman a los servicios (con el nombre terminado en “service”). Estas clases tienen algunas cosas en común. Todas contienen una serie de atributos, que definen las URLs a las que hacer las llamadas y una serie de métodos que las realizan. También un cliente HTTP que se encargará de las peticiones. La excepción a esto es el `HostResolverService`.

Por otro lado, están los `Beans`, que permiten el mapeo de datos de las llamadas al *back-end*. Hay también una clase `GlobalConstans` que almacena la URL básica de la API.

6.3 Diagramas de Interacción

Una de las herramientas imprescindibles para entender un diseño son los diagramas. En capítulos anteriores ya vimos el sistema desde diferentes ángulos, por un lado, la distribución de los diferentes artefactos, tanto lógica como físicamente, la composición de las clases y sus métodos y ahora es el momento de seguir bajando niveles, para ver unos diagramas muy similares al código de la aplicación.

En las próximas líneas veremos diagramas de interacción para las partes más importantes del sistema. A veces este apartado se divide y se va explicando por casos de uso, pero en nuestra aplicación hay grandes variaciones entre la complejidad de unos casos y la de otros. Para hacerlo más inteligible, iremos desgranando procesos importantes del sistema, uno a uno.

Los diagramas de interacción que hagamos mostrarán los objetos y métodos que se llaman en el desarrollo de un proceso. Se intenta que sean exhaustivos y para ello se usan los mismos nombres que en el código. Sin embargo, en algún punto hemos quitado alguna llamada que sólo aporta complejidad al diagrama. Por ejemplo, una llamada a un método privado, en la misma clase, que simplemente facilita el mantenimiento. También se simplifican las llamadas a objetos externos (como al proveedor de correos o a alguna librería). Por último, indicar el código de colores, el color amarillo representa objetos en el *front-end*; el azul, objetos en el *back-end* y el rojo, objetos externos al sistema desarrollado.

6.3.1 Abrir una conexión

Este apartado si se corresponde con un caso de uso, el Caso de Uso 1 – Abrir conexión. El diagrama de interacción es el siguiente.

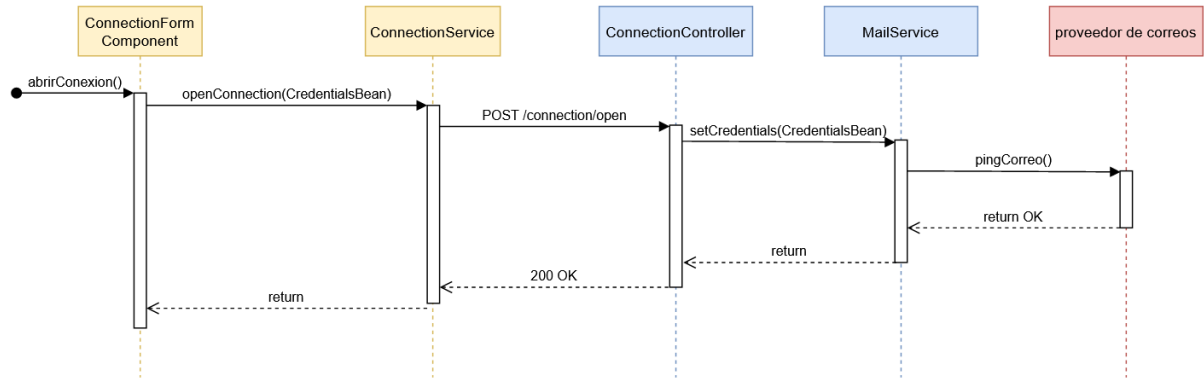


Figura 45 – Diagrama de Interacción (Abrir conexión)

La llamada a `abrirConexion` desencadena este proceso. Esto se hace desde el botón de la interfaz que abre las conexiones. Como puede verse es un proceso que encadena llamadas desde el *front-end* al *back-end*. Primero envía las credenciales hasta probar una conexión con el proveedor de correo y después va retornando mensajes de OK.

6.3.2 Cambiar idioma

El cambio de idioma también se puede relacionar con un caso de uso de manera sencilla, el Caso de Uso 2 – Cambiar idioma.

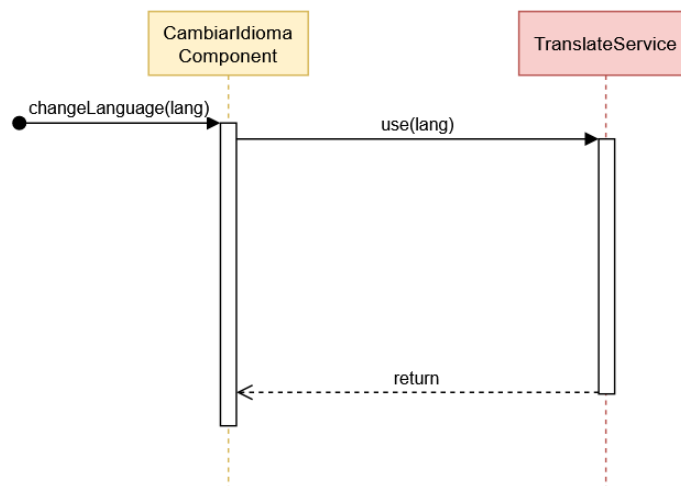
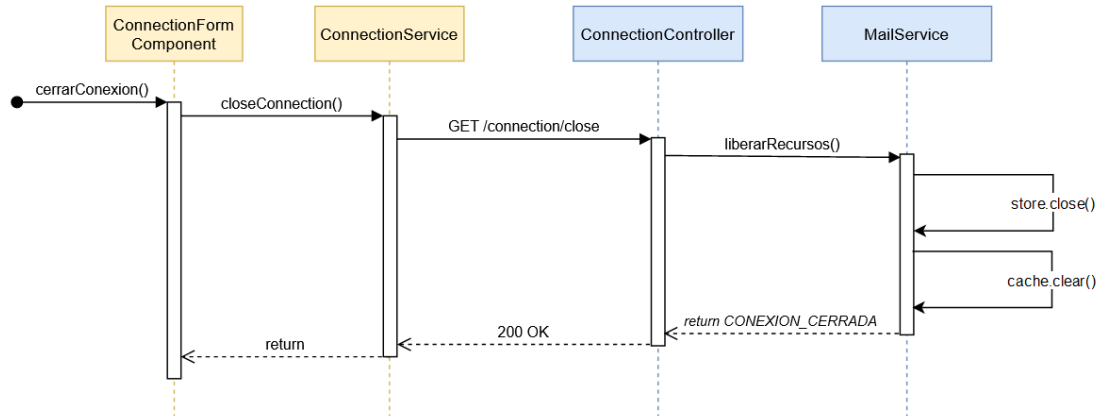


Figura 46 – Diagrama de Interacción (Cambiar Idioma)

No presenta ninguna dificultad, pues al llamar al método `changeLanguage`, con un idioma como parámetro, se llama al servicio de traducción, que es una librería externa y se modifica allí. Este servicio de traducción devolverá en la fase de renderizado los textos adecuados de acuerdo con el idioma seleccionado.

6.3.3 Cerrar Conexión

Este diagrama también puede relacionarse con un caso de uso, el Caso de Uso 3 – Cerrar conexión.



El cierre de la conexión es muy similar a su apertura. Sin embargo, tiene varias particularidades. Aquí no hay conexión con terceros, simplemente se cierran los objetos abiertos en el *back-end* y se vacía la caché. Se propagan mensajes de retorno confirmando el cierre.

6.3.4 Obtener Carpetas

La obtención de carpetas es uno de los procesos clave de la aplicación. No hay ningún caso de uso que haga exactamente esto, pero es un punto clave del Caso de Uso 4 – Seleccionar correos. La obtención de carpetas es un paso imprescindible de cara al listado de los correos. El proceso es el siguiente.

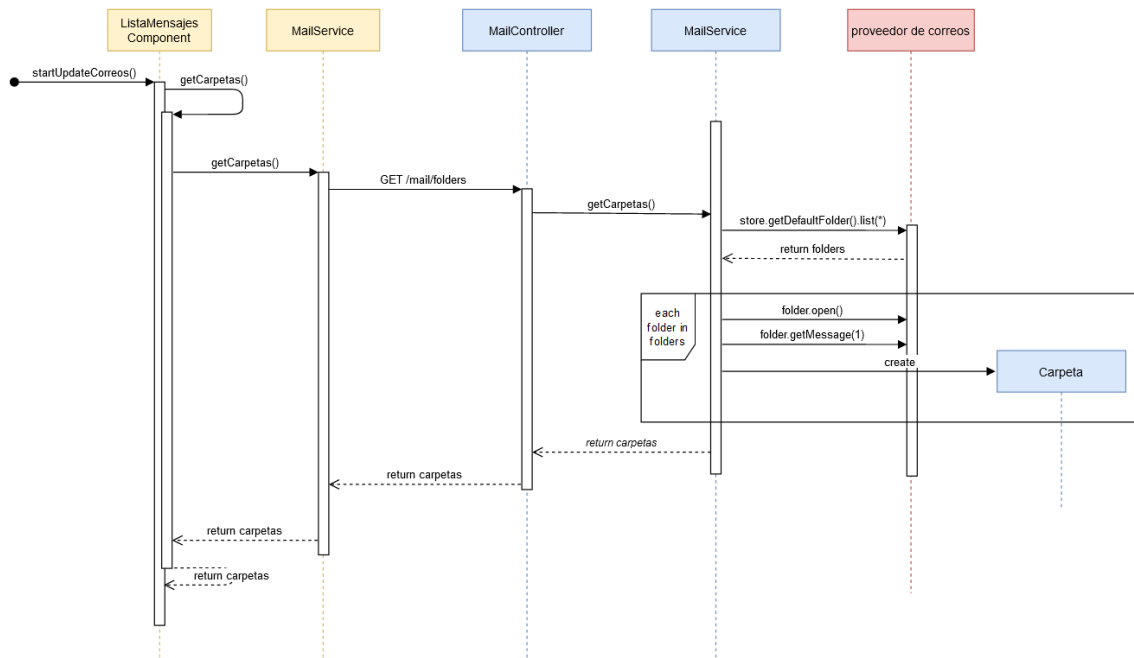


Figura 47 – Diagrama de Interacción (Obtener carpetas)

Este proceso se desencadena con la llamada a `startUpdateCorreos`, esto puede ser por la apertura exitosa de una conexión o por la vuelta atrás desde la pantalla de exportación. La parte compleja de este proceso es la comunicación entre el **MailService** y el proveedor de correos. Aquí es donde se obtienen las carpetas y para cada una se intenta obtener el primer mensaje, de esta manera se confirma que la carpeta está disponible. De ser así se crea un objeto **Carpeta**. Se devolverá una instancia de este objeto por cada una de las carpetas obtenidas del proveedor.

6.3.5 Obtener Correos

Tras el proceso del punto anterior hay que obtener los correos de cada carpeta. Este proceso debe ejecutarse una vez que se carguen las carpetas por primera vez, cuando se cambia de carpeta o cuando se cambia de página en la lista de correos.

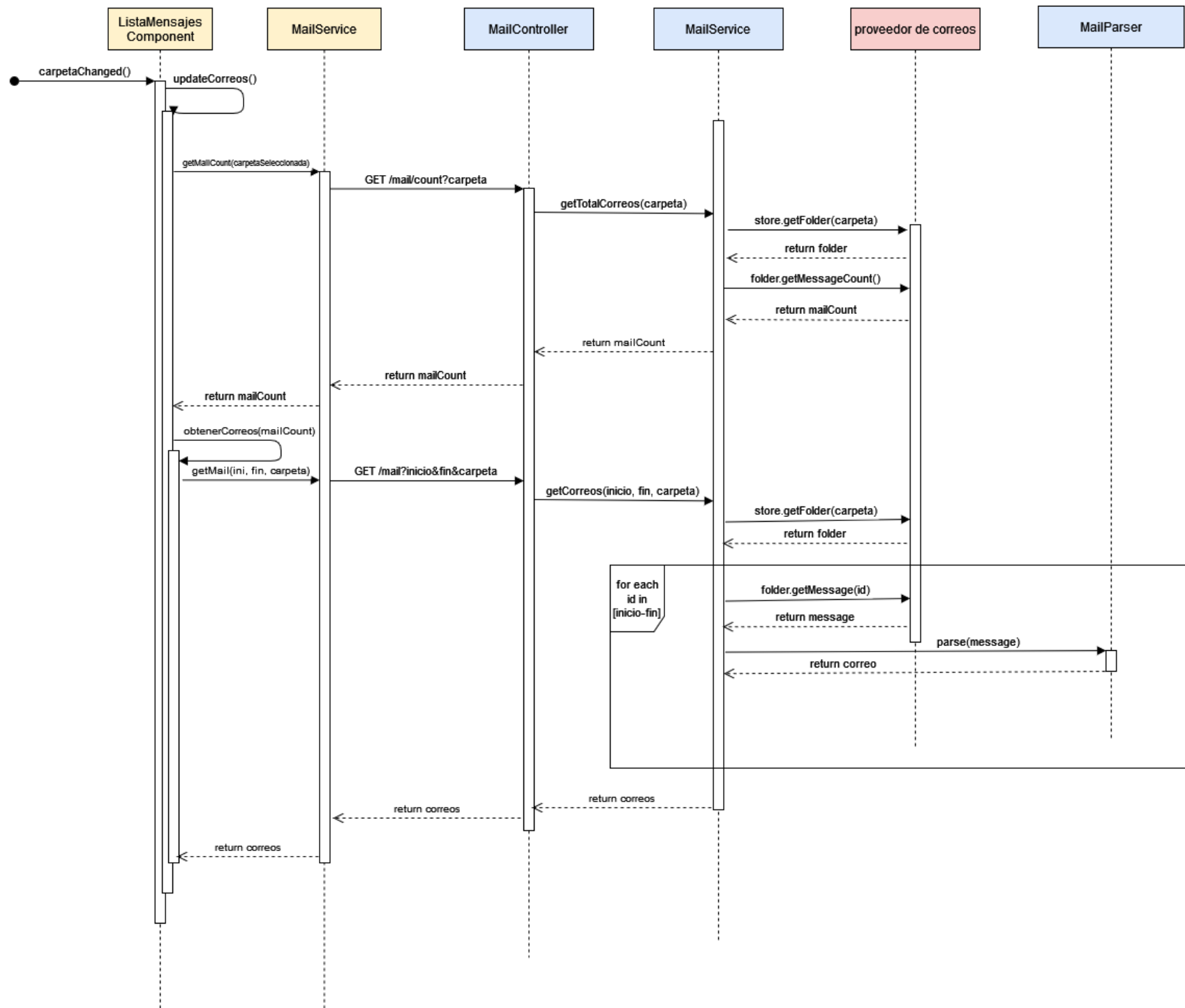


Figura 48 – Diagrama de interacción (Obtener correos)

El proceso contiene dos pasos, por un lado, la obtención del número de correos que hay en la carpeta y por otro la obtención de los correos en sí. Para conseguir el número de correos se debe especificar una carpeta y se harán las llamadas correspondientes hasta obtener el valor numérico. Acto seguido se actualizan los correos. Este segundo proceso puede también llamarse de manera independiente y se optimiza en la aplicación para evitar llamadas innecesarias. Los correos se piden por medio de tres valores: una carpeta, el identificador del primer correo y el del último. Hay que indicar que los identificadores de los correos son correlativos. El proceso verdaderamente complejo está en la parte del `MailService`. Aquí se obtiene la carpeta. Posteriormente se itera sobre los correos de esa carpeta cuyo id esté en el rango pedido. Para cada mensaje obtenido se llama al `MailParser` y se obtiene el correo parseado. Todos los correos se devuelven posteriormente al *front-end*.

6.3.6 Generar Acta

El último proceso clave del sistema es la generación de un acta. En este caso se hará el diagrama para un PDF, pero el proceso es igual para un CSV. Su diagrama de interacción es este.

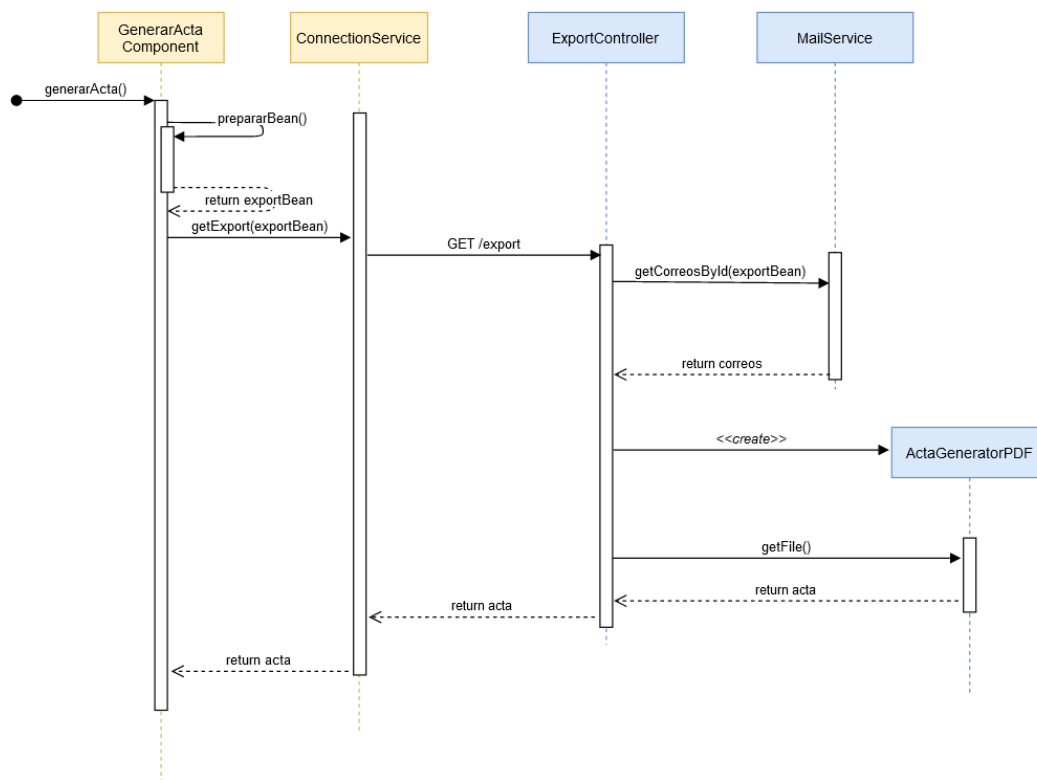


Figura 49 – Diagrama de interacción (Generar acta)

El proceso empieza clicando el botón correspondiente en la interfaz, que llama al método `generarActa`. En un método privado de la clase, se crea un objeto con todos los datos de la exportación, es el `exportBean`. Se realizan las llamadas correspondientes hasta el `MailService`, donde deben obtenerse los correos. El proceso aquí es igual que en el punto anterior, por lo que se ha suprimido. En definitiva, el método `getCorreosById` devuelve

los correos listos para exportar. Con ellos se crea el generador de PDF, que prepara el archivo y lo devuelve al *front-end*.

Podrían hacerse más diagramas, pero los procesos básicos son estos. También hay otros muy parecidos, como la generación de archivos adjuntos, que es muy similar a la generación de actas. La mayoría de los que no hemos incluido son pequeñas interacciones entre elementos de la interfaz, extremadamente simples o partes muy específicas sin ninguna complejidad intrínseca, como el cacheo de correos, que no es más que un par de operaciones *put* y *get* sobre un mapa.

6.4 Diagramas de Actividades

Este será el último de los diagramas de la aplicación y con él queremos mostrar una visión rápida del proceso general de crear un acta. Si en los anteriores buscamos la exhaustividad, aquí queremos detallar de manera genérica el proceso, destinado a facilitar la comprensión del sistema a una persona que deba realizar modificaciones sobre él.

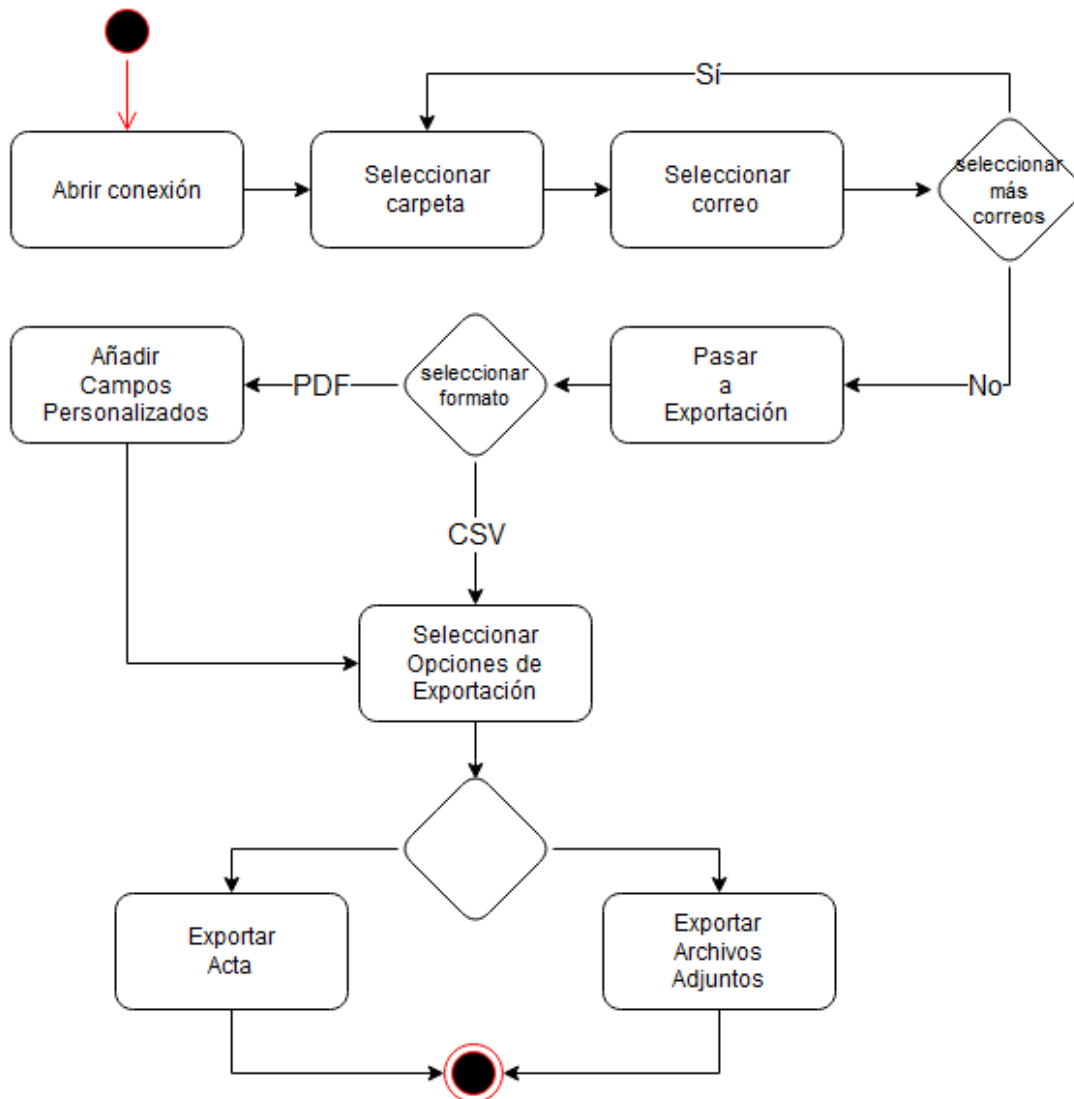


Figura 50 – Diagrama de actividades (Generar un acta/Exportar archivos)

El proceso se empieza abriendo una conexión, y posteriormente seleccionando los correos de las carpetas que queremos añadir. Una vez que ya los tenemos todos, pasamos a la fase de exportación. Elegimos el formato (PDF o CSV) y en función de ello deberemos añadir campos personalizados y opciones exportación. Por último, decidiremos si exportar el acta o los archivos adjuntos.

6.5 Diseño de la Interfaz

En este apartado ya se va a mostrar la aplicación definitiva. Se mostrará la interfaz final, componente a componente, construida en base a los bocetos del Análisis de Interfaces de Usuario redactado anteriormente. Primero se mostrará la apariencia general y después se irá profundizando componente a componente.

Hay dos vistas generales, la de listado y la de exportación. La primera es la Figura 51 y muestra el componente de conexión y el de listado de correos. La segunda es la de la Figura 52, que sustituye el listado por el componente de exportación. Hay algunos componentes que se mostrarán permanentemente. Uno es el encabezado, que contiene el logo, el nombre y el subtítulo de la aplicación. Otro es el pie, con los logos corporativos y los datos del trabajo. El otro componente que esta siempre visible es el de cambio de idioma. Las banderas de Asturias, España y Reino Unido permiten cambiar los textos de la aplicación al asturiano, castellano o inglés, respectivamente.




Figura 51 – Vista General de la Aplicación (Listado)

U M A


Unvíos Na Andecha

Una sencilla herramienta para generar actas a partir de tus correos



Correo electrónico

Contraseña



Reconectar Conectado con éxito

Cerrar Conexión

←

Selecciona el formato en que se generará el acta

CSV (Comma Separated Values) ▾

Configuración

- Incluir direcciones (Remitente, destinatario, CC)
- Mostrar la fecha de los correos exportados
- Incluir nombre de los archivos adjuntos
- Incluir resumen participantes

Se van a exportar los siguientes correos:

- Descubre de las novedades que hemos preparado para ti

Descargar archivo

Descargar archivos adjuntos

Figura 52 – Vista general de la aplicación (Exportación)

6.5.1 Componente de conexión

El componente que abre y cierra conexiones se muestra en la Figura 53. Está basado en un formulario con dos botones que abren y cierran la conexión en base a los datos proporcionados. El engranaje que se muestra bajo el campo “Contraseña” muestra u oculta el campo “Host”. La letra “i” muestra las aclaraciones para la conexión con Gmail, por tener este proveedor medidas de seguridad extra que dificultan su uso. Las aclaraciones se muestran en un diálogo modal, que se muestra superpuesto a la pantalla principal, y es el de la Figura 54. Contiene un texto y un enlace.

The screenshot shows a form with three input fields and two buttons. The first field is labeled 'Correo electrónico' and contains 'xxxxxxx@gmail.com'. The second field is labeled 'Contraseña' and is filled with dots. The third field is labeled 'Host' and contains 'smtp.gmail.com'. Below the fields are two buttons: a green 'Conectar' button and a red 'Cerrar Conexión' button. To the right of the password field is a gear icon and a red 'i' icon.

Figura 53 – Componente de conexión

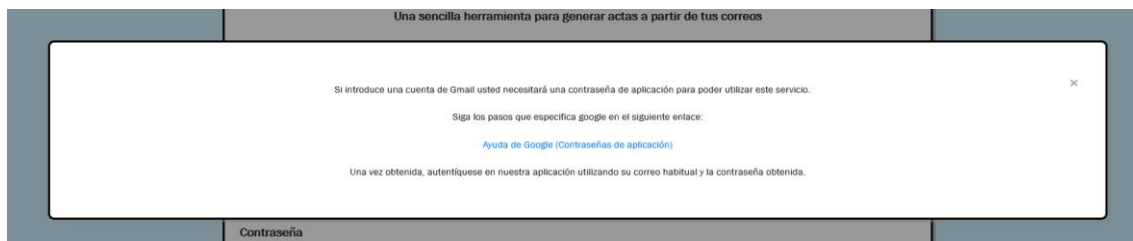


Figura 54 – Aclaraciones para la conexión con Gmail

6.5.2 Componente de selección

La selección de correos se hace a través de un listado paginado. Es el de la Figura 55. Tiene un desplegable en el que se muestran las carpetas disponibles, cuando este se modifica los correos se actualizan. Los *checkboxes* permiten seleccionar los correos para la exportación y los botones de la parte inferior permiten navegar entre páginas. También puede mostrarse el contenido de los correos clicando sobre el asunto. En ese caso, se desplegará como en la Figura 56, donde se incluyen el contenido y los metadatos del correo.

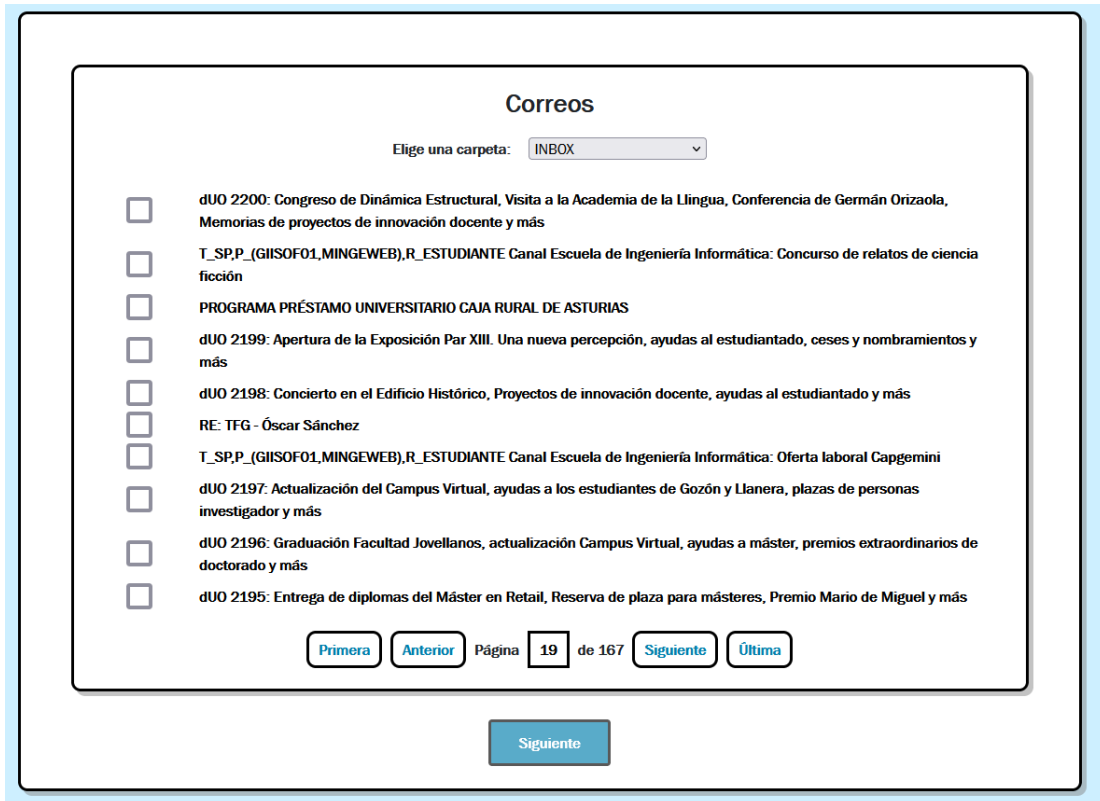


Figura 55 – Listado de correos

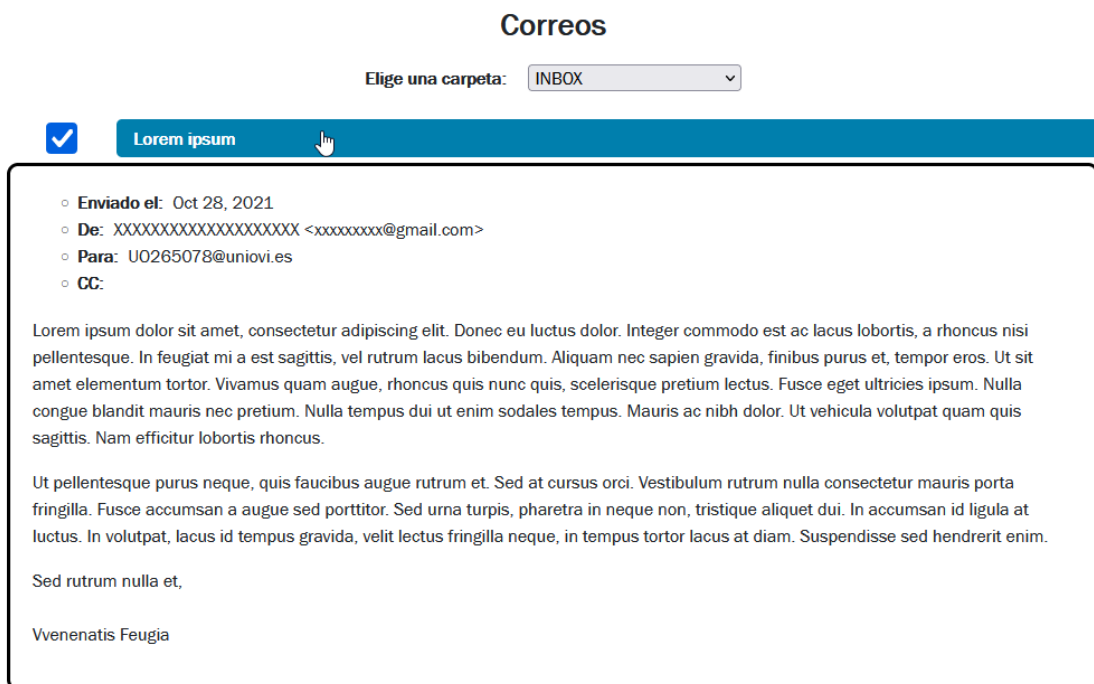


Figura 56 – Detalles de un correo seleccionado

6.5.3 Componente de exportación

Al clicar sobre el botón “Siguiente” de la Figura 55 se pasará a la fase de exportación, mostrándose el componente correspondiente. Este contiene un selector que permite la generación en PDF tal y como se ve en la Figura 57 o en formato CSV (Figura 58). Ambos tienen un botón para volver a la pantalla anterior. También comparten el mismo selector que alterna entre formatos, las opciones de exportación, que pueden seleccionarse o deseleccionarse usando las *checkboxes* que las acompañan, el resumen de los correos seleccionados y los botones que permiten generar el acta o descargar los archivos adjuntos. Sin embargo, al seleccionar el formato PDF aparece un componente más, que es exclusivo de este tipo de archivos, y que consta de una serie de campos, que pueden aumentarse o reducirse. Aquí se especificarán los campos personalizados para añadir al acta.

The screenshot shows a web interface for configuring PDF export. At the top left is a back arrow. Below it, the text 'Selecciona el formato en que se generará el acta' is followed by a dropdown menu currently set to 'PDF (Portable Document Format)'. Underneath, 'Datos que se añadirán al acta:' is followed by two rows of input fields. The first row has a checked checkbox, a label 'Título', a text input field containing 'Ejemplo de Acta', and a red trash icon. The second row has an unchecked checkbox, a label 'Fecha', a text input field containing '28/10/2021, 16:18:22', and another red trash icon. A green button labeled 'Añadir otro' is positioned below these fields. The 'Configuración' section contains four checked checkboxes: 'Incluir direcciones (Remitente, destinatario, CC)', 'Mostrar la fecha de los correos exportados', 'Incluir nombre de los archivos adjuntos', and 'Incluir resumen participantes'. Below this, the text 'Se van a exportar los siguientes correos:' is followed by a single bullet point '• Lorem ipsum'. At the bottom, there are two blue buttons: 'Descargar archivo' and 'Descargar archivos adjuntos'.

Figura 57 – Componente de exportación para archivos PDF

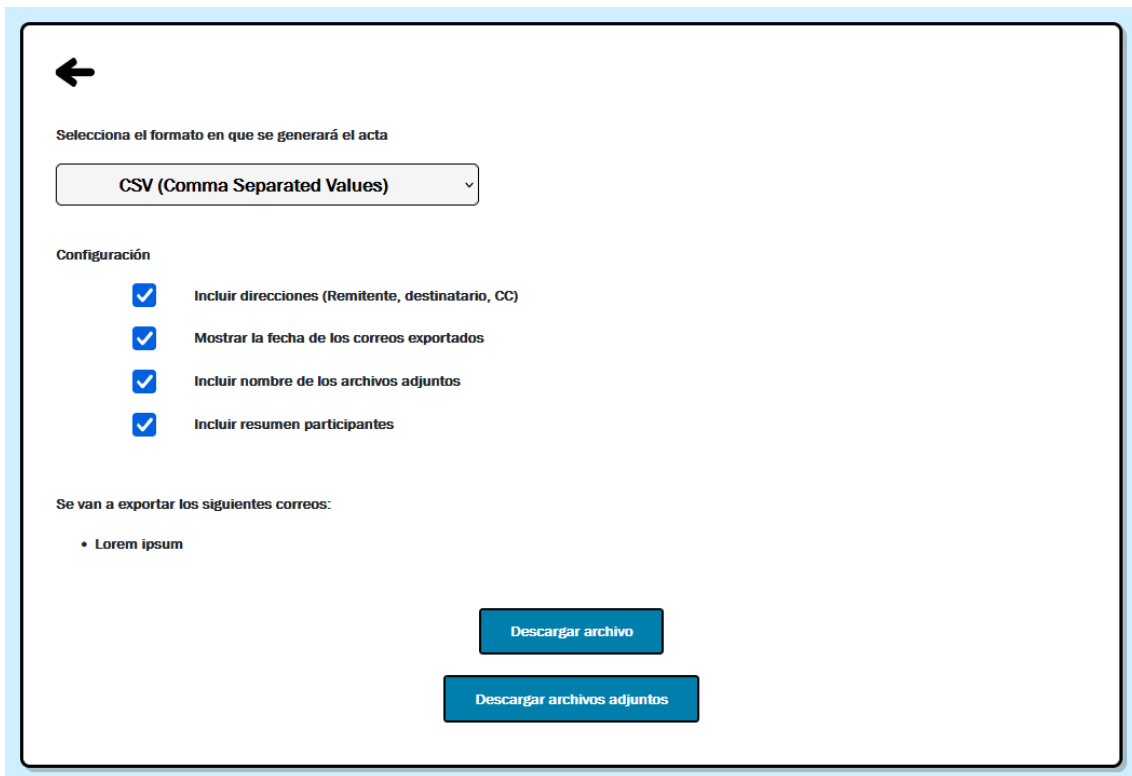


Figura 58 – Componente de exportación para archivos CSV

6.5.4 Aviso de cookies

El aviso de cookies es muy sencillo en cuanto a interfaz. Aparecerá al pie de la pantalla de nuestro navegador y podrá ocultarse clicando sobre el dibujo de la galleta. Este icono, aparecerá siempre en la parte inferior izquierda de la pantalla, sin llegar a ser invasivo, y permitirá volver a mostrar el aviso si este está oculto.



Figura 59 – Aviso de Cookies

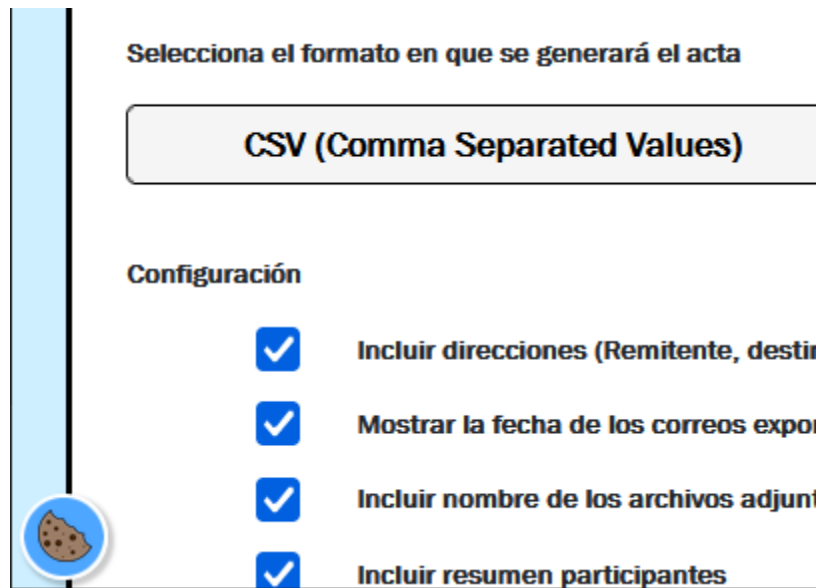


Figura 60 – Icono para mostrar/ocultar el aviso de cookies.

6.5.5 Actas

El producto final de todo el proceso es el acta generada. Tiene el formato que se muestra en la Figura 61, es decir, primero los campos personalizados que se hayan incluido, después la tabla de participantes y posteriormente cada uno de los correos junto a sus metadatos.

- Título: Acta de Ejemplo
- Fecha: 10/11/2021
- Descripción: Ejemplo de Acta Generado

Participantes

Participante	Remitente	Destinatario	CC
OFICINA DE COMUNICACION DE LA UNIVERSIDAD DE OVIEDO <comunica@uniovi.es>	3	3	0

dUO 2268: Premio Begoña Busto, seminario guerra anglo-española, actualización protocolo COVID, inicio campaña antigripal, código de conducta y gestión, becas departamentos, ayudas al estudiantado

- Remitente: OFICINA DE COMUNICACION DE LA UNIVERSIDAD DE OVIEDO <comunica@uniovi.es>
- Destinatario: OFICINA DE COMUNICACION DE LA UNIVERSIDAD DE OVIEDO <comunica@uniovi.es>
- Fecha de envío: martes 2 de noviembre de 2021

Diario de la Universidad de Oviedo (dUO) – Titulares – 02/11/21

—

- [Entrega del premio Begoña Busto](#)
- [Seminario "Nuevas miradas sobre la guerra anglo-española \(1585-1604\)"](#)
- [Jornada sobre "Desafíos del blockchain en la Contabilidad"](#)
- [Seminario de la Cátedra de Crecimiento e Internacionalización Empresarial](#)

Figura 61 – Ejemplo de acta generada

6.6 Especificación Técnica del Plan de Pruebas

En el capítulo anterior, realizamos una introducción a las pruebas que comprobarán el correcto funcionamiento de nuestro sistema. En ese punto, que concretamente es el apartado 5.8, sentamos unas bases de qué queremos probar y cómo vamos a hacerlo. Ahora, siguiendo esos principios, es momento de describir técnicamente y en detalle qué pruebas se van a hacer.

Las pruebas se realizarán en diferentes momentos del desarrollo sobre un sistema Microsoft Windows 10 Home 10.0.19042, con 16GB de RAM y procesador AMD Ryzen 7. Por ser una aplicación web, el navegador utilizado principalmente será Firefox 93, aunque también se usará Google Chrome 95 y Microsoft Edge 94 donde se necesiten probar varios navegadores.

En los siguientes subapartados, explicaremos las pruebas que se harán de cada tipo y todo aquello que las rodea. Esto implicará, por ejemplo, indicar las situaciones a cubrir de las pruebas unitarias, los resultados técnicos que se deben obtener de las pruebas del Sistema, las preguntas de los cuestionarios en las pruebas de usabilidad... También deben incluirse otros aspectos, como los sistemas en que se realizarán o el procedimiento de las mismas. Todo ello lo veremos en futuras líneas.

6.6.1 Pruebas Unitarias

A continuación, describiremos las pruebas unitarias y el resultado esperado de cada una. Dividiremos este apartado en dos partes, pues ni las herramientas utilizadas ni las pruebas realizadas en el *front-end* son las mismas que las del *back-end*. Estas pruebas se ejecutan constantemente a lo largo del desarrollo, y van ampliándose a la par que los incrementos realizados sobre el producto. Se realizarán estas pruebas tanto al final del desarrollo, como al final de los *sprints* e incluso antes de realizar cada *commit*.

6.6.1.1 *Front-end*

Las pruebas unitarias del cliente web Angular, se realizarán usando Karma como *test-runner* y se definirán usando Jasmine.

Aquí se describirán separados por componentes, especificando para cada uno el título de la prueba, los pasos que se realizarán y el resultado que se espera. Las pruebas se realizan únicamente sobre el componente que se indica, siendo *mocks* el resto de los que interactúan con él. También se usan *spies* en determinadas circunstancias.

CambiarIdiomaComponent		
Prueba	Procedimiento	Resultado esperado
cambiar idioma	Se cambia el idioma a asturiano y a castellano.	El idioma seleccionado es primero asturiano y posteriormente castellano.

ConexionComponent		
Prueba	Procedimiento	Resultado esperado

abrir conexión	Se especifican email y contraseña, se calcula el host y se abre la conexión.	El host se calcula correctamente, la conexión pasa a estado abierto y el botón muestra la etiqueta "reconectar".
abrir conexión – datos vacíos	No se especifica email y contraseña y se intenta abrir una conexión.	El formulario se detecta como vacío y la conexión no se abre.
abrir conexión – datos erróneos	Se especifica un email y una contraseña que no le corresponde, se calcula el host y se intenta abrir una conexión.	La conexión no se abre y se muestra el mensaje de "fallo de autenticación".
cerrar conexión – ok	Se cierra una conexión abierta.	La conexión pasa a estado cerrado y se muestra la etiqueta "conectar".
cerrar conexión – error	Se intenta cerrar una conexión cerrada.	La conexión se mantiene en estado cerrado y se muestra la etiqueta "conectar".
mostrar/ocultar host	Se muestra el campo host y posteriormente se oculta el campo host	Tras el primer paso se muestra el campo host y tras el segundo se oculta el campo host.
mostrar/ocultar aviso Gmail	Se muestra el aviso de Gmail y posteriormente se oculta el aviso de Gmail	Tras el primer paso se muestra el aviso de Gmail y tras el segundo paso no se muestra este aviso.

CookiesComponent		
Prueba	Procedimiento	Resultado esperado
ocultar aviso	El aviso se muestra en pantalla y se intenta ocultar.	El aviso se oculta.
mostrar aviso	Un aviso no se muestra en pantalla y se intenta mostrar.	El aviso se muestra.
primer acceso	Se accede por primera vez a la aplicación, es decir, no hay cookies almacenadas de la misma.	El aviso de cookies se muestra.
segundo acceso	Se accede por segunda vez a la aplicación, es decir, hay una cookie que dice que el aviso ya se ha ocultado.	El aviso de cookies no se muestra.

GenerarActaComponent		
Prueba	Procedimiento	Resultado esperado
marcar/desmarcar opción	Marca el <i>checkbox</i> de una opción de exportación y posteriormente la desmarca.	Después del primer paso aparecerá marcada esa opción, y tras el segundo: desmarcada.
marcar/desmarcar campo personalizado	Se clica sobre el <i>checkbox</i> de un campo personalizado	El <i>checkbox</i> aparece con el valor contrario
modificar campo personalizado	Se modifica la etiqueta de un campo personalizado y el valor de otro	La etiqueta y el valor deben reflejar los valores introducidos
añadir/quitar campo personalizado	Se añade un campo personalizado, se eliminan todos los campos personalizados, se añade un campo nuevo.	El número de campos personalizados mostrados debe ser el correcto (3, 0 y 1)

modificar formato	Se selecciona el formato PDF y luego el formato CSV	Tras el primer paso se muestran los campos personalizados y tras el segundo no
generar acta	Se llama al método <code>generarActa</code> (que es el que se ejecuta al clicar el botón de generar el acta)	Se llama sin errores al método correspondiente del servicio de exportación
descargar comprimidos	Se llama al método <code>descargarAdjuntos</code> (que es el que se ejecuta al clicar el botón de descargar los adjuntos)	Se llama sin errores al método correspondiente del servicio de exportación
descargar comprimidos – error	Se llama al método <code>descargarAdjuntos</code> y se devuelve un error desde el servicio	Se comprueba que se devolvió el error correcto
ir atrás	Se clica el botón de volver a la lista de mensajes	Se comprueba que se vuelve a la lista de mensajes

GmailAvisoComponent

Prueba	Procedimiento	Resultado esperado
cerrar componente	Se clica sobre el botón de cerrar el componente	Se oculta el componente y ya no se muestra en pantalla

ListaMensajesComponent

Prueba	Procedimiento	Resultado esperado
cargar mensajes – ok	Se inicia el procedimiento de carga de correos.	Se comprueba que se carga el número esperado de correos totales, carpetas y correos en la página actual.
cargar mensajes – carpeta undefined	No se selecciona ninguna carpeta y se intentan cargar los correos	Se comprueba que no se han cargado correos
cargar mensajes fin > total correos	Se intentan obtener los correos de un rango mayor que el número de correos totales	Se muestran sólo los correos en el rango válido sin errores
cargar mensajes si total correos es 0	Se intentan obtener los correos de una carpeta vacía	Se muestra la lista de correos vacía
carpeta no existente	Se intentan obtener los correos de una carpeta con un identificador que no existe	Se muestra la lista de correos vacía, un error en consola y el spinner de carga se oculta
detener carga mensajes	Se detiene la carga de correos (esto sucede, por ejemplo, cuando se cierra la conexión actual)	Se comprueba que las variables toman los valores iniciales
get última página (0)	Se intenta obtener la última página del paginador, cuando la carpeta está vacía	El valor devuelto es 1.
get última página (undefined)	Se intenta obtener la última página del paginador, cuando no se han cargado correos aún.	El valor devuelto es 1.

get última página (divisible)	Se intenta obtener la última página del paginador, cuando el número de correos es divisible entre el máximo de correos por página	El valor devuelto es "página dada / máximo de correos por página"
get última página (con resto)	Se intenta obtener la última página del paginador, cuando el número de correos no es divisible entre el máximo de correos por página	El valor devuelto es "página dada / máximo de correos por página - 1"
get página siguiente (válida)	Se intenta obtener la siguiente página del paginador estando en una página que no sea la última	Se carga la página siguiente a la actual
get página siguiente (mayor)	Se intenta obtener la siguiente página del paginador estando ya en la última página	El paginador se mantiene en la página actual
get página anterior (válida)	Se intenta obtener la página anterior del paginador estando en una página distinta de la primera	Se carga la página anterior a la actual
get página anterior (menor)	Se intenta obtener la página anterior del paginador estando ya en la primera página	Se carga la primera página del paginador
ir a página (válida)	Se intenta ir a una página en un rango de páginas válido	Se carga la página con el número dado
ir a página (mayor)	Se intenta ir a una página con un número superior a la última página	Se carga la última página del paginador
ir a página (negativa/0)	Se intenta ir a la página 0 y a una página con un número negativo	Se carga la página 1 del paginador
seleccionar correo	Se clicca el <i>checkbox</i> para seleccionar un correo	El correo clicado aparece como seleccionado
deseleccionar correo	Se clicca el <i>checkbox</i> de un correo seleccionado	El correo clicado aparece como deseleccionado
mostrar cuerpo correo	Se clicca un correo para mostrar su cuerpo	Se muestra el cuerpo del correo clicado
ocultar cuerpo correo	Se clicca el cuerpo de un correo que se muestre para ocultarlo	No se muestra el cuerpo del correo clicado
parse Number (válido)	Se <i>parsea</i> una cadena con un número entero a Number	Se devuelve el número entero con el tipo Number
parse Number (inválido)	Se <i>parsea</i> una cadena con letras a Number	Se devuelve un objeto Number con valor 1
parse Number (decimal)	Se <i>parsea</i> una cadena con un número decimal a Number	Se devuelve un objeto Number con el valor decimal dado
cambio carpeta	Se selecciona una carpeta y se cargan sus correos	Se carga el número de correos adecuado y sin errores
get carpetas – error	Se intentan cargar las carpetas y el servicio devuelve un error	Se oculta el spinner de carga y se muestra un desplegable vacío
get carpetas es 0	Se intentan cargar las carpetas y no se devuelve ninguna	Se muestra un desplegable vacío
generar acta	Se clicca el botón para pasar a la pantalla de exportación	Se pasa a la pantalla de exportación

Prueba	Procedimiento	Resultado esperado
abrir conexión – ok	Se proporcionan unas credenciales válidas y se abre una conexión	Se devuelve el mensaje de “Conexión Abierta”
abrir conexión – credenciales incorrectas	Se proporcionan unas credenciales incorrectas y se intenta abrir una conexión	Se devuelve un error con status 401 y el texto correspondiente
cerrar conexión – ok	Se intenta cerrar una conexión	Se devuelve una respuesta con status 200 y el texto correspondiente
cerrar conexión – ya cerrada	Se intenta cerrar una conexión que no se haya abierto	Se devuelve una respuesta con status 409 y el texto correspondiente
cerrar conexión - error inesperado	Se produce un error en el servidor al cerrar la conexión	Se devuelve una respuesta con status 500 y el texto correspondiente
comprobar conexión con vida	Se obtiene el status actual de la conexión (cerrada), se modifica a abierta y se suscribe a la promesa para obtener su siguiente valor	El valor que devuelve la promesa es “conexión abierta”.

ExportService		
Prueba	Procedimiento	Resultado esperado
generar acta - ok	Se proporcionan unos datos válidos y se realiza la petición para generar un acta	Se devuelve un archivo con el acta del tipo pedido
generar acta – error inesperado	Se realiza la petición para generar un acta y se produce un error en el servidor	Se devuelve un error del servidor y con él se genera un error de la aplicación
obtener adjuntos – ok	Se proporcionan unos datos válidos y se realiza la petición para exportar los archivos adjuntos	Se devuelve un archivo de tipo <i>application/x-zip</i>
obtener adjuntos – error inesperado	Se realiza la petición para obtener los archivos adjuntos y se produce un error en el servidor	Se devuelve un error del servidor y con él se genera un error de la aplicación

HostResolverService		
Prueba	Procedimiento	Resultado esperado
valor correcto - Gmail	Se pide obtener el host para una dirección de Gmail	Se devuelve smtp.gmail.com
valor correcto - Outlook	Se pide obtener el host para una dirección de Outlook	Se devuelve smtp.office365.com
proveedor no soportado	Se pide obtener el host para un proveedor no soportado	Se devuelve null
no es dirección correo	Se pide obtener el host para una cadena que no sea una dirección de correo	Se devuelve null

MailService		
Prueba	Procedimiento	Resultado esperado

obtener correos – ok	Se intentan obtener una serie de correos en un rango válido	Se devuelven los correos sin errores
obtener correos – error inesperado	Se intentan obtener los correos y se produce un error en el servidor	Se devuelve un error del servidor y con él se genera un error de la aplicación
obtener número correos – ok	Se intenta obtener el número de correos en una carpeta	Se devuelve el número de correos en la carpeta
obtener número correos - error inesperado	Se intenta obtener el número de correos en una carpeta y se produce un error en el servidor	Se devuelve un error del servidor y con él se genera un error de la aplicación
obtener carpetas - ok	Se intentan obtener las carpetas del usuario en el proveedor	Se devuelven las carpetas
obtener carpetas - error inesperado	Se intentan obtener las carpetas del usuario en el proveedor y se produce un error en el servidor	Se devuelve un error del servidor y con él se genera un error de la aplicación

AppComponent		
Prueba	Procedimiento	Resultado esperado
empezar actualización correos	Se comunica al componente “lista de correos” que debe empezar a actualizar los correos	La sesión pasa a estado iniciado
parar actualización correos	Se comunica al componente “lista de correos” que debe parar la actualización de los correos	La sesión pasa a estado no iniciado
pasar a exportación	Se pone en funcionamiento el componente “exportación” dada una lista de correos	La variable “exportando” pasa a true
pasar a listado	Se pone en funcionamiento el componente “lista de correos”	La variable “exportando” pasa a false

6.6.1.2 Back-end

Las pruebas del *back-end* se realizan en Java usando JUnit 4. También se usa PowerMockito para la inyección de *mocks* donde hagan falta. Al igual que en el *front-end* se ejecutarán las pruebas sobre los componentes especificados, siendo *mocks* cada uno de los otros con los que interactúen.

CorreoBeanParsing		
Prueba	Procedimiento	Resultado esperado
parseCorreo	Dado un objeto de la entidad Correo se construye un objeto CorreoBean	Las propiedades se mapean de manera correcta

ConnectionControllerTests		
Prueba	Procedimiento	Resultado esperado
abrirConexionOk	Se proporcionan unas credenciales válidas y se abre una conexión	Se devuelve el mensaje de éxito adecuado

abrirConexionCredencialesInvalidas	Se proporcionan unas credenciales inválidas y se abre una conexión	Se devuelve un error 401
cerrarConexionOk	Se intenta cerrar una conexión	Se devuelve un mensaje de éxito
cerrarConexionCerradaPreviamente	Se intenta cerrar una conexión ya cerrada	Se devuelve un mensaje indicando la circunstancia, pero ningún error
cerrarConexionErrorInesperado	Se intenta cerrar una conexión y se produce un error inesperado	Se devuelve un error con el texto "Error inesperado"
cerrarConexionNoEstablecida	Se intenta cerrar una conexión no establecida	Se devuelve un error con el texto "La conexión no se llegó a abrir"

ExportControllerTests		
Prueba	Procedimiento	Resultado esperado
exportarActaPdfOk	Se realiza una petición para generar un acta PDF	Se devuelve un acta en formato PDF
exportarActaCsvOk	Se realiza una petición para generar un acta CSV	Se devuelve un acta en formato CSV
exportarActaPdfErrorEtiquetas	Se realiza una petición para generar un acta PDF, pero el número de etiquetas de los campos personalizados no coinciden con el número de valores	Se produce un error con un mensaje al respecto
exportarActaPdfIoUtilsError	Se realiza una petición para generar un acta PDF y se produce un error en la manipulación del archivo para su exportación	Se produce un error con un mensaje al respecto
exportarActaFormatoNoSoportado	Se intenta exportar un acta con un formato no soportado	Se produce un error con un mensaje al respecto
exportarActaConexionCerrada	Se intenta exportar un acta cuando no se ha abierto ninguna conexión	Se produce un error con un mensaje al respecto
exportarAdjuntosOk	Se realiza una petición para exportar los archivos adjuntos	Se devuelve un archivo ZIP
exportarAdjuntosIoException	Se produce una IOException generando el ZIP con los archivos adjuntos	Se produce un error con un mensaje al respecto

MailControllerTests

Prueba	Procedimiento	Resultado esperado
getMailOk	Se realiza una petición para obtener los correos de una carpeta	Se devuelven los correos de esa carpeta
getMailRangosInvalidos	Se realiza una petición para obtener los correos de una carpeta en múltiples rangos inválidos: <ul style="list-style-type: none"> • [a,b] con $a, b < 0$ y $a < b$ • [negativo, positivo] <ul style="list-style-type: none"> • [0,1] • [a,b] con $a, b < 0$ y $a < b$ <ul style="list-style-type: none"> • [positivo, 0] • [a,b] con $a = b$ <ul style="list-style-type: none"> • [0,0] 	Se comprueba que todos los rangos producen un error
getMailConexionCerrada	Se intentan obtener los correos en un rango válido pero la conexión está cerrada	Se produce un error con un mensaje al respecto
getMailCountOk	Se realiza una petición para obtener el número de correos de una carpeta	Se devuelve el número de correos de la carpeta
getMailCountConexionCerrada	Se realiza una petición para obtener el número de correos de una carpeta, pero la conexión está cerrada	Se produce un error con un mensaje al respecto
getCarpetasOk	Se realiza una petición para obtener las carpetas del usuario en el proveedor	Se devuelven las carpetas correctamente
getCarpetasConexionCerrada	Se realiza una petición para obtener las carpetas del usuario en el proveedor, pero la conexión está cerrada	Se produce un error con un mensaje al respecto

GeneradorCSVTest		
Prueba	Procedimiento	Resultado esperado
getFileOk	Se genera un acta en formato CSV	Se devuelve un archivo CSV
getFileIoException	Se intenta generar el acta y se produce una IOException	Se produce un error con un mensaje al respecto
getMimeType	Se intenta obtener el tipo MIME del acta llamando al método "getMimeType"	Se devuelve el string "text/csv"
getExtension	Se intenta obtener la extensión del archivo	Se devuelve el

	llamando al método "getExtension"	string "csv"
--	-----------------------------------	--------------

GeneradorPdfTest		
Prueba	Procedimiento	Resultado esperado
getFileOk	Se genera un acta en formato PDF	Se devuelve un archivo PDF
getFileDocumentException	Se intenta generar el acta y se produce una DocumentException	Se produce un error con un mensaje al respecto
getMimeType	Se intenta obtener el tipo MIME del acta llamando al método "getMimeType"	Se devuelve el string "application/pdf"
getExtension	Se intenta obtener la extensión del archivo llamando al método "getExtension"	Se devuelve el string "pdf"

OptionsManagerTest		
Prueba	Procedimiento	Resultado esperado
errorVacio	Se crea un nuevo OptionsManager con un valor nulo como parámetro	Todas las opciones de exportación devuelven "false"

MailServiceTest		
Prueba	Procedimiento	Resultado esperado
messagingExceptionCerrando	Se intenta cerrar la conexión y se produce una MessagingException	Se lanza una RuntimeException con un mensaje al respecto
getTotalCorreosOk	Se intenta obtener el número de correos de una carpeta	Se devuelve el número de correos en la carpeta
getTotalCorreosConexionCerrada	Se intenta obtener el número de correos de una carpeta, pero la conexión está cerrada	Se lanza una EstadoInvalidoException con un mensaje al respecto
getTotalCorreosMessagingException	Se intenta obtener el número de correos de una carpeta, pero se produce una MessagingException	Se lanza una RuntimeException con un mensaje al respecto
getCorreosOk	Se intentan obtener los correos de un rango válido en una carpeta	Se devuelven los correos pedidos sin errores
getCorreosRangoInvalido	Se intentan obtener los correos de una carpeta en	Se comprueba que todos los rangos

	<p>múltiples rangos inválidos:</p> <ul style="list-style-type: none"> • [negativo, 0] • [0, positivo] • [a,b] con $a > b$ y $a, b > 0$ • [a,b] con $a < b$ y $a, b < 0$ • [negativo, positivo] 	<p>producen una <code>RuntimeException</code></p>
getCorreosConexionCerrada	<p>Se intentan obtener los correos de un rango válido en una carpeta, pero la conexión está cerrada</p>	<p>Se lanza una <code>EstadoInvalidoException</code> con un mensaje al respecto</p>
getCorreosParsingException	<p>Se intentan obtener los correos de un rango válido en una carpeta, pero se produce un error <i>parseando</i> el contenido de los correos</p>	<p>Se devuelve un array de correos vacío</p>
getCorreosMessagingException	<p>Se intentan obtener los correos de un rango válido en una carpeta, pero se produce una <code>MessagingException</code></p>	<p>Se devuelve un array de correos vacío</p>
getCorreosByIdOk	<p>Se intentan obtener los correos generando <code>IdentificationBeans</code> manualmente</p>	<p>Se devuelven los correos pedidos correctamente</p>
getCorreosParaAdjuntar	<p>Se intentan obtener los correos en un rango, pero en modo adjunto (es decir, descargando también los archivos y no sólo sus nombres)</p>	<p>Se devuelven los correos correctamente y con los archivos adjuntos completos</p>
getCarpetasOk	<p>Se intentan obtener las carpetas desde el proveedor</p>	<p>Se devuelven correctamente las carpetas</p>
getCarpetasConexionCerrada	<p>Se intentan obtener las carpetas desde el proveedor, pero no hay una conexión abierta</p>	<p>Se lanza una <code>EstadoInvalidoException</code> con un mensaje al respecto</p>
getCarpetasTipoErróneo	<p>Se intentan obtener las carpetas desde el proveedor, pero estas son de un tipo no soportado</p>	<p>Se devuelven 0 carpetas</p>
getCarpetasVacía	<p>Se intentan obtener las carpetas desde el proveedor, pero estas no contienen mensajes</p>	<p>Se devuelven 0 carpetas</p>
getCarpetasExceptionLecturaInterna	<p>Se intentan obtener las</p>	<p>Se devuelven 0 carpetas</p>

	carpetas desde el proveedor, pero al comprobar si tienen mensajes se lanza una MessagingException	
getCarpetasExceptionLecturaExterna	Se intentan obtener las carpetas desde el proveedor, pero al intentar obtener el listado de ellas se produce una MessagingException	Se lanza una RuntimeException con un mensaje al respecto
getCarpetaByldException	Se intenta obtener una carpeta dado un id que no se corresponde con ninguna	Se lanza una RuntimeException con un mensaje al respecto

AttachmentManagerTest		
Prueba	Procedimiento	Resultado esperado
zipFilesOk	Se intenta comprimir una serie de archivos adjuntos de unos correos dados	Se devuelve un archivo ZIP correcto y sin errores

6.6.2 Pruebas de Integración

Las pruebas de integración se usarán principalmente para realizar diferentes pruebas con el proveedor de correo, sin utilizar una cuenta real, que puede ser bloqueada por un elevado número de peticiones. Usando un servidor simulado específico para estas pruebas se realizarán diferentes peticiones y se comprobarán las respuestas.

La primera serie de pruebas simula la apertura de una conexión, especificando las credenciales. Se realizan peticiones a la API y se comprueba el status de la respuesta. Las situaciones que se van a cubrir son las siguientes:

Abrir conexión			
Correo	Contraseña	Host	Status esperado
Válido	Válido	Válido	200
Válido	Válido	Inexistente	500
Válido	Válido	Vacío	401
Válido	Erróneo	Válido	401
Válido	Erróneo	Inexistente	500
Válido	Vacío	Válido	401
Erróneo	Válido	Válido	401
Erróneo	Válido	Inexistente	500
Erróneo	Erróneo	Válido	401
Erróneo	Erróneo	Inexistente	500
Erróneo	Erróneo	Vacío	401
Erróneo	Vacío	Inexistente	500

Vacío	Válido	Válido	401
Vacío	Erróneo	Inexistente	500

La otra serie de pruebas es la del cierre de la conexión. Aquí se parametrizarán una serie de acciones, se hará la llamada para cerrar la conexión y se obtendrá el status de la respuesta. Los parámetros son “Abrir conexión”, si debe abrirse la conexión antes de hacer la llamada y “Cerrar dos veces”, que especifica si una vez hecha la llamada hay que volver a hacerla.

Cerrar conexión		
Abrir conexión	Cerrar dos veces	Status esperado
Sí	No	200
No	No	409
Sí	Sí	200

6.6.3 Pruebas de las Historias de Usuario

Siguiendo las mismas convenciones que en apartados anteriores, pasamos a mostrar las pruebas relativas a las historias de usuario que se ejecutaron en los diferentes *sprints*. Para cada una se indica la acción a realizar y el resultado que debe mostrarse.

Se recuerda que no sólo se prueba la funcionalidad, sino que también debe evaluarse la experiencia de uso, la apariencia y en definitiva, todo aquello que no permita decir que el incremento realizado sea el que se espera.

6.6.3.1 Historias del Primer Sprint

A continuación, incluiremos las pruebas correspondientes al primer *sprint*, divididas por historias.

Historia 1: Conexión con correo	
Prueba	Resultado Esperado
Abrir una conexión utilizando un correo correcto	Se abre una conexión satisfactoria, mostrándose el mensaje de éxito.
Prueba	Resultado Esperado
Abrir una conexión utilizando un correo existente (Gmail o Outlook) y una contraseña incorrecta.	No se abre ninguna conexión y se muestra un mensaje de error al efecto.
Prueba	Resultado Esperado
Abrir una conexión utilizando un correo de un proveedor desconocido (ni Gmail ni Outlook)	No se abre ninguna conexión y se muestra un mensaje de error indicando que no se reconoce el host.

Historia 2: Desconexión

Prueba	Resultado Esperado
Abrir una conexión con éxito y luego cerrarla	La conexión se abra y se cierra sin errores, mostrando los mensajes adecuados para ambos pasos.
Prueba	Resultado Esperado
Cerrar una conexión sin haberla abierto previamente.	No deberíamos poder, debería mostrarse un mensaje de error u otra indicación similar.

Historia 3: Listar correos	
Prueba	Resultado Esperado
Abrir una conexión y ver los correos recibidos	La conexión debe abrirse y mostrarnos los correos recibidos. Comprobar que los textos se muestran correctamente.

Historia 4: Añadir campos extra	
Prueba	Resultado Esperado
Añadir un título	Debe mostrárenos un campo título que podremos editar.
Añadir una fecha	Debe mostrárenos un campo fecha, con la fecha actual, que podremos editar.

Historia 5: Generar PDF Simple	
Prueba	Resultado Esperado
Seleccionar varios correos y generar un PDF	Deberemos poder descargar un archivo en PDF que contenga los correos.
Seleccionar varios correos, indicar que incluya la fecha de los mismos y generar un PDF	Deberemos poder descargar un archivo en PDF que contenga los correos y sus fechas de envío.
Seleccionar varios correos, añadir fecha y título y generar un PDF	Deberemos poder descargar un archivo en PDF que contenga los correos, un título y la fecha.
Generar un PDF sin seleccionar ningún correo	No deberíamos poder, debería mostrarse un mensaje de error u otra indicación similar.
Seleccionar varios correos, añadir fecha y título, marcar la opción de incluir la fecha y generar un acta.	Deberemos poder descargar un archivo en PDF que contenga los correos con su fecha de envío, un título y la fecha.

6.6.3.2 Historias del Segundo Sprint

De la misma manera que en el primero, para el segundo *sprint* se han redactado estas pruebas funcionales.

Historia 6: Previsualizar contenido correos	
Prueba	Resultado Esperado
Abrir un correo del listado de correos	Se muestra el cuerpo del correo, así como el remitente, el destinatario y la fecha de envío.
Abrir varios correos del	No deben producirse problemas de renderizado y deben

listado de correos	mostrarse todos los correos.
Abrir un correo, cambiar de página y volver a la actual	El correo abierto debe seguir abierto al volver a la página original.

<i>Historia 7: Añadir campos personalizados</i>	
Prueba	Resultado Esperado
Modificar la etiqueta y el valor de un campo por defecto	Al generar el acta deben mostrarse las modificaciones
Desactivar un campo	Al generar el acta no debe aparecer el campo
Desactivar todos los campos	Al generar el acta no debe aparecer ningún campo
Añadir un campo personalizado	Debe añadirse sin problemas, poder modificar sus datos y que estos se reflejen en el acta.
Añadir varios campos personalizados	Deben añadirse sin problemas, poder modificar sus datos y que estos se reflejen en el acta.
Añadir un campo personalizado, modificar sus valores y eliminarlo	El campo no deberá añadirse en el acta

<i>Historia 8: Logos de la aplicación</i>	
Prueba	Resultado Esperado
Acceder a la aplicación	Se muestran los logos correctamente
Reducir el tamaño de la pantalla hasta tamaño móvil.	Se eliminan los logos del pie y los textos de la cabecera.

<i>Historia 9: Cachear correos</i>	
Prueba	Resultado Esperado
En la lista de correos ir a la página 1 y esperar a que cargue. Posteriormente, ir a la 2 y volver a la 1.	Volver a la página 1 deberá tardar menos de 0.5 s.
En la lista de correos ir a la última página y esperar a que cargue. Posteriormente, ir a la 1 y volver a la última.	Volver a la última página deberá tardar menos de 0.5s.

<i>Historia 10: Navegación por páginas no secuencial</i>	
Prueba	Resultado Esperado
Estando en la página 1, tocar el botón de "Siguiente".	Deberá mostrarse la página 2.
Estando en la página 1, tocar el botón de "Anterior"	No debería permitirse la acción.
Estando en la página 1, tocar el botón de "Primera"	No debería permitirse la acción.
Estando en la página 1, tocar el botón de "Última"	Debería mostrarse la última página.
Estando en la página 7 (debe	Deberá mostrarse la página 8.

haber más de 7 páginas), tocar el botón de "Siguiente"	
Estando en la página 7 (debe haber más de 7 páginas), tocar el botón de "Anterior"	Deberá mostrarse la página 6.
Estando en la página 7 (debe haber más de 7 páginas), tocar el botón de "Primera"	Deberá mostrarse la página 1.
Estando en la página 7 (debe haber más de 7 páginas), tocar el botón de "Última"	Deberá mostrarse la última página.
Estando en la última página, tocar el botón de "Siguiente"	No debería permitirse la acción.
Estando en la última página, tocar el botón de "Anterior"	Debería mostrarse la página penúltima.
Estando en la última página, tocar el botón de "Primera"	Debería mostrarse la primera página.
Estando en la última página, tocar el botón de "Última"	No debería permitirse la acción.
Introducir una página existente en el cajetín de búsqueda.	Debería mostrarse esa página existente.
Introducir una página mayor que el número total de páginas en el cajetín de búsqueda.	Debería mostrarse la última página.
Introducir un valor no numérico en el cajetín de búsqueda.	Debería mostrarse la primera página.

Historia 11: Independencia de sesiones

Prueba	Resultado Esperado
Abrir sesión con dos navegadores diferentes (y dos cuentas de correo diferentes) y generar actas válidas.	Ningún dato del segundo navegador se ve en el primero y viceversa. Hay que comprobar también que sólo se cachean correos del usuario propio.

Historia 12: Navegación por páginas no secuencial

Prueba	Resultado Esperado
Ver las fechas de envío de los correos de la primera página y de la última página.	Los correos se muestran en orden descendente de fecha (de más recientes a más antiguos).

Historia 13: Internacionalización del back-end

Prueba	Resultado Esperado
--------	--------------------

Cambiar el idioma a asturiano y generar un acta.	Las fechas y etiquetas estáticas del acta deberán mostrarse en idioma asturiano.
Cambiar el idioma a castellano y generar un acta.	Las fechas y etiquetas estáticas del acta deberán mostrarse en idioma castellano.
Cambiar el idioma a inglés y generar un acta.	Las fechas y etiquetas estáticas del acta deberán mostrarse en idioma inglés.

6.6.3.3 Historias del Tercer Sprint

Para las historias del tercer y último *sprint*, se incluyen estas pruebas.

<i>Historia 14: Exportar a CSV</i>	
Prueba	Resultado Esperado
Crear un CSV con todas las opciones de exportación	Se crea un CSV con todas las columnas pedidas.
Crear un CSV sin seleccionar opciones de exportación	Se crea un CSV con sólo las columnas básicas.
Crear un CSV sin correos	No debería permitirse

<i>Historia 15: Más proveedores de correo</i>	
Prueba	Resultado Esperado
Poner un host correcto manualmente	Se abre una conexión correcta
Poner un host erróneo manualmente	Se muestra un mensaje de error
Poner una dirección de Gmail válida y que el host se calcule automáticamente	Se abre una conexión correcta
Poner una dirección de Outlook válida y que el host se calcule automáticamente	Se abre una conexión correcta
Poner una dirección de Yahoo y que el host se calcule automáticamente	Se muestra un mensaje de error diciendo que no se reconoce el Host

<i>Historia 16: Gestionar múltiples carpetas</i>	
Prueba	Resultado Esperado
Abrir una conexión y moverse entre carpetas	Deben mostrarse los correos de las diferentes carpetas
Generar un acta con correos de varias carpetas diferentes	Debería generarse con todos los correos (sin importar que sean de varias carpetas)

<i>Historia 17: Aviso de Cookies</i>	
Prueba	Resultado Esperado

Entrar en la aplicación (ventana de incógnito)	Debe mostrarse el aviso de cookies.
Cerrar el aviso de cookies y entrar nuevamente a la aplicación	No debe mostrarse el aviso de cookies.
Abrir el aviso de cookies una vez cerrado	Debe mostrarse el aviso de cookies.

Historia 18: Certificado digital HTTPS

Prueba	Resultado Esperado
Abrir una conexión con la aplicación	Ver en el inspector del navegador que las llamadas se hacen por HTTPS

Historia 19: Advertencia Correos Google

Prueba	Resultado Esperado
Mostrar el aviso de Gmail	Debe poder verse el aviso de Gmail

Historia 20: Mostrar adjuntos

Prueba	Resultado Esperado
Ver el cuerpo de los correos	Debe mostrarse los archivos adjuntos de cada correo
Generar un acta	Debe mostrarse los archivos adjuntos de cada correo como texto en el acta
Generar un ZIP con los archivos adjuntos	Debe generarse un ZIP donde estén comprimidos los archivos adjuntos y el cuerpo de los correos, cada uno en un archivo diferente.

Historia 21: Resumir participantes

Prueba	Resultado Esperado
Generar un acta con varios participantes	Deben mostrarse los participantes
Generar un acta con participantes repetidos	Debe mostrarse cada uno una vez
Generar un acta donde alguno de los participantes no tenga nombre establecido, sólo correo	Deberá mostrarse sólo el correo

Historia 22: Botón atrás en la exportación

Prueba	Resultado Esperado
Ir a la pantalla de generación y volver atrás	Deben seguir seleccionados los correos que estaban seleccionados
Ir a la pantalla de generación, volver atrás y volver a la generación	Deben estar seleccionadas las mismas opciones que antes

Historia 23: Mostrar correos en formato HTML	
Prueba	Resultado Esperado
Mostrar un correo con cuerpo en formato HTML	Debe mostrarse el cuerpo en formato HTML
Generar un acta PDF que tenga correos HTML	Debe mostrarse en el PDF con una apariencia similar a como se vería en un navegador.
Mostrar un correo con cuerpo en formato HTML, pero que contenga errores de marcado.	Debe mostrarse en formato HTML
Generar un acta PDF que tenga correos HTML con errores de marcado	El acta no debe contener los correos, sólo el nombre del archivo en la carpeta de adjuntos.

Historia 24: Añadir campo CC	
Prueba	Resultado Esperado
Mostrar un correo con destinatarios CC	Debe mostrarse el destinatario CC.
Mostrar un correo sin destinatarios CC	No debe mostrarse el campo CC.
Generar un acta seleccionado correos CC y la opción de incluir direcciones	Deben aparecer las direcciones CC en los correos seleccionados.
Generar un acta seleccionando correos CC y no seleccionando la opción de incluir direcciones	No deben aparecer las direcciones CC en los correos seleccionados.

Historia 25: Añadir estadísticas participantes	
Prueba	Resultado Esperado
Generar un acta con varios participantes	Deben mostrarse las estadísticas de los participantes
Generar un acta con participantes repetidos	Debe mostrarse cada uno en una fila de la tabla

Historia 26: Desplegar en Docker	
Prueba	Resultado Esperado
Arrancar los contenedores y generar un acta	Debe generarse correctamente

6.6.4 Pruebas Funcionales

Estas pruebas pretenden cubrir una interacción básica de la aplicación ya desplegada en un entorno real. Esto nos causa ciertas limitaciones, que condicionan el diseño de las pruebas. Por ejemplo, no puede abrirse una nueva conexión para cada funcionalidad que podamos cubrir. También debemos parametrizar los *tests* de alguna manera, para configurarlos con los datos reales que haya en el proveedor y se esperan como respuesta.

En este apartado, especificaremos una serie de situaciones a cubrir que posteriormente se agruparán en pruebas y serán los diferentes pasos de cada una.

De la conexión se probará:

- Abrir una conexión correcta
- Cerrar una conexión
- Abrir una conexión (contraseña errónea)
- Abrir una conexión (host erróneo)

De la lista de correos:

- Cambiar de carpeta
- Filtrar correos
- La paginación
 - Ir a la siguiente página
 - Ir a la página anterior
 - Ir a la primera página
 - Ir a la última página
 - Ir a una página específica
- Cargar el cuerpo de un correo
- Marcar un correo para exportación

De la exportación:

- Cambiar el formato del acta
- Campos personalizados
 - Añadir nuevos campos
 - Eliminar campos existentes

Como comentábamos anteriormente, estas situaciones se agruparán en pruebas, intentando crear el menor número de conexiones posibles con el proveedor. Se crean las siguientes pruebas con sus pasos.

Abrir conexión correcta		
1	Abrir conexión correcta	Se abre una conexión con credenciales válidas.
2	Cerrar conexión	Se cierra satisfactoriamente la conexión actual.
Abrir Conexión (Contraseña Errónea)		
1	Abrir conexión (contraseña errónea)	Se intenta abrir una conexión, pero la contraseña es errónea y falla. Se comprueban los mensajes de error.

Abrir Conexión (Host Erróneo)		
1	Abrir conexión (host erróneo)	Se intenta abrir una conexión, pero el host es erróneo y falla. Se comprueban los mensajes de error.
Prueba listado correos		
1	Cambiar carpeta	Se selecciona otra carpeta distinta a la primera y se comprueba que se muestra el número de correos correcto.
2	Filtrar correos	Se filtran los correos de una carpeta utilizando la barra de búsqueda y se comprueba que se muestra el número correcto.
3	Ir a la página siguiente	Se cambia de la página actual a la siguiente y se comprueba que se muestran los correos correspondientes.
4	Ir a la página anterior	Se cambia de la página actual a la anterior y se comprueba que se muestran los correos correspondientes.
5	Ir a la última página	Se cambia de la página actual a la última y se comprueba que se muestran los correos correspondientes.
6	Ir a la primera página	Se cambia de la página actual a la primera y se comprueba que se muestran los correos correspondientes.
7	Ir a la página n	Se cambia de la página actual a una página diferente usando el campo de texto y se comprueba que se muestran los correos correspondientes.
8	Cargar el cuerpo de un correo	Se clica sobre un correo y se comprueba que se muestra su cuerpo.
9	Marcar un correo para exportación	Se marca un correo para exportar y se comprueba que aparece en la lista de seleccionados.
10	Cerrar conexión	Se cierra satisfactoriamente la conexión actual.
Probar exportación		
1	Cambiar a formato CSV	Se selecciona el formato CSV y se comprueba que no se muestran los campos personalizados.
2	Cambiar a formato PDF	Se selecciona el formato PDF y se comprueba que se muestran los campos personalizados.
3	Añadir un campo personalizado	Se añade un nuevo campo personalizado a la lista y se comprueba que se muestra.
4	Eliminar todos los campos personalizados	Se eliminan los campos personalizados que aparecen en la interfaz y se comprueba que no aparece ninguno.
5	Añadir un campo personalizado	Se añade un nuevo campo personalizado a la lista vacía y se comprueba que se muestra.
6	Cerrar conexión	Se cierra satisfactoriamente la conexión actual.

6.6.5 Pruebas de Rendimiento

Se realizarán pruebas de rendimiento sobre la aplicación para comprobar que los procesos principales se realizan en un tiempo razonable. Se tomarán tiempos por medio de la introducción de sondas en el código que permitan precisar los milisegundos transcurridos en diferentes partes de las operaciones. De esta manera se podrán identificar los cuellos de botella y obtener un mayor rendimiento siguiendo el conocido como Principio de Pareto [14]. Las sondas se incluirán en el proceso de lectura de los correos, que es el más tiempo tarda en el uso normal de la aplicación.

Normalmente suelen realizarse también pruebas de carga, pero prescindiremos de ellas por varios motivos. Primero, las sesiones entre usuarios son independientes, por tanto, dos

usuarios nunca interactúan sobre los mismos datos. Tampoco tenemos en este momento una base de datos compartida a la que accedan. Simplemente estaríamos probando cómo reaccionan a la carga computacional los contenedores Docker con las imágenes de del Nginx y del JAR que tenemos funcionando. Sería un punto que tendríamos que probar en un desarrollo real una vez desplegado en un servidor. Poco podríamos hacer sobre el código de la aplicación, sería más bien un asunto a nivel de infraestructura. Por otro lado, el coste de realizar estas pruebas sería alto, pues deberíamos montar un servidor de correo propio, ya que no se pueden realizar muchas peticiones sobre la misma cuenta en un proveedor externo. Y, de hecho, montando un servidor propio podríamos llegar a estar probando la carga que soporta ese servidor, en lugar de la aplicación, pues un escenario más realista sería tener múltiples proveedores. Por todos estos inconvenientes se ha prescindido de las pruebas de carga, centrándose en la inclusión de las sondas que se propone al principio de este apartado.

6.6.6 Pruebas de Accesibilidad

Las pruebas de usabilidad nos proporcionan el punto de vista de los usuarios, como comentamos anteriormente. Sin embargo, además de la experiencia de estos, es importante contar con una serie de criterios, estipulados por expertos en la materia, que nos ayuden a aplicar las mejores prácticas a la hora de comprobar si nuestra aplicación funciona de manera correcta. Para esto, están las pruebas de accesibilidad.

En nuestro caso se harán usando la extensión WAVE – Web Accessibility Evaluation Tool para el navegador Firefox. Aquí comprobaremos que no se reportan errores automáticos y se revisarán manualmente las alertas. También se utilizará una checklist para la revisión manual de la accesibilidad, con el objetivo de realizar una evaluación de conformidad que permita cumplir con los requisitos de las WCAG 2.1.

6.6.7 Pruebas de Usabilidad

El objetivo de las pruebas de usabilidad es obtener información sobre nuestra aplicación desde el punto de vista de los usuarios finales. Queremos saber cosas como: que opinan de las funcionalidades disponibles, si su experiencia de uso es buena o que necesidades tendrían si usasen nuestro software.

En un equipo de desarrollo la opinión sobre la experiencia de uso de la aplicación que este esté desarrollando se encuentra sesgada, eso es innegable. El haber realizado uno mismo la interfaz nos facilita su uso, nos convierte en usuarios expertos de nuestro programa, y además por ser nuestro, la condición humana nos dificulta encontrarle defectos, pues muchas veces establecemos una serie de patrones de uso que tenemos interiorizados, que a nosotros nos resultan naturales y que luego demuestran no ser adecuados para el público objetivo. También somos expertos informáticos y quizás, si estuviéramos diseñando un IDE, podríamos ser unos usuarios de un nivel adecuado, pero en la mayoría de los casos y en especial en este, estamos orientando la aplicación a otras personas con unas habilidades informáticas mucho más reducidas.

Como así lo especificamos en muchísimas partes de este documento y en especial en los Objetivos del Proyecto, queremos una aplicación que destaque por su facilidad de uso y por satisfacer las necesidades de los usuarios de manera eficiente. Por ello, estas pruebas de usabilidad cobran aún más importancia si cabe. El resultado de las mismas debe ser tomado muy en cuenta de cara a futuros *sprints* y todas las recomendaciones deben ser valoradas.

En definitiva, tienen una importancia mayúscula y por eso pondremos mucho esfuerzo en los cuestionarios que se diseñarán a continuación. Si tenemos éxito en esta tarea, obtendremos muchos beneficios, pues las pruebas de usabilidad pueden ser útiles en varios aspectos, por ejemplo:

- Mejorando la eficiencia de los procesos de uso de la aplicación.
- Brindando al usuario una mejor experiencia de uso.
- Disminuyendo el rechazo de los usuarios a un nuevo sistema.
- Disminuyendo las barreras de entrada y facilitando el uso a usuarios novatos.
- Incrementando la comunicación usuario-equipo, vital en metodologías ágiles como la usada en este proyecto.

Para probar todo esto contactaremos con varios usuarios sin ninguna relación con el proyecto. Intentaremos que la muestra sea representativa, es decir, que esté compuesta por personas con diferente experiencia, edad y parecer. Una vez “reclutados”, pasaremos a realizarles unas preguntas sobre su experiencia en el campo de la informática, para precisar ante qué tipo de usuario nos encontramos respecto a su habilidad técnica. También haremos hincapié en su familiaridad con el correo electrónico, ya que nuestra aplicación está orientada a personas que usen este servicio.

Tras esta presentación, pondremos al usuario frente a la pantalla principal de nuestra aplicación. Una vez allí, deberá realizar una serie de acciones que le propondremos. No se le dará ningún tutorial o indicación de uso previo. Sólo se le dará alguna instrucción si se le ve incapaz de avanzar. Un punto que hay que tener en cuenta es que, para una experiencia completa con la aplicación, se debe disponer de una cuenta de correo electrónico. Por motivos de seguridad, algunos servicios de correo, especialmente Gmail, obligan a crear una contraseña específica para aquellas aplicaciones que sean externas. Si el usuario dispone de una o su proveedor de correo no la necesita puede usar su propia cuenta y si no se le proporcionará una. Mientras el usuario ejecuta esa serie de acciones, otra persona, a la que de aquí en adelante nos referiremos como “el responsable”, observará y anotará diferentes aspectos, de acuerdo con el cuestionario que se incluye más abajo.

Cuando el usuario termine de realizar las acciones, también deberá cumplimentar un cuestionario de evaluación, que detallamos en el siguiente punto y donde le preguntaremos por su experiencia de uso.

6.6.7.1 *Materiales de las Pruebas de Usabilidad*

Como hemos indicado anteriormente, las pruebas de usabilidad necesitarán una serie de elementos para su correcto desarrollo. En los siguientes subapartados se incluyen todos ellos.

6.6.7.1.1 Preguntas introductorias

El siguiente cuestionario comprende aquellas preguntas destinadas a precisar la habilidad y familiaridad del usuario con el entorno, así como algunos otros datos. Todas las cuestiones de este punto deberán cumplimentarse al principio de la prueba. Exceptuando las preguntas que indican lo contrario, deberá elegirse una sola respuesta.

<p>1-¿Con qué frecuencia usa usted un ordenador o un smartphone?</p> <ol style="list-style-type: none"> 1. Todos los días 2. Varias veces a la semana 3. Ocasionalmente 4. Nunca o casi nunca
<p>2-¿Qué tipo de actividades realiza? (Pueden elegirse varias respuestas o incluso ninguna)</p> <ol style="list-style-type: none"> 1. Actividades relacionadas con mi trabajo o profesión 2. Aplicaciones de mensajería y/o uso de redes sociales 3. Uso otro tipo de aplicaciones móviles 4. Navegación por páginas web
<p>3-¿Dispone de alguna cuenta de correo electrónico?</p> <ol style="list-style-type: none"> 1. Sí, tengo al menos una cuenta corporativa (del trabajo, universitaria...) y otra personal. 2. Sí, tengo al menos una cuenta corporativa (del trabajo, universitaria...). 3. Sí, tengo al menos una cuenta personal. 4. No dispongo de correo electrónico.
<p>4-¿Con qué frecuencia usa el correo electrónico?</p> <ol style="list-style-type: none"> 1. Todos los días. 2. Lo reviso en algunas ocasiones a lo largo de la semana. 3. Lo uso cuando me lo piden para registrarme en alguna web, para tramitar algún pedido o similar. 4. Nunca o casi nunca.
<p>5-¿Para qué usa el correo electrónico? (Pueden elegirse varias respuestas o incluso ninguna)</p> <ol style="list-style-type: none"> 1. Para mantener conversaciones personales. 2. Para recibir notificaciones o registrarme en páginas web. 3. Para mantener conversaciones en el ámbito laboral/educativo. 4. Otro: _____
<p>6- Marque las afirmaciones con las que está de acuerdo.</p> <ol style="list-style-type: none"> 1. A veces me cuesta encontrar determinados correos electrónicos en mi bandeja de entrada. 2. Suelo descargarme correos electrónicos. 3. Me parece complejo enviar una conversación de correo a otra persona en mi proveedor de correo. 4. Al compartir una conversación de correo electrónico, tengo miedo de enviar a alguna

persona información confidencial por error.

6.6.7.1.2 Actividades guiadas

En esta parte de la prueba, el usuario deberá realizar una serie de acciones sobre la aplicación. A continuación, se enumerarán todas ellas, que deberá llevar a cabo sin ningún tipo de ayuda. Si se quedase atascado, se le darán pistas hasta que pueda completarlas. Son las siguientes:

- Abrir una conexión y posteriormente cerrarla.
- Seleccionar tres correos con una antigüedad de, al menos, un año y generar un acta con ellos.
- Generar un acta con un correo que tenga más de un destinatario.
- Generar un acta introduciendo tres campos personalizados.
- Generar un acta en que se muestren las fechas de los correos.
- Cambiar el idioma y generar un acta.

6.6.7.1.3 Preguntas Cortas sobre la Aplicación y Observaciones

Una vez realizadas las actividades, el usuario que ha ejecutado la prueba deberá responder al siguiente cuestionario:

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Le resulta intuitivo el proceso de generación de un acta, es decir, los diferentes pasos siguen un orden correcto?</i>				
<i>¿Los componentes de la interfaz funcionan como usted se esperaba?</i>				
<i>¿Le resulta fluida la navegación, sin grandes tiempos de espera?</i>				
Calidad de la Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
<i>El tipo y tamaño de letra es</i>				
<i>Los iconos e imágenes usados son</i>				
<i>Los colores empleados son</i>				
Indique si está de acuerdo con las siguientes afirmaciones. 3 (muy de acuerdo), 0 (totalmente en desacuerdo)				
<i>El uso de la aplicación me vendría bien en mi día a día</i>	3	2	1	0
<i>La aplicación necesita más opciones de personalización para las actas</i>	3	2	1	0
<i>La aplicación necesita más funcionalidades para satisfacer mis necesidades</i>	3	2	1	0

<i>La aplicación tiene muchos fallos que deben arreglarse</i>	3	2	1	0
<i>La aplicación es demasiado técnica para mi nivel</i>	3	2	1	0
Opinión personal				
<i>Lo que más me ha gustado</i>				
<i>Lo que menos me ha gustado</i>				
Observaciones				

6.6.7.1.4 Cuestionario para el responsable de las Pruebas

Mientras el usuario realiza las tareas previstas, el responsable de las Pruebas deberá observarlo y tomar notas sobre los siguientes aspectos.

Aspecto Observado	Notas
<i>Tiempo que tarda en comprender la tarea a realizar</i>	
<i>Tiempo en realizar cada tarea</i>	
<i>Necesita ayuda / pistas</i>	
<i>Muestra signos de fastidio o rechazo</i>	
<i>Muestra signos de duda en alguna parte concreta de la aplicación</i>	
<i>El usuario comete errores leves</i>	
<i>El usuario comete errores graves</i>	

Capítulo 7. Implementación del Sistema

Tras el análisis y el diseño es hora de tratar la implementación del sistema. Aquí se reseñará todo aquello destacable durante el desarrollo de esta fase.

7.1 Estándares y Normas Seguidos

Para la implementación se han seguido una serie de estándares que pasamos a numerar.

7.1.1 Java Code Conventions

Parte del sistema se realiza en lenguaje Java, para su implementación se han seguido las convenciones del lenguaje⁷ Aquí se detallan aspectos como el uso de la indentación, el tamaño de las líneas, cómo debe realizarse el nombrado de los métodos y clases, y, en definitiva, todo aquello que se refiere al estilo del código desarrollado. Durante el proceso de creación, se ha usado la herramienta de análisis estático de código SonarLint para comprobar esto.

7.1.2 JSON (JavaScript Object Notation)

Se ha usado este estándar para el traspaso de datos entre el *front-end* y el *back-end*, su uso supone una ventaja frente a otras alternativas, como XML, ya que, al usar Angular, el *parseo* de datos es directo y sencillo, sin necesidad de trabajar con el cuerpo de las peticiones HTTP. JSON es un formato ligero de intercambio de datos [...], de fácil lectura y escritura para los usuarios [...], fácil de analizar y generar por parte de las máquinas [15].

⁷ <https://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

7.2 Lenguajes de Programación

A lo largo de la implementación se han usado diferentes lenguajes. Su enumeración y detalles técnicos, junto a una pequeña reseña se muestran a continuación.

7.2.1 Java

Para el desarrollo del *back-end* se ha usado Java 8. Todas las clases de esta parte de la aplicación y sus pruebas están hechas usando este lenguaje. Su elección es obligada por el uso de SpringBoot, cuya justificación viene en el apartado 2.3.2.3 - Elección del *back-end*.

Es un lenguaje orientado a objetos, fuertemente tipado y uno de los más usados en la actualidad.



Figura 62 – Logo
Java ©Oracle

7.2.2 TypeScript



Figura 63 – Logo
TypeScript
©Microsoft

Los scripts del *front-end* están realizados en este lenguaje. La otra alternativa era el uso de JavaScript del que TypeScript es un superconjunto. Tiene la ventaja de añadir tipos estáticos y objetos basados en clases lo cual permite facilita la creación de un código más seguro, a la par que mantenible.

7.2.3 HTML

Las páginas de la aplicación se definen por medio de HTML (Hypertext Markup Language) 5. Este lenguaje de marcado es mundialmente conocido y permite crear los elementos (textos, imágenes, estructuras...) que conforman las páginas web.



Figura 64 – Logo
HTML ©W3C

7.2.4 CSS



Este lenguaje de diseño gráfico se ha usado para especificar la apariencia de la aplicación. Es complementario a HTML y permite aplicar estilos a los elementos allí definidos. Estos tres últimos lenguajes son las herramientas principales del *framework* de aplicaciones Angular.

Figura 65 –
Logo CSS ©
W3C

7.3 Herramientas y Programas Usados para el Desarrollo

Para el desarrollo del software de la aplicación, se han utilizado una serie de programas. En los siguientes subapartados pasaremos a enumerarlos, describiendo cada uno e indicando sus especificaciones técnicas.

7.3.1 IntelliJ IDEA



Figura 66 – Logo IntelliJ ©JetBrains

Este entorno de desarrollo integrado (IDE) se ha usado para la construcción del *back-end*. Desarrollado por JetBrains, facilita la creación de aplicaciones por medio de avanzadas herramientas de completación dinámica de código, terminal integrada y facilidad para añadir extensiones. Se han usado varios de estos *plugins* para facilitar ciertas tareas. Todos ellos se han obtenido desde el Marketplace de la herramienta.

- CSV Plugin 2.14.3 de Martin Sommer, para la edición sencilla de archivos CSV, que se usaron como entrada de parámetros en las pruebas automatizadas.
- SonarLint 4.12.1.22375 de SonarSource, para el análisis estático de código (linting).
- Sketch It! 0.07.00, para la generación de diagramas en formato PlantUML en base al código programado por medio de ingeniería inversa.
- PlantUML integration 3.6.2 de Eugene Steinberg, para la visualización gráfica de los diagramas UML creados con la extensión anterior.

La versión de este IDE es IntelliJ IDEA 2019.3.2 (Ultimate Edition).

7.3.2 WebStorm



Figura 67 – WebStorm © JetBrains

Para el *front-end* se ha usado este otro entorno de desarrollo. Es la adaptación del anterior al desarrollo de aplicaciones web, en este caso al *framework* Angular. Las características básicas utilizadas son las mismas que en el apartado anterior y las extensiones usadas, las siguientes:

- PlantUML integration 3.6.2 de Eugene Steinberg, para la visualización gráfica de diagramas en formato PlantUML.

7.3.3 Trello

La gestión de las historias de usuario se ha hecho por medio de un tablero en Trello. Esta herramienta online permite el uso de tarjetas para la gestión del trabajo.

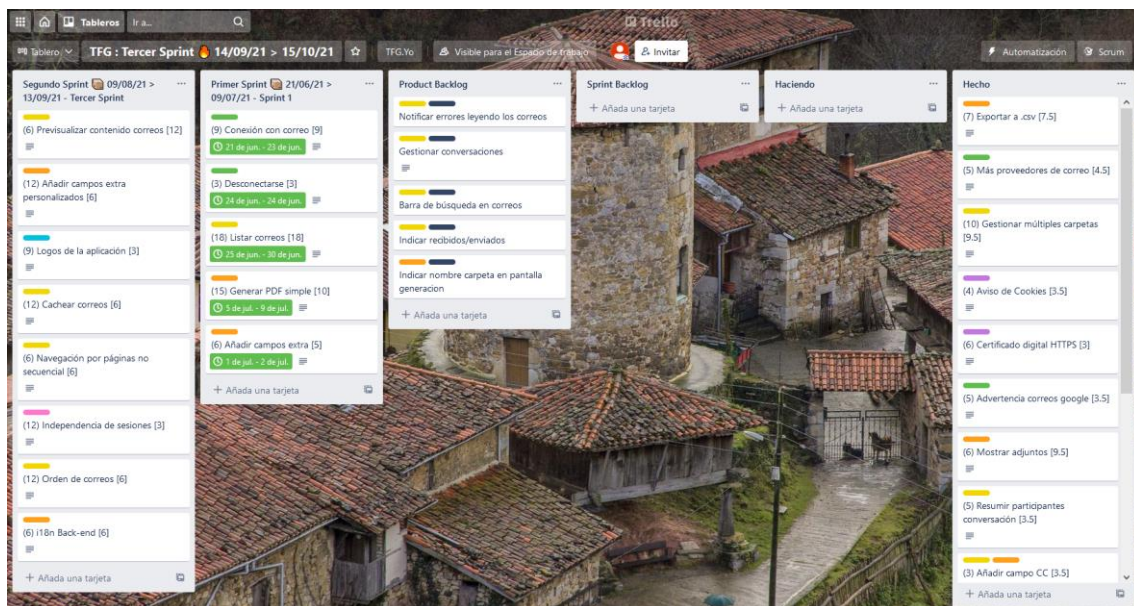


Figura 68 – Tablero Trello del Proyecto

7.3.4 Docker Desktop

También se ha usado Docker Desktop para la gestión de los contenedores de la aplicación, en combinación con las herramientas de línea de comandos.

7.3.5 Git/GitHub

Se ha usado Git con repositorios remotos en GitHub para el control de versiones del código. En concreto se han usado dos repositorios diferentes, uno para el *front-end* y otro para el *back-end*.

7.3.6 Otras herramientas

Además de estas herramientas de “uso diario”, en algunas ocasiones se ha echado mano a otras que han servido para responder a necesidades específicas en algún punto del desarrollo. Estas son:

- Postman 7.36.5, para probar llamadas aisladas a la API, por medio de la especificación manual de los parámetros de la aplicación.
- Para la generación de diagramas PlantUML en el front-end por medio de ingeniería inversa, se ha usado la herramienta tplant⁸ 3.1.0, de Brian Folts.
- Para la conversión de los comentarios de TypeScript a HTML se ha usado la herramienta TypeDoc 0.21.9⁹.

⁸ <https://github.com/bafolts/tplant>

⁹ <https://typedoc.org/>

7.4 Creación del Sistema

Usando las herramientas enumeradas anteriormente se construyó el sistema. En este apartado se incluye todo aquello relacionado con los pormenores de su creación.

7.4.1 Problemas Encontrados

Como en todos los desarrollos de software ha habido problemas que han entorpecido el avance del proyecto y lo han limitado en alcance, gasto y tiempo. A continuación, se describen aquellos de mayor calado.

7.4.1.1 *Uso de JavaMail*

Entre las principales razones para decidir el uso de JavaMail se encontraban su madurez y estabilidad. El largo recorrido de esta librería nos proporcionaba una herramienta fiable para el manejo de los correos electrónicos, siendo un buen pilar para asentar esta aplicación, de la que precisamente son los protagonistas.

Sin embargo, una vez que se empezó el desarrollo surgió un problema con el que no se contaba. Aunque se había visto que la documentación era amplia, casi toda estaba orientada al envío de correos usando JavaMail y no a su lectura. Esto obligó a una formación mucho más amplia y a la necesidad de estudiar la librería a más bajo nivel, teniendo que examinar las clases a partir de su JavaDoc, obligando a realizar algunas pruebas de ensayo y error e incluso llegando a tener que informarse sobre conceptos muy específicos de los propios protocolos de correo.

Por fortuna, que pudieran surgir problemas de este estilo en los primeros días de la implementación era algo con lo que ya se contaba, por ello, las historias del primer *sprint* tienen unas estimaciones más amplias y no se registraron retrasos temporales pese al problema encontrado.

7.4.1.2 *El caso Gmail*

Otro de los problemas que surgió en los primeros días fue el intentar usar cuentas de Gmail. Sucedió que al intentar autenticarse en la aplicación con cuentas de este proveedor se producía un error de credenciales incorrectas. Tras cierto tiempo de investigación, se descubrió que este proveedor tiene unas medidas de seguridad mayores que el resto. La autenticación no puede hacerse con la contraseña propia del usuario y debe obtenerse una específica para la aplicación a través de una herramienta que proporciona Google. Esta se usa posteriormente para la autenticación.

Este problema no frenó el desarrollo, pues era específico de un proveedor y podían usarse muchos otros. Sin embargo, siendo uno de los más usados a nivel mundial, no podíamos ignorarlo. Por ello, se tomaron medidas para informar a los futuros usuarios y se incluyó una

historia en la pila del producto (la Historia H19), que se desarrollaría en el tercer *sprint*, para incluir este aviso.

7.4.1.3 Previsualización de correos

La Historia H6, que pretendía mostrar una serie de datos básicos de los correos, resultó ser más compleja de lo que se pensaba, duplicándose el tiempo que se había previsto para su desarrollo. Aunque la fecha y los correos involucrados eran sencillos de obtener, su cuerpo resultaba difícil de conseguir.

Al contrario de lo que se pensaba, el contenido de los correos no era HTML o texto, si no una compleja unión de partes, cada una con diferentes tipos MIME. Para correos que contuviesen unas pocas líneas sin mucho formato, esto resultaba fácil, sin embargo, a la hora de probarlo con correos reales, muchas veces se producían errores de *parseo*. Visto el retraso que esto producía, se decidió mostrar el texto de los correos únicamente, sin contenido multimedia y abordar en una historia futura el conseguir mostrar todo el cuerpo, tal y como se vería en un cliente de correo. Esto se logró completando la Historia H23.

7.4.1.4 Independencia de sesiones

La Historia H11 surge de un problema detectado en las pruebas del primer *sprint*. Se daba el caso de que un usuario podía ver los correos de otro usuario que estuviese en una sesión diferente, siendo esto un gravísimo problema de seguridad.

En el desarrollo de esta historia se descubrió que los componentes de SpringBoot que se inyectan por medio de la inversión de dependencias actúan siguiendo el patrón Singleton, es decir, tienen una única instancia. Por tanto, los correos guardados se podían acceder desde distintas sesiones. Esto fue fácilmente subsanado en el *back-end* cambiando el *scope* del servicio que almacena correos a "Session". De esta manera, cada usuario tendría sus propios datos.

Sin embargo, esto trajo consigo problemas en el *front-end*, que no recuperaba los datos de peticiones anteriores y creaba una nueva sesión en cada una, haciendo imposible la conexión con el proveedor. Esto se subsanó tras cierta investigación modificando los parámetros de las peticiones para mantener el token identificativo de la sesión.

7.4.1.5 El locale asturiano

La Historia H13 pretendía internacionalizar las actas generadas, usando los textos adecuados en función del idioma a la hora de imprimir los correos. Debían localizarse a los diferentes idiomas de la aplicación términos como "fecha", "remitente" o "destinatario". También había un pequeño detalle que era localizar el valor de la fecha de envío de los correos en sí. En Java esto es sencillo usando los Locales que vienen por defecto, que contienen los símbolos de fecha para cada idioma (nombres de los días y los meses, orden de los elementos...). Sin embargo, el Locale para el idioma asturiano no cuenta con esos símbolos, por tanto, hubo que

crear a mano algunas clases que los implementasen, incrementando esto el tiempo para el desarrollo de la historia en 9 horas por encima de las 6 previstas.

7.4.1.6 Resolución de hosts

Para establecer una conexión con un proveedor de correo utilizando JavaMail debe proporcionarse: la dirección de correo, la contraseña y el host, es decir, la dirección del servidor de correo del proveedor en que se alojan los mensajes recibidos por el usuario. En los primeros incrementos de este producto, se soportaban sólo un par de proveedores de correo, que tenían sus hosts *hardcodeados* en el código. Posteriormente, se intentó ampliar a todos los posibles, suponiendo que se podía obtener el host del proveedor desde la dirección de correo. Esto no era correcto.

Si bien se puede obtener, dada una dirección de correo cualquiera, el host, usando por ejemplo el comando “nslookup” en CMD. La dirección que se devuelve es la de envío, es decir, el servidor al que tendrías que enviar los correos, que puede ser el mismo, o no que al que debes acceder para leerlos. Esto no nos aseguraba que un sistema que resolviese automáticamente los hosts iba a devolvernos el que necesitábamos para la lectura. Por este motivo, se decidió añadir una lista configurable con las direcciones soportadas, y permitir a los usuarios especificar su host manualmente. Todas estas modificaciones se hicieron en la Historia H15.

7.4.1.7 Correos HTML

Uno de los problemas anteriores detallaba la dificultad de mostrar el cuerpo de los correos en formato HTML. En la Historia H23 se abordó esto creando la clase MailParser, que centraliza el *parseo* de los correos desde el proveedor. Aquí se pretendía ir leyendo las diferentes partes del correo, obteniendo el HTML de las partes que tuviesen este tipo, los archivos adjuntos desde otras... Después, se mostraría el HTML obtenido en la interfaz y en el PDF. Los problemas surgidos aquí fueron muchos.

A la hora de incluirlo en el *front-end* fue sencillo, Angular proporciona utilidades para esto. Sin embargo, la traducción a PDF es más complicada. Primero hubo que conseguir una librería que realizase la conversión. Esto funcionaba bien con algunos correos, pero con otros fallaba, pues tenían errores (etiquetas abiertas, propiedades no válidas de XML...). Para ello hubo que incluir una segunda librería que los limpiase. Tras esto, algunos correos aún daban errores a causa de problemas con los estilos, pues ciertas propiedades de CSS que contenían los correos reducían algunos elementos a unas dimensiones de 0px y esto no es soportado por la librería que genera los PDF. Para solventar estos últimos que aún daban error, se decidió incluir en la exportación de los archivos adjuntos también el cuerpo de los correos, de esta manera, se podrían abrir con un navegador, que sí renderiza todo el HTML sin problemas.

7.4.1.8 *Tiempos de carga de adjuntos*

La Historia H20 también resultó más compleja de lo que se pensaba. A la hora de exportar los archivos adjuntos se pretendía que se descargasen en la fase de lectura, cuando se recuperan todos los correos. Sin embargo, una vez implementado, los tiempos de carga de los correos se duplicaban, llegando a tiempos de espera inasumibles.

Para solucionar esto, se decidió realizar en dos fases. A la hora de leer los correos, se descargaría sólo el nombre de los adjuntos, mientras que, en la fase de exportación, se volvería a acceder al proveedor y se descargarían los archivos. Esta segunda aproximación evita que se descarguen archivos de correos que no serán exportados, reduciendo los tiempos de espera, que además se trasladan de una fase ya lenta de por sí, como es la lectura de correos, a otra más propensa a la espera como es la exportación. Por el contrario, hay que realizar una segunda conexión con el proveedor para obtener los mismos correos, introduciendo carga computacional. Tras valorar estos pros y contras se decidió implementar esta segunda opción.

7.4.1.9 *Doble factor de autenticación UNIOVI*

A principios del mes de marzo y con el desarrollo del proyecto ya finalizado, la Universidad de Oviedo activó el doble factor de autenticación obligatorio para el acceso al correo corporativo. Tras esto, nuestra aplicación dejó de funcionar con cuentas de la Universidad. Ante este hecho nos pusimos en contacto con el CAUSI (Centro de Atención al Usuario), y nos comunicaron que: “Por el momento el acceso al correo debe realizarse a través de las aplicaciones corporativas que son las que permiten la autenticación de doble factor”.

Si bien esto nos hizo perder la compatibilidad (quizás temporal) con nuestro principal proveedor, se siguió manteniendo la compatibilidad con el resto, bien sea por medio de los servidores que se resuelven automáticamente o introduciéndolos manualmente.

Capítulo 8. Desarrollo de las Pruebas

Planteadas en el Capítulo 5 y diseñadas en el Capítulo 6, los resultados de los diferentes tipos de prueba pasan a detallarse a continuación. En función del tipo se ejecutaron en un momento u otro del desarrollo, en base al calendario de pruebas definido en la Planificación Inicial. Junto a los resultados de las pruebas se incluyen las acciones tomadas para subsanar los problemas encontrados.

8.1 Pruebas Unitarias

Las pruebas unitarias se ejecutaron múltiples veces a lo largo del desarrollo, por tanto, no es pertinente la inclusión aquí de un informe detallado de todos los errores encontrados en su ejecución. Se utilizaron como una herramienta más de trabajo, señalando partes defectuosas que se fueron corrigiendo. Al intentar realizar una programación al estilo TDD (*Test-Driven Development*), el desarrollo de las pruebas, del código y su ejecución van de la mano. Al final del desarrollo todas las pruebas unitarias se ejecutaban correctamente.

Para asegurarse de que las pruebas eran significativas, se utilizaron algunas herramientas incluidas en los IDE y en los marcos de pruebas utilizados para medir el *coverage*. Esta métrica resultó muy satisfactoria llegando a cubrir el 100% de las líneas, funciones y ramas posibles en el *front-end* y alrededor de un 80% en el *back-end*. Esta disminución no es importante, pues tiene que ver con funciones para las que la automatización de las pruebas tiene un coste muy alto, en comparación con su ejecución manual. Por ejemplo, en el caso de la generación del PDF con el acta. El correcto funcionamiento de estas funcionalidades está cubierto por otro tipo de pruebas.

8.2 Pruebas de Integración

De la misma manera, las pruebas de integración se ejecutaron en múltiples ocasiones, generalmente junto a las pruebas de usabilidad. Los fallos que se detectaron se arreglaron en su momento, en todo caso, menores y siempre antes del cierre de cualquier historia de usuario. Al final del desarrollo todas las pruebas de integración obtenían los resultados correctos.

8.3 Pruebas de las Historias de Usuario

Las pruebas a partir de las historias de usuario se realizan sobre la interfaz final de cada una, de manera manual y proporcionan un punto de vista diferente a otras pruebas automáticas. Siempre es importante realizar las interacciones nosotros y ver cuál es la experiencia. La desventaja es que llevan mucho más tiempo que las automáticas y, por tanto, se realizan en menos ocasiones. El desglose fue por historias y una vez que se tenía desarrollada, se pasaba a la realización manual de las pruebas. De no salir el resultado esperado, se corregía y se volvían

a ejecutar. De esta manera se detectaron algunos errores. Todo esto forma parte del desarrollo normal de la historia y se considera dentro de los problemas encontrados en el desarrollo y en el tiempo dedicado a cada una.

8.4 Pruebas Funcionales

Las pruebas funcionales tienen una consideración similar a las unitarias y a las de integración. Nos sirven como una potente herramienta que comprueba que el producto que desplegamos no produce fallos en una ejecución estándar. Estas se ejecutaron tanto en los momentos en que el código se desplegaba en local, como cuando se crearon los contenedores Docker. Se mantienen actualizadas y se entregan con el código final.

8.5 Pruebas de Rendimiento

Las pruebas de rendimiento se ejecutaron al final del desarrollo. Consistieron en ir avanzando por las diferentes páginas del paginador y viendo cuánto tiempo se tardaba en cargar los correos. Esto nos permite comprobar que se cumple el RNF9 de la lista de Tabla 15 – Requisitos funcionales del tercer sprint

Requisitos No Funcionales. Se colocaron cuatro sondas en diferentes puntos del código y se tomó el tiempo transcurrido (en milisegundos) para realizar la operación. Los puntos elegidos fueron:

- Component: Calcula el tiempo transcurrido entre que se realiza el cambio de página y se muestran los correos.
- Service: Es el tiempo entre que se realiza la petición desde el *front-end* al *back-end* y se obtiene una respuesta.
- Controlador: Mide el tiempo transcurrido entre que se recibe una petición en el *back-end* y se devuelve una respuesta.
- Mail S.: Mide el tiempo que se tarda en pedir al proveedor los correos de un rango y obtener una respuesta.

Con los datos obtenidos se realizaron una serie de cálculos, los que se muestran en la columna tiempos, donde se indica:

- Proveedor: El tiempo que tarda el Proveedor en devolvernos una respuesta.
- Back: El tiempo que se tarda en procesar los datos en el *back-end*.
- Serv: El tiempo que tardan en procesarse los datos en el *front-end*.
- Comp: El tiempo que tardan en renderizarse los datos en los componentes.

Pág	Sondas				Tiempos			
	Component	Service	Controlador	Mail S.	Proveedor	Back	Serv	Comp
1->2	3889	3889	3876	3773	3773	103	13	0

2->3	3992	3992	3981	3714	3714	267	11	0
3->4	3611	3611	3603	3600	3600	3	8	0
4->5	3984	3984	3977	3976	3976	1	7	0
5->6	4137	4137	4128	4052	4052	76	9	0
6->5	85	85	74	0	0	74	11	0
5->4	79	79	71	0	0	71	8	0
4->3	89	89	82	0	0	82	7	0
3->2	84	84	78	0	0	78	6	0
2->1	102	102	95	0	0	95	7	0
Promedio	2005	2005	1997	1912	1912	85	9	0

Tabla 17 – Resultados de las pruebas de rendimiento

De estos resultados se pueden sacar varias conclusiones. Por un lado, que el mayor retraso se produce en el proveedor, con unos tiempos muy altos de carga, que, en las segundas llamadas, al almacenarse en caché los correos ya leídos, se reducen a cero. El resto de tiempo de procesamiento es prácticamente imperceptible para el usuario.

Aunque estas pruebas nos permiten comprobar que se cumple con los requisitos planteados, los tiempos de carga en las primeras llamadas se destacan como punto a mejorar, antojándose muy elevados y siendo superiores a los 3,5 segundos. En futuras actualizaciones del sistema debería trabajarse en esto.

8.6 Pruebas de Accesibilidad

Las pruebas de accesibilidad se realizaron al final del desarrollo con el objetivo de no sólo mejorar el uso de la aplicación para el público general e incrementar también el rango de usuarios que pueden utilizarla, también para cumplir con uno de los Tabla 15 – Requisitos funcionales del tercer sprint

Requisitos No Funcionales de la aplicación. Una vez analizada la accesibilidad en dos pasos (uno automático y otro manual), se corrigieron los errores que se encontraron.

8.6.1 Revisión Preliminar

Primero se realizó una revisión automática de la aplicación usando WAVE – Web Accessibility Evaluation Tool para el navegador Firefox. Por ser una SPA sólo hace falta ejecutar el análisis automático sobre una página. Los resultados obtenidos fueron estos:

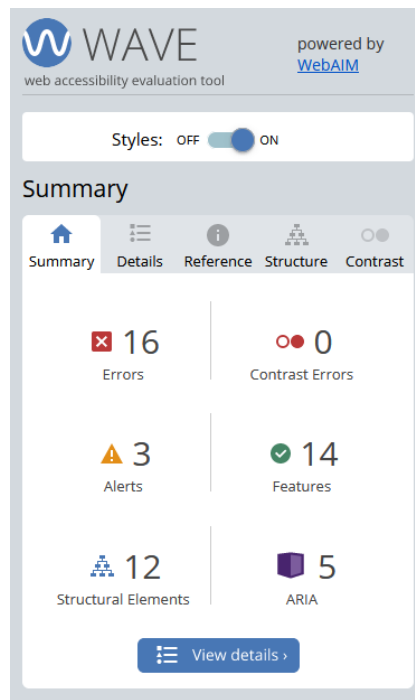


Figura 69 – Resumen de los resultados de las pruebas con WAVE

Se encontraron 16 errores y 3 alertas, que se pasaron a revisar manualmente:

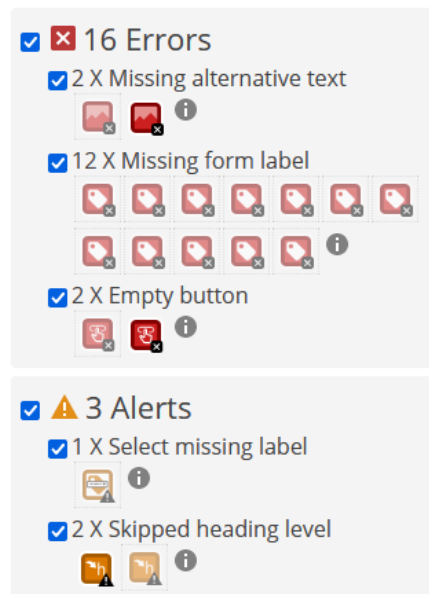


Figura 70 - Detalle de los errores encontrados en las pruebas con WAVE

- Cuatro de los errores, los de texto alternativo y botón vacío corresponden a un par de botones, en concreto el de mostrar el aviso de cookies y el de volver al listado. Faltaba añadirles la propiedad “alt”. Una vez añadida se solucionaron ambos.
- Seis de los errores de etiquetas faltantes corresponden a olvidos de establecer el atributo “for”. Se subsanó incluyéndolo.

- Los seis errores restantes son falsos positivos. Por el propio diseño de la interfaz esto no puede considerarse un error, ya que lo que sería la “etiqueta” de este formulario, es un input editable, pues el objeto de este formulario es definir las etiquetas del acta.

Datos que se añadirán al acta:



<input checked="" type="checkbox"/>	Título		
<input checked="" type="checkbox"/>	Fecha	09/11/2021, 16:45:47	

Figura 71 - Detalle de la interfaz

Pasando a las alertas, también son fácilmente subsanables:

- Son dos errores en el marcado de los elementos. Pues se usan “h6” en lugar de “label”. Una vez modificado ya no se reporta ninguna alerta, pues la tercera de ellas se soluciona al establecer uno de esos “h6” como “label” de un elemento “select”.

Con esto se termina la revisión automática.

8.6.2 Revisión Manual WCAG 2.1

Tras detectar los errores más evidentes en el cumplimiento de las directrices del WCAG con la revisión automática, se procedió a realizar una revisión manual, más profunda y detallada. Para ello se fueron revisando las directrices de la versión 2.1, la última disponible¹⁰.

La siguiente tabla va realizando un repaso sobre las directrices y especificando si se cumplen o no.

ID	Nivel	Título	Status	Observaciones
1 -> Perceptible				
1.1	Texto Alternativo			
1.1.1	A	Contenido no textual	OK	Se cumple en todos los casos, por medio del uso de título y etiquetas. Hay una excepción (de las que permite esta directriz) que son las etiquetas, cuyo propósito se deduce de elementos visuales establecidos por defecto.
1.2	Contenido multimedia en tiempo real			
1.2.1	A	Solo-audio y solo-video (Pregrabado)	N/A	No hay contenido multimedia en tiempo real (video, audio). Lo mismo para el resto de directrices de este bloque.
1.2.2	A	Subtítulos (Pregrabado)	N/A	
1.2.3	A	Audiodescripción o	N/A	

¹⁰ <https://www.w3.org/WAI/WCAG21/quickref/>

		contenido multimedia alternativo (Pregrabado)		
1.2.4	AA	Subtítulos (En directo)	N/A	
1.2.5	AA	Audiodescripción (En directo)	N/A	
1.2.6	AAA	Lenguaje de signos (Pregrabado)	N/A	
1.2.7	AAA	Audiodescripción extendida	N/A	
1.2.8	AAA	Contenido multimedia alternativo (Pregrabado)	N/A	
1.2.9	AAA	Sólo audio (En directo)	N/A	
1.3	Adaptable			
1.3.1	A	Información y relaciones	OK	Se cumple proporcionando una estructura adecuada.
1.3.2	A	Secuencia expresiva	OK	Los elementos guardan el orden correcto.
1.3.3	A	Características sensoriales	OK	Se proporcionan textos alternativos para elementos visuales.
1.3.4	AA	Orientación	OK	La aplicación se puede usar sin problemas tanto en vertical como en horizontal.
1.3.5	AA	Identificación del propósito de las entradas	Parcial ³	Los únicos datos del usuario pedidos son email y contraseña. Se usa el atributo "type" pero no el "autocomplete".
1.3.6	AAA	Identificación de propósitos	Parcial ⁵	Sólo se usan puntos de referencia ARIA para los <i>spinners</i> de carga.
1.4	Distinguishable			
1.4.1	A	Uso del color	OK	Los colores sólo aportan más información o son puramente decorativos, nunca se usan como única fuente de información.
1.4.2	A	Control de audio	N/A	No hay audio.
1.4.3	AA	Contraste (mínimo)	OK	Validado de manera automática.
1.4.4	AA	Redimensionar texto	OK	Se puede redimensionar sin problemas (incluso hasta un 500%).
1.4.5	AA	Imágenes de texto	N/A	No hay imágenes que muestren texto.
1.4.6	AAA	Contraste (mejorado)	OK	Validado de manera automática.
1.4.7	AAA	Audio de fondo bajo o no existente	N/A	No hay audio.
1.4.8	AAA	Presentación visual	OK	Se cumplen todas las condiciones para el texto que aparece, aunque no hay grandes párrafos, que es a lo que está destinada esta directriz.
1.4.9	AAA	Imágenes de texto (Sin excepción)	N/A	No hay imágenes que muestren texto.
1.4.10	AA	Redistribución	OK	No se requiere desplazamiento horizontal.
1.4.11	AA	Contraste de elementos no textuales.	OK	Validado de manera automática.

1.4.12	AA	Espaciado del texto	OK	Validado con el Text Spacing Booklet ¹¹ .
1.4.13	AA	Contenido al pasar por encima o poner el foco	OK	No se muestra contenido extra de este estilo, solo se producen cambios decorativos (p. ej. oscurecer color).
2 -> Operable				
2.1	Accesible con teclado			
2.1.1	A	Teclado	OK	Se pueden operar todos los elementos usando sólo el teclado.
2.1.2	A	Sin trampas de teclado	OK	No hay ningún elemento del que no se pueda quitar el foco.
2.1.3	AAA	Teclado (Sin excepciones)	OK	Ídem que en 2.1.1.
2.1.4	A	Atajos de un carácter	N/A	No se han implementado atajos de teclado.
2.2	Tiempo suficiente			
2.2.1	A	Tiempo ajustable	N/A	No hay límites de tiempo implementados.
2.2.2	A	Pausar, parar, ocultar	N/A	
2.2.3	AAA	Sin tiempo	N/A	
2.2.4	AAA	Interrupciones	N/A	
2.2.5	AAA	Volver a autenticarse	No OK ⁶	Si la sesión expira, a la hora de reconectarse los datos se pierden
2.2.6	AAA	Caducidades	No OK ⁷	No se informa al usuario de la duración de la sesión
2.3	Convulsiones y reacciones físicas			
2.3.1	A	Tres flashes o por debajo del límite	N/A	La aplicación no emite flashes.
2.3.2	AAA	Tres flashes	N/A	La aplicación no emite flashes.
2.3.3	AAA	Animación a causa de interacciones	N/A	No hay animaciones que impliquen desplazamiento de elementos.
2.4	Navegable			
2.4.1	A	Saltar bloques	N/A	No hay bloques que se repitan, no es necesario este tipo de navegación
2.4.2	A	Páginas tituladas	OK	Hay sólo una página, pero contiene título
2.4.3	A	Enfoque ordenado	OK	La navegación con teclado permite moverse de manera ordenada.
2.4.4	A	Propósito de los enlaces (en contexto)	OK	El único enlace que hay es auto explicativo mirando el texto.
2.4.5	AA	Múltiples opciones	N/A	No hay múltiples páginas
2.4.6	AA	Encabezados y etiquetas	OK	Los encabezados describen correctamente el contenido.
2.4.7	AA	Foco visible	OK	El propio navegador ya establece esto.
2.4.8	AAA	Localización	N/A	No hay múltiples páginas
2.4.9	AAA	Propósito de los enlaces (Sólo enlace)	OK	El único enlace cumple con esta directriz.
2.4.10	AAA	Encabezados de sección	OK	Todas las secciones están identificadas correctamente por medio de títulos y

¹¹ <https://codepen.io/stevef/full/YLMqbo>

				etiquetas.
2.5	Tipos de entrada			
2.5.1	A	Gestos de puntero	N/A	No se utilizan interacciones que requiera varios clics o que estén basadas en recorrer caminos.
2.5.2	A	Cancelación del puntero	OK	La funcionalidad no se ejecuta con el evento <i>click-down</i> y se pueden abortar los clics.
2.5.3	A	Etiqueta en nombre	OK	El texto visible se corresponde con el nombre accesible de los controles.
2.5.4	A	Actuación por movimiento	N/A	No hay componentes que se operen por movimiento.
2.5.5	AAA	Tamaño del objetivo	No OK ⁸	Varios elementos son menores de 44x44 (iconos, botones paginador, selector de carpetas, checkboxes...)
2.5.6	AAA	Mecanismos de entrada concurrentes	OK	No se restringen mecanismos y se usan manejadores de eventos de alto nivel (focus y clic).
3 -> Entendible				
3.1	Legible			
3.1.1	A	Idioma de la página	No OK ¹	El idioma de la página es estático, marcando siempre "en".
3.1.2	AA	Idioma de las partes	N/A	No hay bloques de texto en un idioma diferente al genérico de la página.
3.1.3	AAA	Palabras inusuales	N/A	No aparecen palabras de este tipo.
3.1.4	AAA	Abreviaturas	No OK ⁹	Aparece la abreviatura CC y no su forma ampliada.
3.1.5	AAA	Nivel de lectura	N/A	No aparece texto complejo
3.1.6	AAA	Pronunciación	N/A	No hay palabras de pronunciación ambigua.
3.2	Predecible			
3.2.1	A	Al poner el foco	OK	No se producen cambios sólo al poner el foco.
3.2.2	A	Al introducir información	OK	No se producen cambios de este estilo.
3.2.3	AA	Navegación consistente	N/A	No hay múltiples páginas.
3.2.4	AA	Identificación consistente	N/A	No hay múltiples páginas.
3.2.5	AAA	Cambio en petición	OK	No se producen cambios de contexto sin una acción directa del usuario.
3.3	Asistencia en la introducción de datos			
3.3.1	A	Identificación de errores	OK	Cuando falla una conexión se muestran mensajes de error adecuados.
3.3.2	A	Etiquetas o instrucciones	OK	Se proporcionan etiquetas adecuadas y se relacionan con los campos de los formularios.
3.3.3	AA	Sugerencia de errores	OK	Se proporcionan sugerencias, p.ej., cuando falla la conexión a Gmail.
3.3.4	AA	Prevención de errores	N/A	No se pide información sensible de este

		(legales, financieros, datos)		tipo.
3.3.5	AAA	Ayuda	No OK ¹⁰	No se proporciona contenido de ayuda.
3.3.6	AAA	Prevención de errores (todos)	No OK ¹¹	No se proporcionan diálogos de confirmación.
4 -> Robusto				
4.1	Compatible			
4.1.1	A	Parsing	OK	Se ha validado el HTML y es correcto.
4.1.2	A	Nombre, rol y valor	Parcial ²	Falta incluir etiquetas "aria-label" en aquellos elementos que no contienen una etiqueta visible.
4.1.3	AA	Mensajes de estado	Parcial ⁴	Se utiliza el role "status" solo para los <i>spinners</i> de carga.

Tabla 18 – Resultados de las pruebas de accesibilidad

Tras esta revisión manual, se obtienen los siguientes datos. En general, la accesibilidad a nivel general es buena, aunque quedan por pulir algunos detalles para alcanzar el nivel requerido.

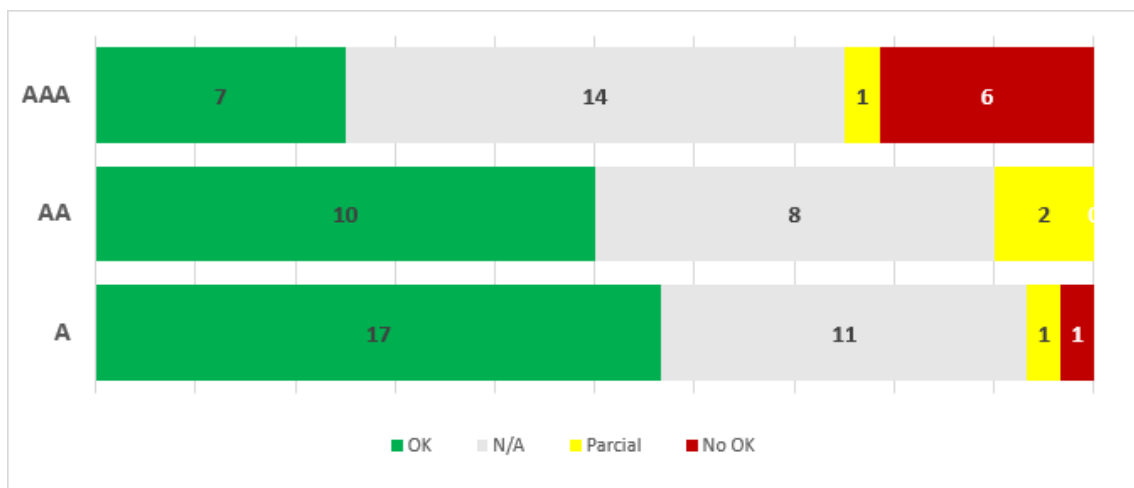


Figura 72 – Resumen del cumplimiento de las directrices WCAG por niveles tras el análisis manual de la accesibilidad.

Para subsanar los problemas se realizaron los siguientes cambios sobre la aplicación. Por el contrario, para otras directrices fallidas se argumenta el prescindir de ellas. Cada uno de los cambios que se van a mostrar en la siguiente lista tiene un identificador que se corresponde con el superíndice asignado en la columna "status" de la tabla anterior.

Nivel A

1 – Se añaden un par de líneas de código que establecen la propiedad "lang" del HTML al cambiar el idioma y al iniciar la aplicación.

2 – Se incluyen las etiquetas necesarias. Con esto también se solventan los problemas que se producían en la evaluación preliminar automática.

Nivel AA

3 – Se incluye el atributo “autocomplete” que faltaba.

4 – Se añaden más roles en elementos donde se necesitaba, p.ej. en los mensajes de estado al abrirse o cerrarse una conexión.

Nivel AAA

5 – Se añade algún “Landmark Role” más, p.ej. “banner” en el logo de la aplicación. No se necesitan añadir muchos, gracias a un correcto uso de los elementos semánticos que proporciona HTML5.

6 y 7 – Se decide no aplicar estas dos directrices, por motivos de recursos. Por un lado, sería muy costoso implementar un sistema que almacenase los datos introducidos por el usuario en tiempo real, obteniendo muy poco beneficio, pues sólo se necesitan unos pocos datos para generar un acta y no es un proceso temporalmente largo. Por otro, el tiempo que dura la sesión debería ser algo a precisar una vez que se despliegue en un servidor real en función de la carga que este puede soportar y los recursos que se destinan a ello.

8 – Hay gran cantidad de elementos menores del tamaño requerido. Cumplir esta directriz obligaría a la modificación de todos los elementos de las interfaces. Se decide dejar esta mejora para futuros incrementos del sistema.

9 – Se soluciona por medio de la modificación de los textos internacionalizados.

10 – Desarrollar una serie de instrucciones de ayuda es un desarrollo costoso, que implicaría muchas horas. Estaría bien incluirlo en futuros incrementos. No se considera necesario en este punto, ya que se proporciona un “Manual de usuario” en esta misma documentación.

11 – La información que requeriría confirmación son los valores especificados en el componente de exportación. Por ser una información poco sensible que desencadena un proceso fácilmente reversible, por mostrarse esta en pantalla en todo momento y por mantenerse también tras el proceso se decide no seguir esta directriz.

Tras estos cambios y decisiones el resumen es el que sigue.

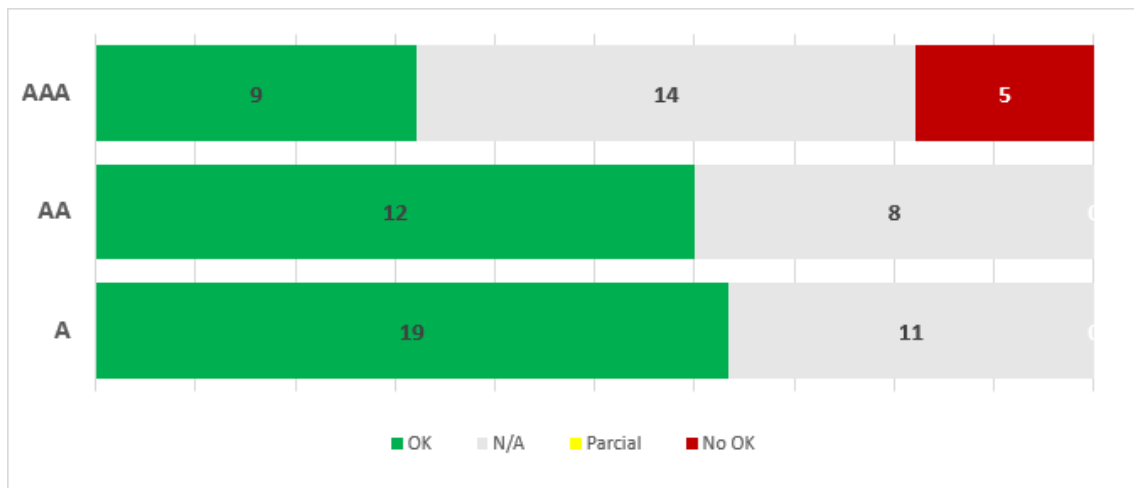


Figura 73 – Resumen del cumplimiento de las directrices del WCAG por niveles tras los cambios realizados.

Tras estas pruebas se puede certificar que se cumple el Nivel AA de accesibilidad y que el Nivel AAA está muy cerca de cumplirse. Futuros desarrollos del sistema deberían trabajar hacia el cumplimiento de las directrices restantes, que permitan poder alcanzarlo.

8.6.3 Accesibilidad con Dispositivos Móviles

Hoy en día es vital lograr un diseño responsivo que permita utilizar las aplicaciones web en dispositivos móviles. Para comprobar que las interfaces funcionan en estos dispositivos, que se caracterizan por las reducidas dimensiones de sus pantallas, se realizarán unas pruebas utilizando el simulador de Firefox. Este navegador tiene una herramienta que permite emular distintos dispositivos y mostrarnos como se vería en ellos nuestra aplicación. Se simularán los siguientes dispositivos:

- iPhone 11 Pro iOS 14.6
- Galaxy Note 20 Linux
- iPad

Las pruebas consistirán en examinar las interfaces de la aplicación y detectar los posibles problemas que puedan surgir.

Tras probar en los tres dispositivos, e incluso cambiando la orientación de la pantalla (vertical/horizontal) se han encontrado varios problemas, que se enumeran a continuación, junto a la solución adoptada.

Problema 1 - Aviso de cookies cortado

Al reducir la pantalla, el texto del aviso de cookies aparece oculto debajo del botón. Esto impide la lectura correcta del mensaje, como puede verse en la Figura 74.

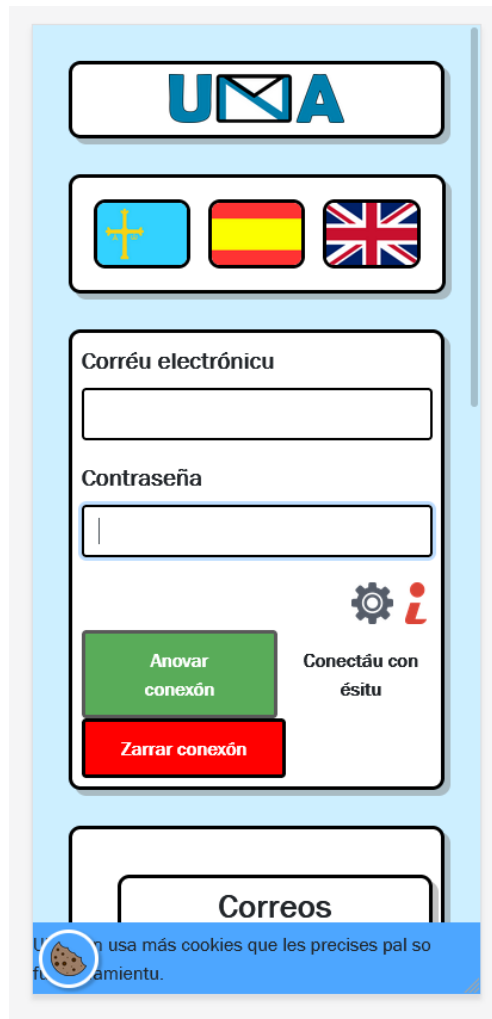


Figura 74 – Problema 1 (Detección)

Modificando algunos valores de CSS se consigue que el texto se muestre a la derecha del botón, tal y como se ve en la Figura 75.



Figura 75 – Problema 1 (Solución)

Problema 2 - Error de centrado

En algunas ocasiones el componente que lista los correos no aparece centrado respecto a su elemento padre.

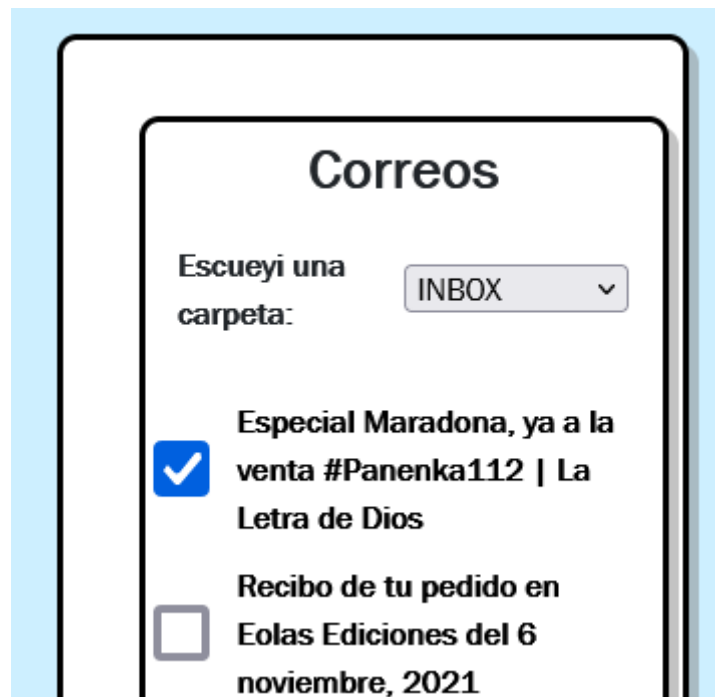


Figura 76 – Problema 2 (Detección)

El problema se debía a que no estaba establecida correctamente la propiedad “overflow” del asunto de los correos. Una vez solucionado aparece así.

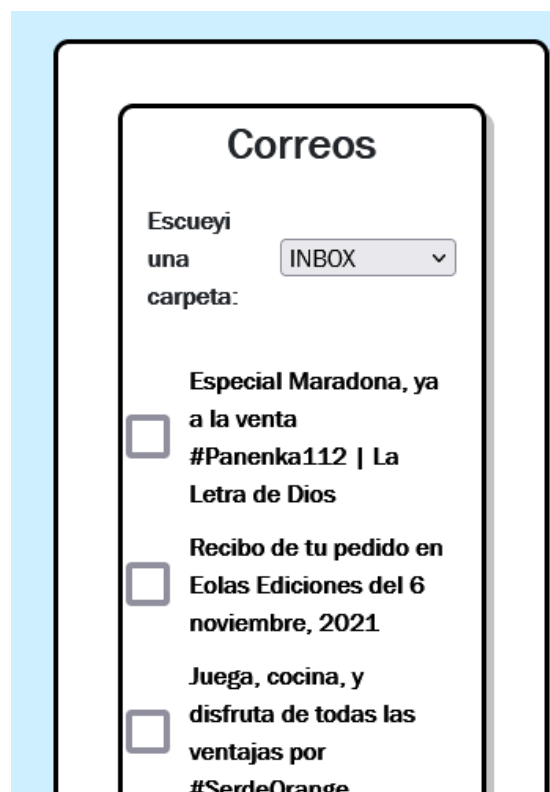


Figura 77 – Problema 2 (Solución)

Problema 3 – Error *overflow*

Algunos de los elementos de la pantalla de generación se van hacia la derecha del área en que deberían estar.

Escueyi'l formatu nel que xenerar l'acta

PDF (Portable Document Form)

Datos p'amestar al acta

Títulu

Data 17/11/2021, 19:20:55

Amestar Otru

Configuración

Incluir direiciones (Remitente, des carbón)

Amosar data de los correos esport

Amestar nome de los ficheros axun

Incluir resume participantes

Van esportase los correos que vienen dar réu:

Especial Maradona, ya a la venta #Fane


- Letra de Dios

Descargar ficheru

Descargar ficheros axuntos

Figura 78 - Problema 3 (Detección)

Para solucionarlo se reestructuran los componentes en dispositivos móviles, de la manera que se ve en las siguientes capturas.



←

Escueyi'l formatu nel que xenerar l'acta

PDF (Portable Document For ▾)

Datos p'amestar al acta

Títulu


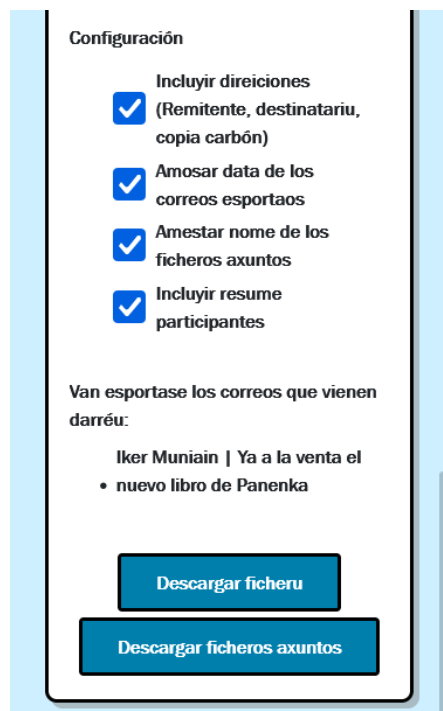


Figura 79 – Problema 3 (Solución)



Configuración

- Incluir direiciones (Remitente, destinatariu, copia carbón)
- Amosar data de los correos esportaos
- Arestar nome de los ficheros axuntos
- Incluir resume participantes

Van esportase los correos que vienen darréu:

Iker Muniain | Ya a la venta el

- nuevo libro de Panenka

Descargar ficheru

Descargar ficheros axuntos

Figura 80 – Problema 3 (Solución)

8.7 Pruebas de Usabilidad

En este punto se incluyen las pruebas realizadas con los usuarios. Se realizaron una vez finalizado el desarrollo, tras las pruebas de rendimiento y accesibilidad. Vamos a dividir este punto en dos apartados, primero los resultados de las pruebas y posteriormente su interpretación.

8.7.1 Resultados de las pruebas

A continuación, se detallan los resultados de las pruebas realizadas con tres usuarios diferentes.

8.7.1.1 Usuario 1

Nivel académico: Bachillerato.

Edad: 59.

Profesión: Repartidor a domicilio.

La siguiente tabla incluye las respuestas a las preguntas introductorias.

1-¿Con qué frecuencia usa usted un ordenador o un smartphone?
<ol style="list-style-type: none"> 1. Todos los días 2. Varias veces a la semana 3. Ocasionalmente 4. Nunca o casi nunca
2-¿Qué tipo de actividades realiza? (Pueden elegirse varias respuestas o incluso ninguna)
<ol style="list-style-type: none"> 1. Actividades relacionadas con mi trabajo o profesión 2. Aplicaciones de mensajería y/o uso de redes sociales 3. Uso otro tipo de aplicaciones móviles 4. Navegación por páginas web
3-¿Dispone de alguna cuenta de correo electrónico?
<ol style="list-style-type: none"> 1. Sí, tengo al menos una cuenta corporativa (del trabajo, universitaria...) y otra personal. 2. Sí, tengo al menos una cuenta corporativa (del trabajo, universitaria...). 3. Sí, tengo al menos una cuenta personal. 4. No dispongo de correo electrónico.
4-¿Con qué frecuencia usa el correo electrónico?
<ol style="list-style-type: none"> 1. Todos los días. 2. Lo reviso en algunas ocasiones a lo largo de la semana. 3. Lo uso cuando me lo piden para registrarme en alguna web, para tramitar algún pedido o similar.

4. Nunca o casi nunca.

5-¿Para qué usa el correo electrónico? (Pueden elegirse varias respuestas o incluso ninguna)

- Para mantener conversaciones personales.
- Para recibir notificaciones o registrarme en páginas web.
- Para mantener conversaciones en el ámbito laboral/educativo.
- Otro: _____

6- Marque las afirmaciones con las que está de acuerdo.

- A veces me cuesta encontrar determinados correos electrónicos en mi bandeja de entrada.
- Suelo descargar correos electrónicos.
- Me parece complejo enviar una conversación de correo a otra persona en mi proveedor de correo.
- Al compartir una conversación de correo electrónico, tengo miedo de enviar a alguna persona información confidencial por error.

Aquí van las respuestas a las Preguntas Cortas sobre la Aplicación y Observaciones.

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Le resulta intuitivo el proceso de generación de un acta, es decir, los diferentes pasos siguen un orden correcto?		X		
¿Los componentes de la interfaz funcionan como usted se esperaba?		X		
¿Le resulta fluida la navegación, sin grandes tiempos de espera?		X		
Calidad de la Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
El tipo y tamaño de letra es	X			
Los iconos e imágenes usados son	X			
Los colores empleados son	X			
Indique si está de acuerdo con las siguientes afirmaciones. 3 (muy de acuerdo), 0 (totalmente en desacuerdo)				
El uso de la aplicación me vendría bien en mi día a día	3	2	1	0
La aplicación necesita más opciones de personalización para las actas	3	2	1	0
La aplicación necesita más funcionalidades para satisfacer mis necesidades	3	2	1	0
La aplicación tiene muchos fallos que deben arreglarse	3	2	1	0
La aplicación es demasiado	3	2	1	0

técnica para mi nivel		XXXX		
Opinión personal				
Lo que más me ha gustado	La apariencia de la aplicación y la interacción amigable.			
Lo que menos me ha gustado	Que los correos no se pudieran cerrar tocando en una "X"			
Observaciones				

Observaciones realizadas en el Cuestionario para el Responsable de las Pruebas

Aspecto Observado	Notas
Tiempo que tarda en comprender la tarea a realizar	Bastante, en función de la tarea.
Tiempo en realizar cada tarea	Aceptable, excepto en algunas interacciones.
Necesita ayuda / pistas	Sí, especialmente en los aspectos más técnicos.
Muestra signos de fastidio o rechazo	En algunas ocasiones, por visible falta de experiencia.
Muestra signos de duda en alguna parte concreta de la aplicación	Sí, abrir y cerrar los correos le resulta muy difícil.
El usuario comete errores leves	No
El usuario comete errores graves	No

8.7.1.2 Usuario 2

Nivel académico: Carrera Universitaria.

Edad: 35

Profesión: Profesor de primaria.

La siguiente tabla incluye las respuestas a las preguntas introductorias.

1-¿Con qué frecuencia usa usted un ordenador o un smartphone?
<ol style="list-style-type: none"> 1. Todos los días 2. Varias veces a la semana 3. Ocasionalmente 4. Nunca o casi nunca
2-¿Qué tipo de actividades realiza? (Pueden elegirse varias respuestas o incluso ninguna)
<ol style="list-style-type: none"> 1. Actividades relacionadas con mi trabajo o profesión 2. Aplicaciones de mensajería y/o uso de redes sociales 3. Uso otro tipo de aplicaciones móviles 4. Navegación por páginas web
3-¿Dispone de alguna cuenta de correo electrónico?
<ol style="list-style-type: none"> 1. Sí, tengo al menos una cuenta corporativa (del trabajo, universitaria...) y otra personal.

<ol style="list-style-type: none"> Sí, tengo al menos una cuenta corporativa (del trabajo, universitaria...). Sí, tengo al menos una cuenta personal. No dispongo de correo electrónico.
4-¿Con qué frecuencia usa el correo electrónico? <ol style="list-style-type: none"> 1. Todos los días. Lo reviso en algunas ocasiones a lo largo de la semana. Lo uso cuando me lo piden para registrarme en alguna web, para tramitar algún pedido o similar. Nunca o casi nunca.
5-¿Para qué usa el correo electrónico? (Pueden elegirse varias respuestas o incluso ninguna) <ol style="list-style-type: none"> Para mantener conversaciones personales. Para recibir notificaciones o registrarme en páginas web. Para mantener conversaciones en el ámbito laboral/educativo. Otro: _____
6- Marque las afirmaciones con las que está de acuerdo. <ol style="list-style-type: none"> 1. A veces me cuesta encontrar determinados correos electrónicos en mi bandeja de entrada. Suelo descargarme correos electrónicos. Me parece complejo enviar una conversación de correo a otra persona en mi proveedor de correo. Al compartir una conversación de correo electrónico, tengo miedo de enviar a alguna persona información confidencial por error.

Aquí van las respuestas a las Preguntas Cortas sobre la Aplicación y Observaciones.

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Le resulta intuitivo el proceso de generación de un acta, es decir, los diferentes pasos siguen un orden correcto?				
¿Los componentes de la interfaz funcionan como usted se esperaba?				
¿Le resulta fluida la navegación, sin grandes tiempos de espera?				
Calidad de la Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
El tipo y tamaño de letra es				
Los iconos e imágenes usados son				
Los colores empleados son				
Indique si está de acuerdo con las siguientes afirmaciones. 3 (muy de acuerdo), 0 (totalmente en desacuerdo)				
El uso de la aplicación me vendría	3	2	1	0

<i>bien en mi día a día</i>				
<i>La aplicación necesita más opciones de personalización para las actas</i>	3	2	1	0
<i>La aplicación necesita más funcionalidades para satisfacer mis necesidades</i>	3	2	1	0
<i>La aplicación tiene muchos fallos que deben arreglarse</i>	3	2	1	0
<i>La aplicación es demasiado técnica para mi nivel</i>	3	2	1	0
Opinión personal				
<i>Lo que más me ha gustado</i>	La aplicación tiene mucho potencial para serme útil.			
<i>Lo que menos me ha gustado</i>	Los tiempos de espera son insufribles, tardan muchísimo en cargar los correos cada vez que se cambia de página.			
Observaciones				
Debéis mejorar los tiempos de carga de los correos, tardan mucho y se desespera uno. Además, no se pueden filtrar, por tanto, hay que moverse obligatoriamente por esas páginas que tanto tardan. También me gustaría que las actas tuviesen más cálculos, como las fechas de inicio y fin de los correos, no sólo los participantes. Parece que faltan parámetros para crear la conexión, por ejemplo, el puerto, no creo que sea difícil poder cambiar esto.				

Observaciones realizadas en el Cuestionario para el responsable de las Pruebas

Aspecto Observado	Notas
<i>Tiempo que tarda en comprender la tarea a realizar</i>	Muy poco
<i>Tiempo en realizar cada tarea</i>	Rápido.
<i>Necesita ayuda / pistas</i>	No.
<i>Muestra signos de fastidio o rechazo</i>	Sí, indica que los tiempos de espera son muy grandes.
<i>Muestra signos de duda en alguna parte concreta de la aplicación</i>	No.
<i>El usuario comete errores leves</i>	No.
<i>El usuario comete errores graves</i>	No.

8.7.1.3 Usuario 3

Nivel académico: Carrera Universitaria.

Edad: 42

Profesión: Secretario de Ayuntamiento.

La siguiente tabla incluye las respuestas a las preguntas introductorias.

1-¿Con qué frecuencia usa usted un ordenador o un smartphone?
1. Todos los días
2. Varias veces a la semana

<ul style="list-style-type: none"> 3. Ocasionalmente 4. Nunca o casi nunca
2-¿Qué tipo de actividades realiza? (Pueden elegirse varias respuestas o incluso ninguna)
<ul style="list-style-type: none"> 1. Actividades relacionadas con mi trabajo o profesión 2. Aplicaciones de mensajería y/o uso de redes sociales 3. Uso otro tipo de aplicaciones móviles 4. Navegación por páginas web
3-¿Dispone de alguna cuenta de correo electrónico?
<ul style="list-style-type: none"> 1. Sí, tengo al menos una cuenta corporativa (del trabajo, universitaria...) y otra personal. 2. Sí, tengo al menos una cuenta corporativa (del trabajo, universitaria...). 3. Sí, tengo al menos una cuenta personal. 4. No dispongo de correo electrónico.
4-¿Con qué frecuencia usa el correo electrónico?
<ul style="list-style-type: none"> 1. Todos los días. 2. Lo reviso en algunas ocasiones a lo largo de la semana. 3. Lo uso cuando me lo piden para registrarme en alguna web, para tramitar algún pedido o similar. 4. Nunca o casi nunca.
5-¿Para qué usa el correo electrónico? (Pueden elegirse varias respuestas o incluso ninguna)
<ul style="list-style-type: none"> 1. Para mantener conversaciones personales. 2. Para recibir notificaciones o registrarme en páginas web. 3. Para mantener conversaciones en el ámbito laboral/educativo. 4. Otro: _____
6- Marque las afirmaciones con las que está de acuerdo.
<ul style="list-style-type: none"> 1. A veces me cuesta encontrar determinados correos electrónicos en mi bandeja de entrada. 2. Suelo descargarme correos electrónicos. 3. Me parece complejo enviar una conversación de correo a otra persona en mi proveedor de correo. 4. Al compartir una conversación de correo electrónico, tengo miedo de enviar a alguna persona información confidencial por error.

Aquí van las respuestas a las Preguntas Cortas sobre la Aplicación y Observaciones.

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Le resulta intuitivo el proceso de generación de un acta, es decir, los diferentes pasos siguen un orden correcto?				
¿Los componentes de la interfaz				

<i>funcionan como usted se esperaba?</i>	3			
<i>¿Le resulta fluida la navegación, sin grandes tiempos de espera?</i>		3		
Calidad de la Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
<i>El tipo y tamaño de letra es</i>	3			
<i>Los iconos e imágenes usados son</i>	3			
<i>Los colores empleados son</i>	3			
Indique si está de acuerdo con las siguientes afirmaciones. 3 (muy de acuerdo), 0 (totalmente en desacuerdo)				
<i>El uso de la aplicación me vendría bien en mi día a día</i>	3	2	1	0
<i>La aplicación necesita más opciones de personalización para las actas</i>	3	2	1	0
<i>La aplicación necesita más funcionalidades para satisfacer mis necesidades</i>	3	2	1	0
<i>La aplicación tiene muchos fallos que deben arreglarse</i>	3	2	1	0
<i>La aplicación es demasiado técnica para mi nivel</i>	3	2	1	0
Opinión personal				
<i>Lo que más me ha gustado</i>	Es sencilla de utilizar.			
<i>Lo que menos me ha gustado</i>	Las actas no se entienden bien, parece que están hechas de prisa y corriendo.			
Observaciones				
La aplicación está muy guapa, pero las actas son un poco feas en comparación. Además, no puedo ver que correo responde a cuál, debo marcarlos todos.				

Observaciones realizadas en el Cuestionario para el responsable de las Pruebas

Aspecto Observado	Notas
<i>Tiempo que tarda en comprender la tarea a realizar</i>	Muy poco.
<i>Tiempo en realizar cada tarea</i>	Rápido.
<i>Necesita ayuda / pistas</i>	No.
<i>Muestra signos de fastidio o rechazo</i>	No.
<i>Muestra signos de duda en alguna parte concreta de la aplicación</i>	No.
<i>El usuario comete errores leves</i>	No.
<i>El usuario comete errores graves</i>	No.

8.7.2 Interpretación de los resultados

Las pruebas de usabilidad con los usuarios revelan sensaciones positivas y negativas. La aplicación se valora principalmente por su utilidad, potencial, sencillez y apariencia. Parece que se ha logrado el objetivo marcado de realizar una aplicación amigable, fácil de usar. Sin embargo, ciertos aspectos no convencen del todo a los usuarios y se muestran como un gran inconveniente, que puede desembocar en el rechazo total del producto por parte del público final.

Los aspectos negativos más destacados son los tiempos de espera, las soluciones planteadas para subsanarlos durante la fase de desarrollo no parecen convencer a los usuarios y siguen causando problemas que no parecen tener fácil solución. El punto en el que esto se hace patente es la navegación entre páginas, a todas luces inconveniente y que además es obligatoria por carecer de algún tipo de buscador/filtro.

Otro tema que destacan los usuarios son las actas finales, cuya organización resulta confusa y su apariencia poco agradable. También se recomienda aprovechar el potencial que tienen y mostrar algún campo calculado en el documento final.

8.8 Modificaciones tras las pruebas

Las pruebas realizadas fueron muy satisfactorias, especialmente aquellas tras el tercer *sprint*, pues detectaron muchos puntos a mejorar. A la vista de los resultados, se decidió alargar el desarrollo unas semanas y trabajar aquellos puntos donde se veía factible una mejora sustancial. En los próximos apartados especificaremos cuáles fueron estas mejoras, los motivos para su realización y la repercusión que tuvieron en diferentes partes de la interfaz y del diseño.

8.8.1 Descripción de los cambios

Siguiendo el mismo formato que a lo largo del desarrollo describiremos en forma de Historias de Usuario las evoluciones realizadas.

<i>Como usuario quiero que el tiempo que se tarda en cargar los correos de otra página sea mucho menor para tener una experiencia más fluida.</i>	
Título Corto	Reducir tiempo páginas
Identificador	H27
EPIC	LISTADO
Tiempo estimado (horas)	15
Tiempo necesario para completar (horas)	12

Criterios de aceptación

- El tiempo de carga entre páginas del listado será instantáneo.
- El tiempo de carga la primera vez que se conecte el usuario podrá ser más alto.
- El resto de veces que el usuario se conecte el tiempo de carga será menor.
- El usuario podrá ver, tras la primera carga, al menos los 100 correos más recientes de la carpeta seleccionada.
- El usuario podrá cargar correos más antiguos en la carpeta seleccionada en grupos de diez correos.
- El usuario podrá recargar los correos de una carpeta.

Justificación

Pese a lo argumentado en el punto 2.3.2.5 de la Evaluación de Alternativas donde se explican las razones para prescindir de una base de datos, las pruebas revelan que su uso puede ser beneficioso para la experiencia de usuario. En ese punto de la documentación, comentábamos que no es necesario almacenar los correos para el uso de la aplicación, sin embargo, a la vista de las pruebas, su uso puede ser una buena solución si almacenamos algunos datos para agilizar la experiencia de usuario. Con esta historia, modificamos ligeramente la arquitectura de la aplicación, añadiendo una capa extra que reduce las llamadas al proveedor. Esta base de datos no será nada complejo, ni almacenará grandes cantidades de datos que nos traigan los riesgos de seguridad que comentábamos en el punto citado. Tampoco habrá un sistema propio de autenticación y simplemente nos permitirá mantener referencias entre algunos correos del usuario y el proveedor, reduciendo ostensiblemente los tiempos de carga.

Como usuario quiero poder filtrar los correos que me aparecen para evitar perder tiempo navegando entre todas las páginas.

Título Corto	Filtro de correos
Identificador	H28
EPIC	LISTADO
Tiempo estimado (horas)	6
Tiempo necesario para completar (horas)	4,5

Criterios de aceptación

- El usuario podrá especificar un texto en un buscador.
- El listado de correos mostrará sólo aquellos correos en la carpeta que contengan el texto del buscador en el asunto.

Justificación

La introducción de la base de datos en la historia anterior (H27), nos permite añadir sin

mucho esfuerzo esta funcionalidad, que aporta mucha más agilidad a la hora de navegar por la lista de correos, uno de los problemas que se destacaban en las pruebas con los usuarios.

Como usuario quiero poder seleccionar más datos a la hora de crear la conexión para dar cabida a más proveedores.

Título Corto	Conexiones avanzadas
Identificador	H29
EPIC	CONEXIÓN
Tiempo estimado (horas)	3
Tiempo necesario para completar (horas)	3
Criterios de aceptación	
<ul style="list-style-type: none"> El usuario podrá personalizar el puerto, el protocolo y elegir el uso de SSL en las conexiones que cree. 	
Justificación	
<p>Un problema que se destacó y que tenía muy fácil solución fue este. Antes sólo se podía elegir el host al que conectarse, ahora ya pueden personalizarse todos los parámetros que definen la conexión.</p>	

Como usuario quiero unas actas más cuidadas para que sea más fácil su implantación en un entorno profesional.

Título Corto	Apariencia actas
Identificador	H30
EPIC	GENERACIÓN
Tiempo estimado (horas)	6
Tiempo necesario para completar (horas)	3
Criterios de aceptación	
<ul style="list-style-type: none"> Los títulos de las actas serán consistentes, manteniendo todos el tipo de letra y el color y variando el tamaño en función de su importancia. Los campos importantes se destacarán usando colores y/o textos en negrita. Se solucionarán los problemas de codificación evitando la aparición de caracteres extraños. 	

- La fuente será la misma en todo el documento.

Justificación

Las actas generadas anteriormente no gozaban de mucha aceptación, resultando poco cuidadas e inconsistentes gráficamente, con esta historia se mejora ampliamente su apariencia.

Como usuario quiero poder ver un resumen de las fechas de inicio y fin de las conversaciones incluidas en un acta para tener un resumen de su contenido.

Título Corto	Fechas de inicio/fin
Identificador	H31
EPIC	GENERACIÓN
Tiempo estimado (horas)	3
Tiempo necesario para completar (horas)	3

Criterios de aceptación

- Las actas contendrán un nuevo campo obligatorio al inicio.
- El campo contendrá la fecha del correo más antiguo de entre todos los del acta y la del más reciente.
- La fecha usará el formato largo.

Justificación

Se destacó la potencialidad que tenía la aplicación para realizar más cálculos con los correos del acta y se añadió fácilmente este.

Como usuario quiero evitar tener que seleccionar todos los correos de una conversación manualmente si quiero añadirlos al acta para evitar pérdidas de tiempo innecesarias y posibles errores.

Título Corto	Incluir correos anteriores
Identificador	H32
EPIC	GENERACIÓN
Tiempo estimado (horas)	6
Tiempo necesario para completar (horas)	6

Criterios de aceptación
<ul style="list-style-type: none"> • Cuando se seleccione un correo para añadir al acta, se añadirán automáticamente todos los que formen parte del hilo de correos. • Se valorarán sólo aquellos que estén cargados en la base de datos. • No se incluirá el mismo correo dos veces. • Los correos del acta se mostrarán todos ordenados de más reciente a más antiguo.
Justificación
Anteriormente para incluir una conversación entera debían seleccionarse todos sus correos, esta modificación automatiza esa selección, facilitando mucho el uso.

<i>Como usuario no quiero ver en el contenido de los correos parte del contenido de otros para evitar confusiones.</i>	
Título Corto	Eliminar contenido de otros correos
Identificador	H33
EPIC	GENERACIÓN
Tiempo estimado (horas)	6
Tiempo necesario para completar (horas)	9
Criterios de aceptación	
<ul style="list-style-type: none"> • El cuerpo de los correos no contendrá partes que sean de otro correo anterior al que este responda. • En el listado de correos no aparecerán esas partes. • En el acta tampoco aparecerán. 	
Justificación	
El acta resultaba algo confusa en conversaciones que encadenaban varios correos, pues normalmente los clientes de correo añaden en el cuerpo de un correo, el cuerpo de todos los anteriores. Con esta modificación se elimina el cuerpo de los anteriores.	

8.8.1.1 Requisitos

Todas estas historias traen consigo cambios sobre los requisitos funcionales de la aplicación. Se incluye a continuación la tabla, con el formato habitual usado en los puntos anteriores.

Código	Historia	Requisito	Modificaciones
Con1	H1	El sistema deberá abrir una conexión con el servidor de correo del usuario	

Con1.1	H1	El usuario deberá especificar sus credenciales para la conexión	
Con1.1.1	H1	El usuario deberá especificar su correo electrónico	
Con1.1.2	H1	El usuario deberá especificar la contraseña de su correo electrónico	
Con1.1.3	H15	El usuario podrá especificar el host de su proveedor de correo electrónico	
Con1.1.4	H29	El usuario podrá especificar el puerto de su servidor de correo electrónico	Se añaden los nuevos datos personalizables para la conexión
Con1.1.5	H29	El usuario podrá especificar el protocolo que se use para la conexión	
Con1.1.6	H29	El usuario podrá especificar si desea usar SSL/TLS o no	
Con1.2	H1	Se abrirá una conexión con el servidor adecuado	
Con1.2.1	H15	Si se especifica un host en Con1.1.3 se intentará abrir la conexión con el host especificado	
Con1.2.2	H15	Si no se especifica un host en Con1.1.3 se intentará abrir la conexión con el proveedor adecuado de la lista de proveedores para el dominio de la dirección de correo especificada en Con.1.1.1.	
Con1.2.2.1	H15	La lista de proveedores contendrá la equivalencia entre el dominio de correo y el host.	
Con1.2.2.2	H15	La lista de proveedores podrá ser modificada por el administrador del sistema.	
Con1.3	H1	Si la conexión abierta en Con1.2 es exitosa se informará al usuario	
Con1.3.1	H1	Si se producen errores al intentar conectarse a Con1.2 se informará al usuario	
Con2	H2	El sistema deberá permitir al usuario desconectarse del servidor de correo	
Con2.1	H2	El sistema deberá cerrar todas las conexiones abiertas que tenga con el servidor de correo del proveedor del usuario	
Con2.2	H2	Se deberá notificar al usuario si el cierre de la conexión fue satisfactorio o no	
Con3	H18	Se indicará como conseguir una contraseña de aplicación para gmail.	
List1	H3	El sistema deberá listar los correos recibidos en la cuenta de correo cuya conexión fue abierta en Con1	
List1.1	H6	Para cada correo se deberá mostrar una serie de datos	
List1.1.1	H6	Se mostrará el asunto de los correos	
List1.1.1.1	H3	Si el correo no tuviera asunto se mostrará el texto "<sin asunto>"	
List1.1.2	H6	Se mostrará la fecha y hora de envío de los correos	
List1.1.2.1	H6	La fecha y hora tendrán precisión de minutos	
List1.1.3	H6	Se mostrará el remitente de los correos	

List1.1.3.1	H6	Se indicará nombre y dirección del remitente	
List1.1.4	H6	Se mostrará el destinatario o destinatarios de los correos	
List1.1.4.1	H6	Se indicará nombre y dirección de los destinatarios	
List1.1.5	H6	Se mostrará el cuerpo de los correos	
List1.1.5.1	H23	Se incluirá el cuerpo de los correos como HTML	
List1.1.5.2	H33	Se eliminará la parte del HTML que ya aparezca en el cuerpo de otros correos de la misma conversación.	Se eliminan las citas a otros correos.
List1.1.6	H20	Se mostrará una lista con el nombre de los archivos adjuntos de cada correo	
List1.1.7	H24	Se mostrará la copia de carbón de los correos	
List1.1.7.1	H24	Se indicará el nombre y la dirección de los destinatarios de la copia de carbón	
List1.2	H3	Si se produjesen errores obteniendo los correos se notificará el error	
List1.3	H12	Los correos se mostrarán en orden descendiente de fecha y hora (más recientes primero)	
List1.4	H16	Se mostrarán los correos de todas las carpetas	
List1.4.1	H16	No se mostrarán aquellas carpetas que no contengan mensajes	
List1.4.2	H27	La primera vez que el usuario acceda a la aplicación se mostrarán 100 correos por carpeta	Se añade la nueva funcionalidad de cargar poco a poco los correos y de recargar una carpeta.
List1.4.2.1	H27	Si hay menos de 100 correos en una carpeta se mostrarán todos los que haya	
List1.4.3	H27	El usuario podrá cargar manualmente los correos de diez en diez	
List1.4.4	H27	El usuario podrá recargar todos los correos de una carpeta	
List1.5	H28	El sistema permitirá al usuario filtrar correos	Nueva funcionalidad del filtro de correos.
List1.5.1	H28	El usuario deberá especificar un texto que usar como filtro	
List1.5.2	H28	Si se aplica el filtro el sistema deberá mostrar sólo los correos que contengan en el asunto el texto especificado en List1.5.1 para la carpeta seleccionada.	
Gen1	H5	El sistema deberá permitir generar un acta con una serie de correos seleccionados	
Gen1.1	H5	El sistema deberá permitir seleccionar un número ilimitado de correos entre los listados en List1	
Gen1.1.2	H32	Además de los correos seleccionados se incluirán todos aquellos correos previos a estos	Se incluye la detección automática de los correos de la misma conversación.
Gen1.1.2.1	H32	Se considerará un correo previo aquel que tenga su "Message-ID" en la cabecera "In-Reply-To" de alguno de los seleccionados.	
Gen1.2	H5	El sistema deberá generar un documento PDF que será el acta	
Gen1.2.1	H5	El documento PDF generado deberá poder ser	

		descargado por el usuario a su sistema local	
Gen1.2.2	H5	El acta contendrá los correos seleccionados en Gen1.1	
Gen1.2.2.1	H5	Para cada correo se incluirá en el acta el asunto del mismo	
Gen1.2.2.2	H5	Para cada correo se incluirá en el acta el cuerpo del mismo	
Gen1.2.2.2.1	H23	Se mostrará el cuerpo en formato HTML	
Gen1.2.2.2.2	H23	Si se produjesen problemas incluyendo el HTML en el archivo, se indicará que se puede consultar en los archivos adjuntos.	
Gen1.2.2.2.3	H33	Se eliminará la parte del HTML que ya aparezca en el cuerpo de otros correos de la misma conversación.	Se eliminan las citas a otros correos también en el acta.
Gen1.2.2.3	H5	Para cada correo se incluirá en el acta la fecha de envío del mismo	
Gen1.2.2.3.1	H5	El usuario deberá poder elegir si las fechas de envío se incluyen o no	
Gen1.2.2.4	H7	El acta podrá contener una serie de campos personalizados	
Gen1.2.2.4.1	H7	Los campos personalizados estarán formados por un par compuesto por una etiqueta y un valor	
Gen1.2.2.4.2	H7	Una etiqueta será un texto, que puede estar vacío	
Gen1.2.2.4.3	H7	Un valor será un texto, que puede estar vacío	
Gen1.2.2.4.4	H7	El usuario podrá añadir tantos campos personalizados como desee	
Gen1.2.2.4.5	H7	Los campos personalizados se mostrarán en el acta generado y serán la etiqueta y el valor.	
Gen1.2.2.5	H20	El acta contendrá el nombre de los archivos adjuntos de cada correo	
Gen1.2.2.6	H25	El acta contendrá una tabla que resuma los participantes de los correos del acta.	
Gen1.2.2.6.1	H21	Será un participante cada email que salga en el remitente, el destinatario o la copia de carbón de alguno de los correos del acta.	
Gen1.2.2.6.2	H21	Para cada participante se mostrará el email	
Gen1.2.2.6.3	H21	Para cada participante se mostrarán todos los nombres que aparezcan junto al email.	
Gen1.2.2.6.4	H25	Para cada participante se mostrará el número de mensajes enviados	
Gen1.2.2.6.5	H25	Para cada participante se mostrará el número de mensajes recibidos	
Gen1.2.2.6.6	H25	Para cada participante se mostrará el número de mensajes en los que es destinatario de copia de carbón	
Gen1.2.2.7	H24	El acta contendrá los destinatarios de la copia de carbón	
Gen1.2.2.7.1	H24	El usuario podrá indicar si los destinatarios de la	

		copia de carbón se incluyen o no	
Gen1.2.2.8	H31	El acta contendrá un campo que especifique las fechas de inicio y fin de la conversación	Se añade el campo calculado con las fechas de inicio y fin
Gen1.2.2.8.1	H31	La fecha de inicio será la fecha del correo más antiguo incluido en el acta	
Gen1.2.2.8.2	H31	La fecha de fin será la fecha del correo más reciente incluido en el acta	
Gen1.2.2.8.3	H31	Las fechas usarán el formato largo "E dd MM YYYY HHmm"	
Gen1.3	H14	El usuario podrá generar un archivo CSV	
Gen1.3.1	H14	El archivo CSV generado deberá poder ser descargado por el usuario a su sistema local.	
Gen1.3.2	H14	El archivo contendrá los correos seleccionados en Gen1.1	
Gen1.3.3	H14	Para cada correo se incluirán los siguientes datos	
Gen1.3.3.1	H14	El asunto del correo	
Gen1.3.3.2	H14	La fecha de envío del correo	
Gen1.3.3.2.1	H14	El usuario deberá poder elegir si las fechas de envío se incluyen o no	
Gen1.3.3.3	H14	Los archivos adjuntos del correo	
Gen1.3.3.3.1	H14	El usuario deberá poder elegir si los archivos adjuntos del correo se incluyen o no	
Gen1.3.3.4	H14	Las direcciones del remitente, destinatario y copia de carbón	
Gen1.3.3.4.1	H14	El usuario deberá poder elegir si las direcciones se incluyen o no	
Gen1.3.3.5	H14	El cuerpo del correo	
Gen1.4	H20	El usuario podrá descargar una serie de archivos adjuntos	
Gen1.4.1	H20	Se generará un archivo .zip que podrá ser descargado por el usuario a su sistema local.	
Gen1.4.1.1	H20	Los archivos adjuntos de los correos se incluirán en el ZIP generado	
Gen1.4.1.2	H20	Cada uno de los cuerpos de los correos se incluirá en el ZIP generado como un archivo diferente.	

Tabla 19 – Lista de requisitos funcionales tras las pruebas

8.8.2 Análisis

Los cambios realizados en el punto anterior necesitarán modificar algunas partes de la aplicación. Una fase de análisis obliga a añadir ciertas clases al *back-end* que se describen en el diagrama de la Figura 81. Aparecen destacadas en verde y se describen detalladamente debajo.

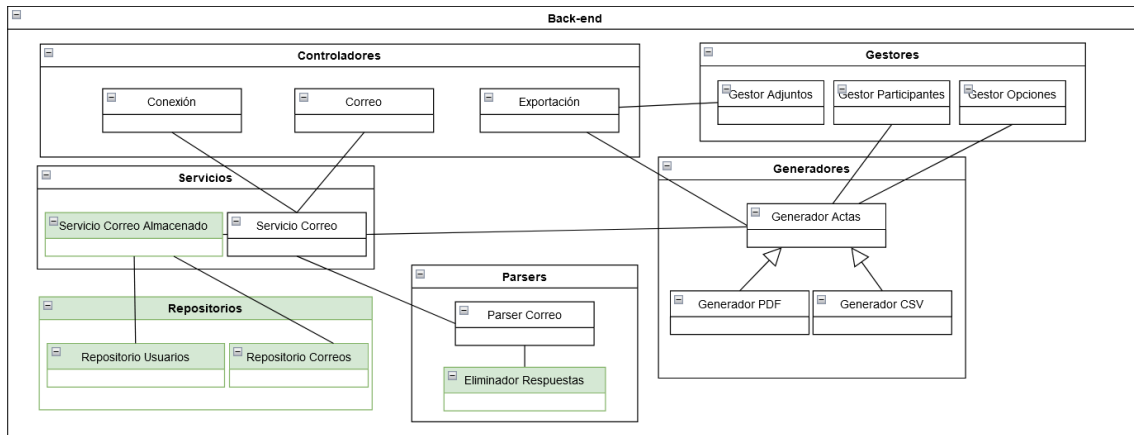


Figura 81 – Diagrama de clases del back-end modificado.

Nombre de la Clase
Servicio Correo Almacenado
Descripción
Servicio que se encarga de las operaciones con los correos almacenados.
Responsabilidades
<ul style="list-style-type: none"> - Responder a las peticiones del servicio de correos. - Tratar los correos que se obtienen de la base de datos. - Gestionar las operaciones de reconstrucción, carga y eliminación de correos y carpetas. - Gestionar las operaciones de lectura de correos y carpetas.
Atributos Propuestos
n/a
Métodos Propuestos
<p>reconstruirBaseDeDatos: Carga los nuevos correos en las carpetas existentes en base de datos y crea de cero las que no estén.</p> <p>cargarMasEnCarpeta: Carga más correos antiguos en la carpeta dada.</p> <p>reconstruirCarpeta: Elimina la carpeta dada y vuelve a crearla con el número inicial de correos.</p>

Nombre de la Clase
Repositorio Usuarios
Descripción
Realiza las consultas a la base de datos correspondientes a los usuarios
Responsabilidades
<ul style="list-style-type: none"> - Realizar consultas CRUD a la tabla Usuario de la base de datos.
Atributos Propuestos
n/a
Métodos Propuestos
<p>seleccionarUsuario: Dados los datos de un usuario autenticado obtiene su identificador en base de datos.</p>

Nombre de la Clase
Repositorio Correos
Descripción
Realiza las consultas a la base de datos correspondientes a los correos y las carpetas.

Responsabilidades
- Realizar consultas CRUD a la tabla Correo de la base de datos.
Atributos Propuestos
n/a
Métodos Propuestos
obtenerCarpetasDeUsuario: Devuelve el nombre de las carpetas de un usuario dado. obtenerCorreosDeCarpeta: Obtiene todos los correos de una carpeta. obtenerCorreosPorID: Obtiene los correos dado su identificador en la carpeta y la carpeta a la que pertenece. obtenerCorreosPorMessageID: Obtiene los correos buscando el valor de su cabecera "Message-ID".

Nombre de la Clase
Eliminador Respuestas
Descripción
A la hora de <i>parsear</i> los correos facilita la eliminación de las partes pertenecientes a otros.
Responsabilidades
- Elimina las partes no deseadas del cuerpo de los correos en función del proveedor.
Atributos Propuestos
n/a
Métodos Propuestos
eliminarRespuestas: Elimina la parte no deseada del HTML del cuerpo. Tendrá una implementación distinta por cada proveedor soportado.

8.8.3 Diseño

Tras la fase de análisis, pasamos a una fase de diseño, que también trae consigo algunos cambios en distintas partes del sistema y que reflejaremos con modificaciones sobre algunos de los diagramas de los puntos previos.

8.8.3.1 Diagrama E-R de la base de datos

Anteriormente no contábamos con este punto, pues la aplicación no tenía base de datos. Tras los cambios realizados se ha añadido una, que, si bien es muy simple y de pequeño tamaño, debe quedar reflejada por medio de un diagrama entidad-relación que muestre sus tablas, campos y claves.

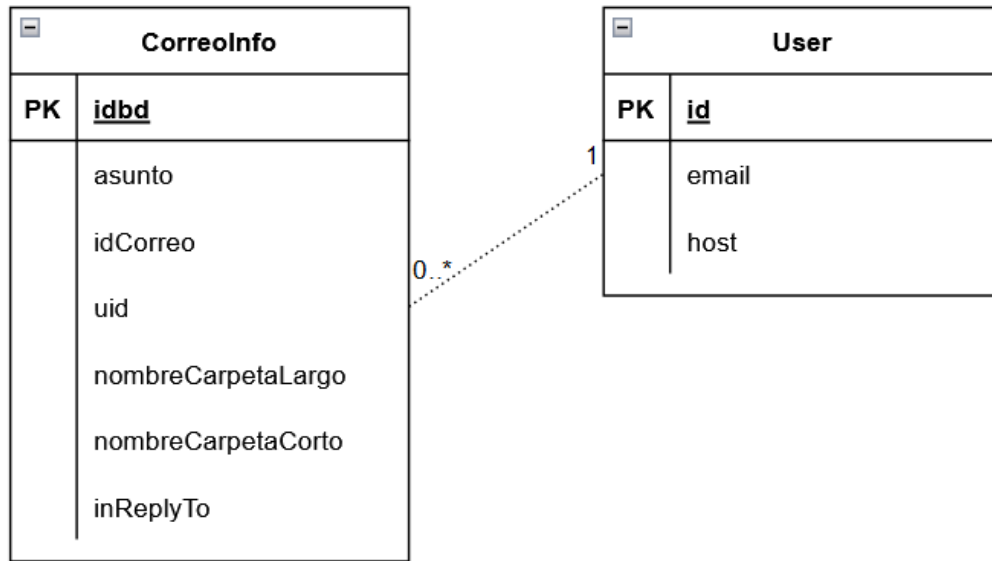


Figura 82 – Diagrama E-R de la Base de Datos

8.8.3.2 Diagrama de paquetes

El diagrama de paquetes también se ve modificado, añadiendo los repositorios para el acceso a la base de datos y las nuevas clases para la limpieza del HTML en el cuerpo de los correos.

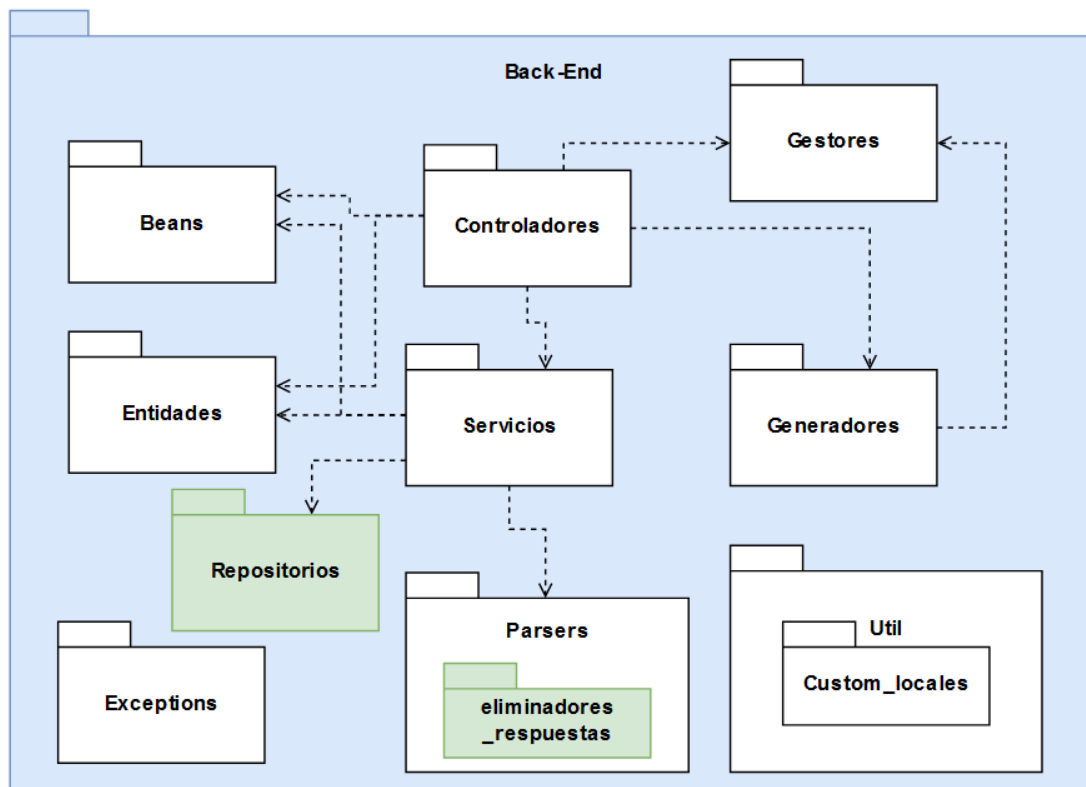


Figura 83 – Diagrama de paquetes modificado

8.8.3.3 Diagrama de componentes

El diagrama de componentes también se modifica ligeramente para añadir las nuevas funcionalidades. De la misma manera que en el anterior, en la Figura 84 se destacan en verde los nuevos componentes. Como puede verse, el Servicio Correo conectará con el Servicio Correo Almacenado en aquellos casos en los que no necesite conectarse al proveedor. Por otro lado, este servicio se conectará también al proveedor para obtener los correos en la reconstrucción o creación de carpetas y correos. Además, los repositorios servirán como Middleware para la conexión con la base de datos. Por último, el Eliminator Respuestas, será el componente que use el Parser Correo. Para el desarrollo de esta funcionalidad se usará el patrón Estrategia (*Strategy*), donde habrá una implementación concreta de la interfaz EliminatorRespuesta para cada uno de los proveedores de correo de referencia. Si se desearan añadir en un futuro nuevos proveedores simplemente habría que realizar más implementaciones de esta interfaz.

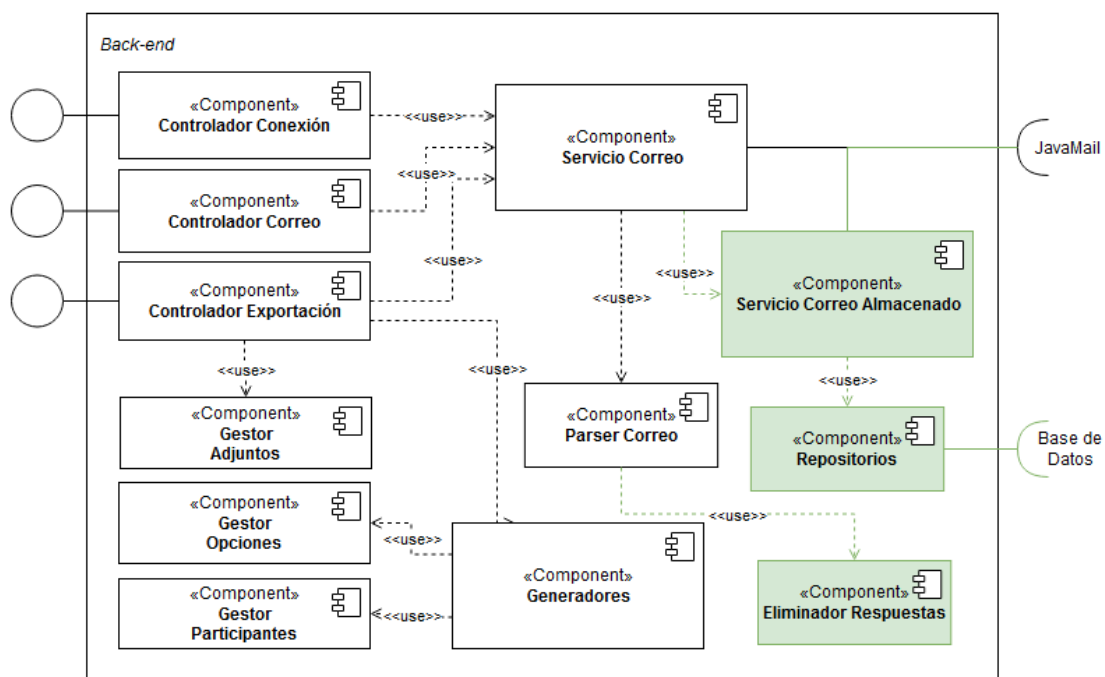


Figura 84 – Diagrama de componentes modificado

8.8.3.4 Diagrama despliegue

Al añadir la base de datos, debemos actualizar el diagrama de despliegue, que añade un contenedor extra.

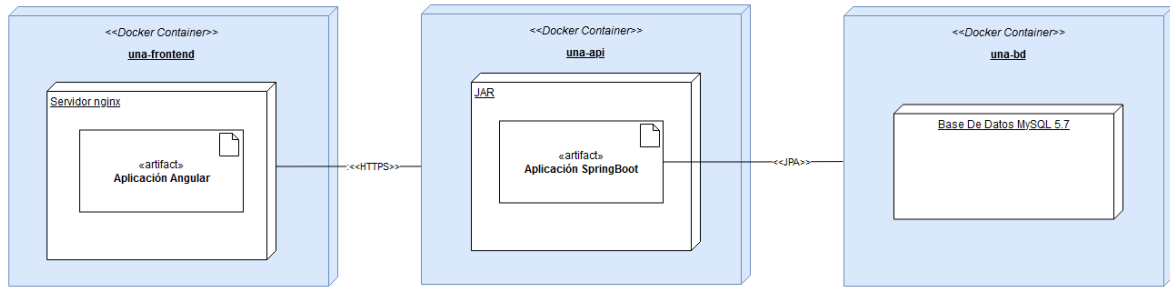


Figura 85 – Diagrama de despliegue modificado

8.8.3.5 Diagramas de interacción

Uno de los puntos clave de estas mejoras es que tocan diferentes aspectos del flujo anterior de la aplicación. Para dejar claro el nuevo funcionamiento, actualizamos algunos de los diagramas contenidos en el punto 6.3 y añadimos uno nuevo. Vamos a analizar uno a uno los diagramas que incluimos en las próximas páginas.

La Figura 86 describe el nuevo proceso que se incluye con estas mejoras. Es el de sincronización de la Base de Datos con el proveedor. Se inicia al abrir una conexión, justo antes del proceso de obtener las carpetas. Como vemos sigue el orden de capas habitual hasta llegar al `MailService`, una vez ahí realiza una llamada al `StoredMailService`, que es quien se encarga de coordinar el `UserRepository` (que realiza el acceso a Base de Datos) y el proveedor de correo (que obtiene los nuevos correos añadidos). En este punto se han omitido varios métodos de utilidades para no ensuciar el diagrama, pero, en esencia, se trata de iterar sobre las carpetas añadiendo los nuevos correos. Este proceso tiene ciertas variantes en función del estado del usuario y la carpeta, por ejemplo, si no estuviese la carpeta o el usuario en BD, los crearía. Seguirían el mismo esquema que este diagrama más funcionalidades que se añaden con los cambios de este punto. La opción de “Reconstruir carpeta” haría lo mismo, pero sólo para la carpeta que se pase como parámetro. También la de “Cargar más correos antiguos” sería el mismo proceso, pero en lugar de obtener los correos más recientes, los más antiguos.

El siguiente esquema es la Figura 87, que muestra un ejemplo de cómo se realizan los nuevos procesos de obtención de datos (obtener carpetas y correos). Este en concreto, particulariza el caso de las carpetas, pero también es aplicable a los correos. La diferencia con el esquema que teníamos al final del desarrollo (Figura 47), es bien sencilla. A la hora de acceder a los datos, se accede desde los repositorios en lugar de directamente al proveedor.

La Figura 88 cierra este apartado, indicando dónde se realizaría la obtención de los correos relacionados a la hora de exportar. Es en el `MailService`, donde por medio de manera recursiva se van obteniendo aquellos que tengan relación, antes de generar el acta.

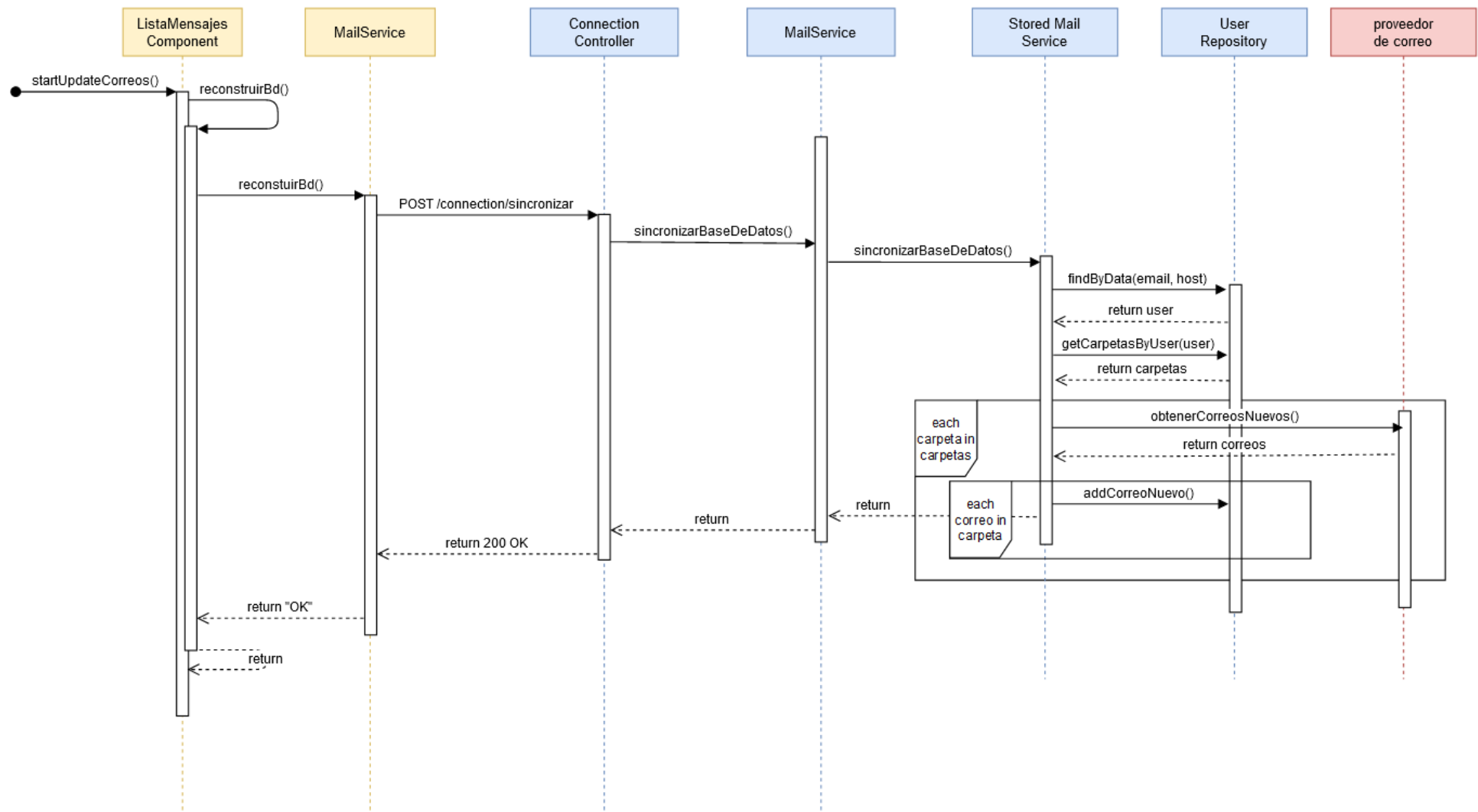


Figura 86 – Diagrama de interacción (sincronizar base de datos)

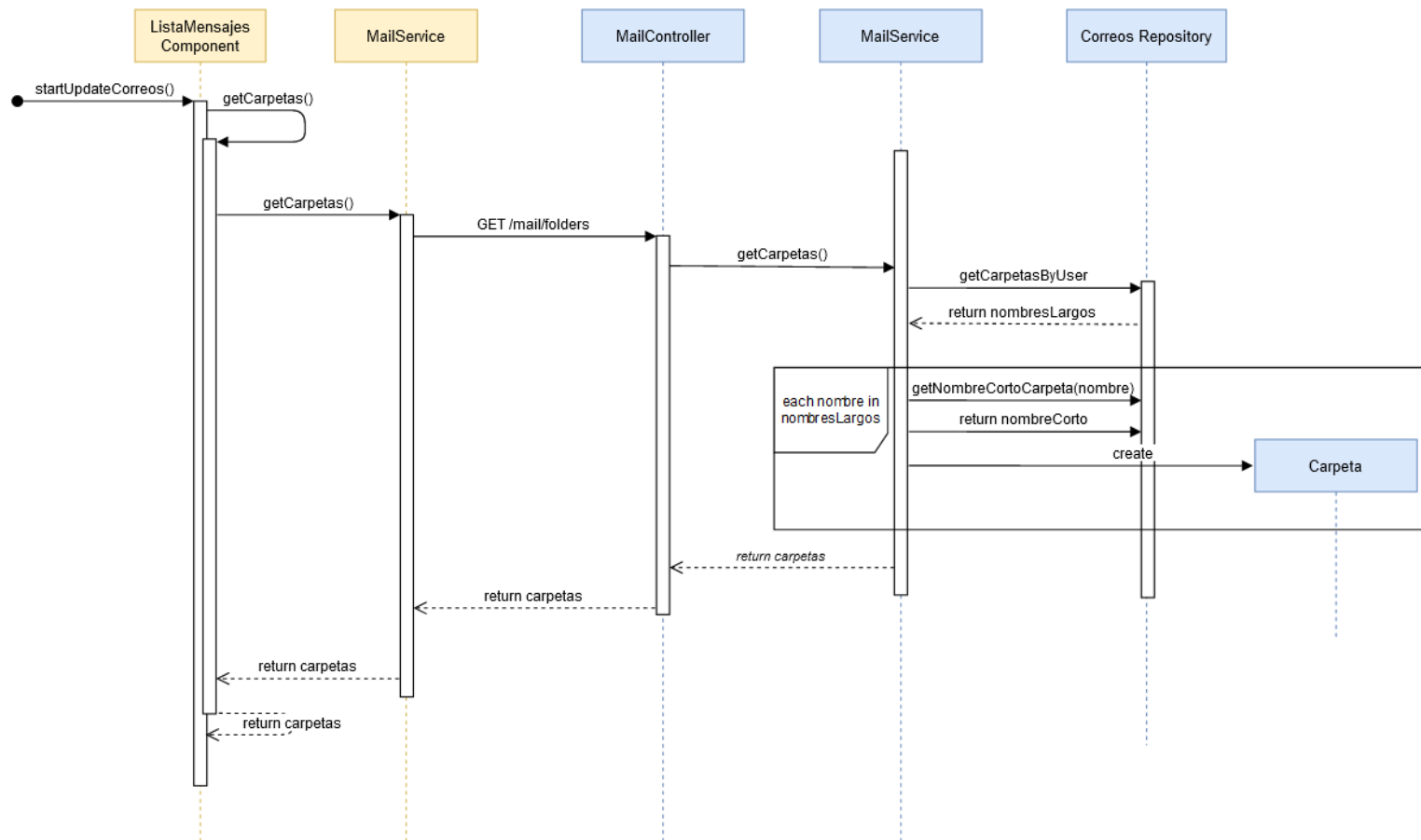


Figura 87 – Diagrama de interacción modificado (obtener carpetas)

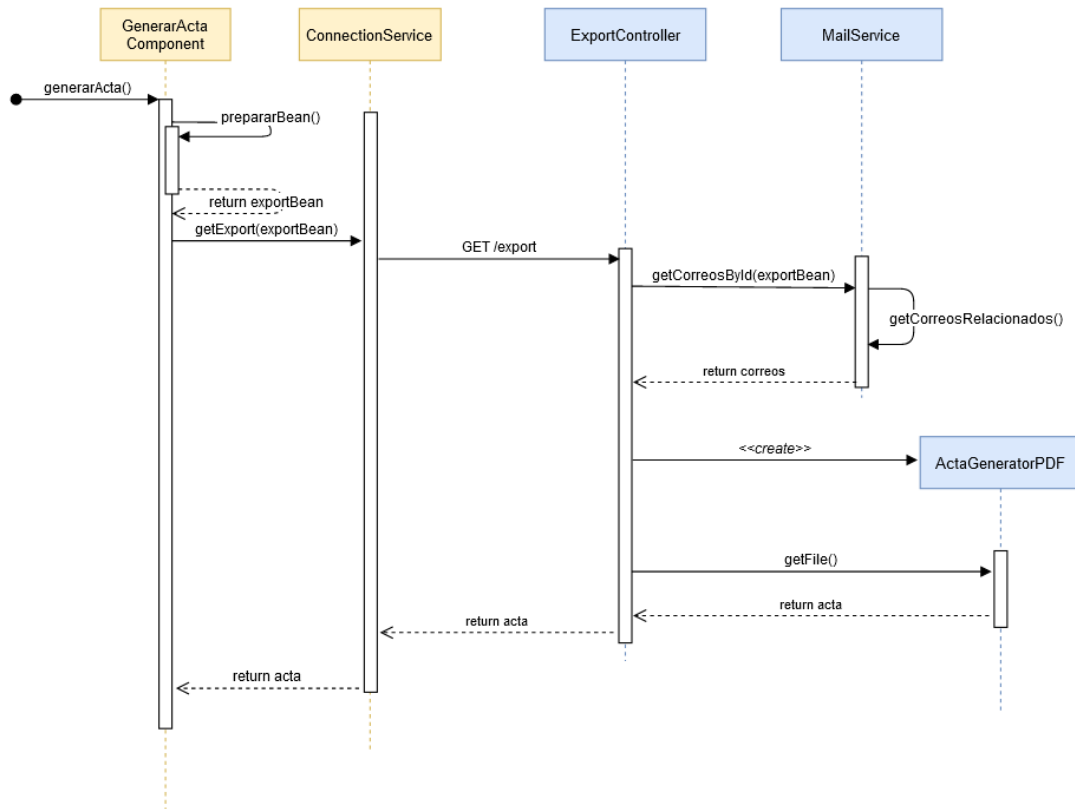


Figura 88 – Diagrama de interacción modificado (exportación)

8.8.4 Cambios sobre la interfaz

La interfaz incluye algunos cambios que es importante reseñar, para tener claro lo que se ha añadido de cara al usuario. La Figura 89 muestra la nueva apariencia. Se puede ver en la parte superior el buscador, donde se introducirá el texto y realizará el filtrado al clicar sobre “Buscar”. Abajo, se ha añadido un contador con los correos cargados en la base de datos y dos botones, uno para cargar más correos antiguos y otro para reconstruir la carpeta. El progreso de estas operaciones se indicará por medio de indicadores dinámicos de carga.

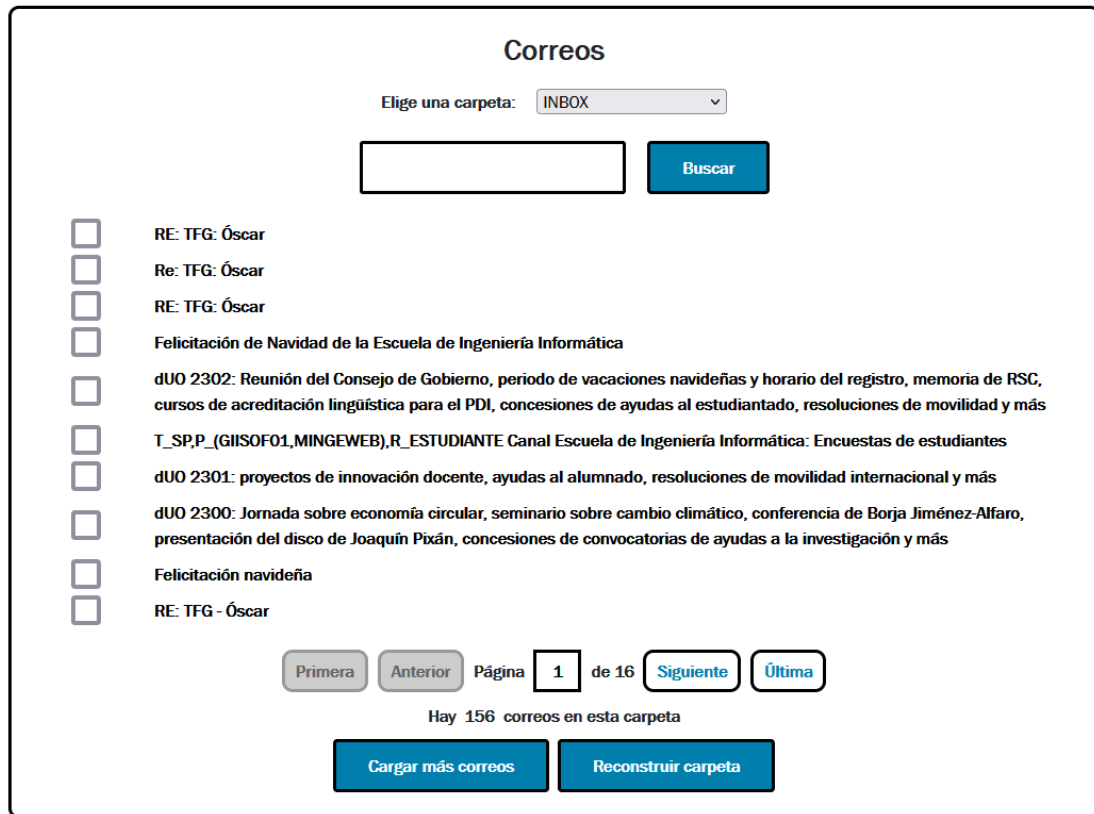


Figura 89 – Interfaz del componente de listado de correos.

Dentro de la parte de interfaz consideramos también las actas, que tienen una apariencia mejorada, como puede verse en la Figura 90.

Acta

Título: Acta de ejemplo

Fecha del acta: 31 de diciembre de 2021

Fechas de inicio y fin de los correos: jueves 23 de diciembre de 2021 09H15' CET al jueves 23 de diciembre de 2021 09H15' CET

Participantes

Participante	Remitente	Destinatario	CC
OFICINA DE COMUNICACION DE LA UNIVERSIDAD DE OVIEDO <comunica@uniovi.es>	1	1	0

Correos

dUO 2302: Reunión del Consejo de Gobierno, periodo de vacaciones navideñas y horario del registro, memoria de RSC, cursos de acreditación lingüística para el PDI, concesiones de ayudas al estudiantado, resoluciones de movilidad y más

- **Remitente:** OFICINA DE COMUNICACION DE LA UNIVERSIDAD DE OVIEDO <comunica@uniovi.es>

- **Destinatario:** OFICINA DE COMUNICACION DE LA UNIVERSIDAD DE OVIEDO <comunica@uniovi.es>

- **Fecha de envío:** jueves 23 de diciembre de 2021 09H15' CET

Diario de la Universidad de Oviedo (dUO) – Titulares – 23/12/2021

—

- [Reunión extraordinaria del Consejo de Gobierno de la Universidad de Oviedo](#)
- [Exposición "Érase una vez Sairutsa", en la Escuela de Ingeniería de Minas, Energía y Materiales](#)

—

- [Periodo de vacaciones navideñas y horario del registro](#)

Figura 90 – Nuevo acta mejorado.

8.8.5 Desarrollo de los cambios

Por último, se implementaron los cambios y si bien la aplicación proporciona una experiencia de uso considerablemente mejor, conseguir el correcto funcionamiento de algunas funcionalidades presentó cierta dificultad. Algunos aspectos no se lograron del todo y otros presentaron inconvenientes que hubo que resolver. Para cerrar este capítulo, comentamos esos problemas encontrados.

8.8.5.1 Problemas de sincronización

La inclusión de la base de datos facilita varias operaciones, como la obtención de carpetas y correos y su procesamiento rápido, sin embargo, el mantener una copia de los correos presenta la dificultad de que puede producirse una inconsistencia entre el proveedor y nuestra base de datos. Por un lado, proporcionamos esa sincronización automática, añadiendo nuevos correos al arrancar la aplicación. Sin embargo, aún puede darse el caso de que se borren

correos en el proveedor, lo cual implicaría que la base de datos contiene referencias a correos inexistentes. Para solucionarlo, usamos una estrategia comúnmente conocida como *lazy*, esto es, en lugar de examinar permanentemente todos los correos buscando aquellos que haya que eliminar, cosa que implicaría mucha carga computacional, decidimos que cuando se llame a un correo y no encontremos referencia en el proveedor sea cuando se elimine de nuestra base de datos. Esta solución presenta buenas prestaciones en la mayoría de los casos, aunque sigue dándose un pequeño problema. La referencia puede no encontrarse por un eventual fallo en la conexión (aunque verdaderamente no se haya eliminado del proveedor). Sin embargo, para ese caso, tenemos el botón de “reconstruir carpeta”, que cargará los correos de nuevo y acabará con cualquier inconsistencia que pueda haberse producido.

8.8.5.2 *Cuerpo de los correos*

La funcionalidad de limpiar el cuerpo de los correos para eliminar referencias a otros parece sencilla, a priori. Sin embargo, presenta muchas más dificultades de las previsibles. Por un lado, cada proveedor maqueta los correos de una manera, de ahí las diferentes implementaciones para cada uno. Esto también nos presenta la dificultad de qué sucede si se mezclan proveedores en la misma conversación, habría que hacer un detector de cuerpos de correo para escoger el eliminador adecuado. Por otro lado, dentro del mismo proveedor pueden aparecer discrepancias, por ejemplo, en función del idioma el remitente saldría como “De” si es en castellano o “From” si es en inglés. Para más dificultad, todo el contenido es editable, por tanto, si el usuario modifica el cuerpo, sería muy complejo poder detectar el cambio. Además, esto es algo frecuente, pues es común responder sobre el propio correo recibido para una mejor comprensión de la respuesta si se tratan diferentes temas. En definitiva, no hay una solución clara. Quizás la recopilación de un gran corpus de correos nos permitiría abarcar muchos más casos de uso, pero es una tarea que se escapa de nuestro tiempo y recursos.

En definitiva, la solución tomada fue recoger algunos casos frecuentes con los ejemplos de los que disponíamos, que proporcionan una salida aceptable en las situaciones más frecuentes. En los casos en los que esto se logra, el efecto que consigue al final en el acta es muy bueno, pudiendo seguir la conversación recogida con muchísima más fluidez.

Capítulo 9. Manuales del Sistema

Para el correcto uso del sistema debemos proporcionar al usuario una serie de manuales que le expliquen las acciones que debe realizar. Se detalla cómo instalar las herramientas necesarias para correr la aplicación, como levantar los contenedores, las principales funcionalidades del sistema y como realizar las modificaciones más frecuentes, este último caso destinado a desarrolladores.

9.1 Manual de Instalación

La aplicación no necesita instalación, sin embargo, si depende de software externo para su despliegue. Para poder hacerlo debemos instalar Docker. Este software se actualiza con mucha frecuencia y varía dependiendo del equipo del usuario, por lo que no es posible realizar un manual detallado para su instalación. Sin embargo, sus propios desarrolladores mantienen un manual de instalación propio que puede consultarse aquí: <https://docs.docker.com/get-docker/>. En las últimas versiones de Docker, “compose” se puede utilizar como comando, por lo que no requiere de instalaciones extra. Los comandos que se proporcionan están diseñados para hacerlo de esta forma.

9.2 Manual de Ejecución

Para ejecutar la aplicación se proporcionan dos archivos .bat. Se encuentran en la carpeta "UNA". Uno de ellos es el "build.bat", que primero compila el código fuente de ambas capas y después despliega los contenedores Docker. Este archivo no sería necesario utilizarlo, pues ya se proporciona la aplicación compilada, por tanto, para desplegarla se puede usar el archivo "run.bat" que realiza sólo el segundo paso, arrancar los contenedores. Ambos se pueden ejecutar en un sistema Windows haciendo doble clic.

9.3 Manual de Usuario

Este manual contiene la explicación detallada de todas aquellas acciones que puede hacer el usuario. Aunque una buena interfaz debe dar pie a pocas equivocaciones, se hace necesario un manual como este para sacarle el máximo partido a la aplicación y resolver aquellas dudas que puedan surgirle al usuario.

En las próximas líneas haremos una introducción al flujo de la aplicación y posteriormente pasaremos a detallar el uso de cada componente. Finalmente, se detallan los *endpoints* de la API y unas nociones básicas para su uso.

9.3.1 Introducción

La aplicación web UNA pretende tener un flujo lineal, es decir, una serie de pasos que deben seguirse en un orden para alcanzar el objetivo final, en este caso la generación de un acta de reunión. Para lograrlo, el usuario debe realizar las siguientes tareas:

1. Abrir una conexión con su proveedor de correo.
2. Navegar por sus carpetas y correos allí almacenados, seleccionando aquellos con los que le interese generar un acta.
3. Seleccionar las personalizaciones que quiere sobre esa acta (formato, títulos, qué metadatos se incluyen y cuáles no...).
4. Descargar el documento para su almacenamiento local.

En las siguientes líneas se describirá como realizar todas estas tareas, así como otras auxiliares, que se salen del flujo normal y que son útiles en determinados casos.

La aplicación está testada en los navegadores Chrome, Firefox y Edge, por lo que se recomienda el uso de alguno de estos.

Es importante reseñar que si se ejecuta en el equipo local con las opciones por defecto, podrá accederse a través de la URL <https://localhost:4200>.

9.3.2 Cambiar Idioma

Antes de nada, hay que seleccionar el idioma que queremos utilizar para manejar la aplicación. Por defecto, se muestra en el idioma preferido que esté configurado en el navegador, de no haber ninguno o no ser soportado aquel que esté elegido, se seleccionará el castellano. En cualquier caso, se podrá cambiar el idioma usando el componente de la aplicación cuya interfaz se muestra en la Figura 91. Clicando sobre cada una de las banderas la web se mostrará en asturiano, castellano o inglés, respectivamente.



Figura 91 – Componente cambio de idioma

9.3.3 Abrir/Cerrar conexión

En este punto se explica como abrir una conexión con el proveedor de correo. Se usarán números para aludir a cada uno de los elementos de la foto.

Figura 92 – Componente abrir/cerrar conexión

[1] -> En este campo se introducirá el correo electrónico del usuario.

[2] -> En este otro campo se introducirá la contraseña del correo especificado en [1].

[3] -> Aquí se especificará el host del servicio de correo, p.ej. para las cuentas de la forma “xxxxxx@uniovi.es” el host correspondiente es smtp.office365.com. Los proveedores de correo suelen hacer público el host al que debe uno conectarse, por lo que será el usuario quien deba asegurarse de cuál es que debe poner. Escribir un host en este campo no es necesario para todos los proveedores, UNA lo establece de manera automática cuando se utiliza uno de los siguientes dominios: uniovi.es o gmail.com.

[4], [5] y [6] -> De la misma manera que el host se debe especificar el puerto, el protocolo y si se desea usar SSL. Actualmente sólo se soporta el protocolo IMAP.

[7] -> Este botón muestra u oculta los campos [3], [4], [5] y [6] que inicialmente aparecen ocultos y no se muestran a menos que se intente abrir una conexión con un dominio no soportado.

[8] -> Si se usan cuentas del dominio “gmail.com” no servirá escribir en el campo [2] la contraseña habitual con la que se accede al correo. Clicando en este icono [8] se mostrarán las instrucciones para obtener una “contraseña de aplicación”, necesaria cuando se usa el correo de Google.

[9] -> Clicando en este botón se abrirá una conexión nueva utilizando las credenciales especificadas en los diferentes campos. El éxito o fallo de la conexión se mostrará a la derecha. En caso de error puede aparecer información más detallada en la consola del navegador. Si la conexión está abierta, el botón cambiará su texto a “Reconectar” y al clicarlo intentará abrir una nueva conexión de la misma manera.

[10] -> Permite cerrar la conexión actual, limpiando todos los datos de la sesión almacenados en el sistema.

Por último, hay que advertir al usuario que la apertura y cierre de demasiadas conexiones simultáneas o en un corto periodo de tiempo, puede provocar el bloqueo temporal de la cuenta por parte del proveedor. Si esto sucede servirá con esperar unos minutos, o en el caso de que sea una prueba local reiniciar el contenedor del *back-end*.

9.3.4 Listar Correos

Una vez que se abra la conexión, se procederá a cargar las carpetas y los correos. La primera vez que se accede a la aplicación puede tardar unos minutos en sincronizarse, pues se descargarán los cien primeros correos de cada carpeta. En posteriores accesos simplemente se añadirán los nuevos correos, tardando mucho menos. Una vez que estén correctos se mostrará el componente de la Figura 93.

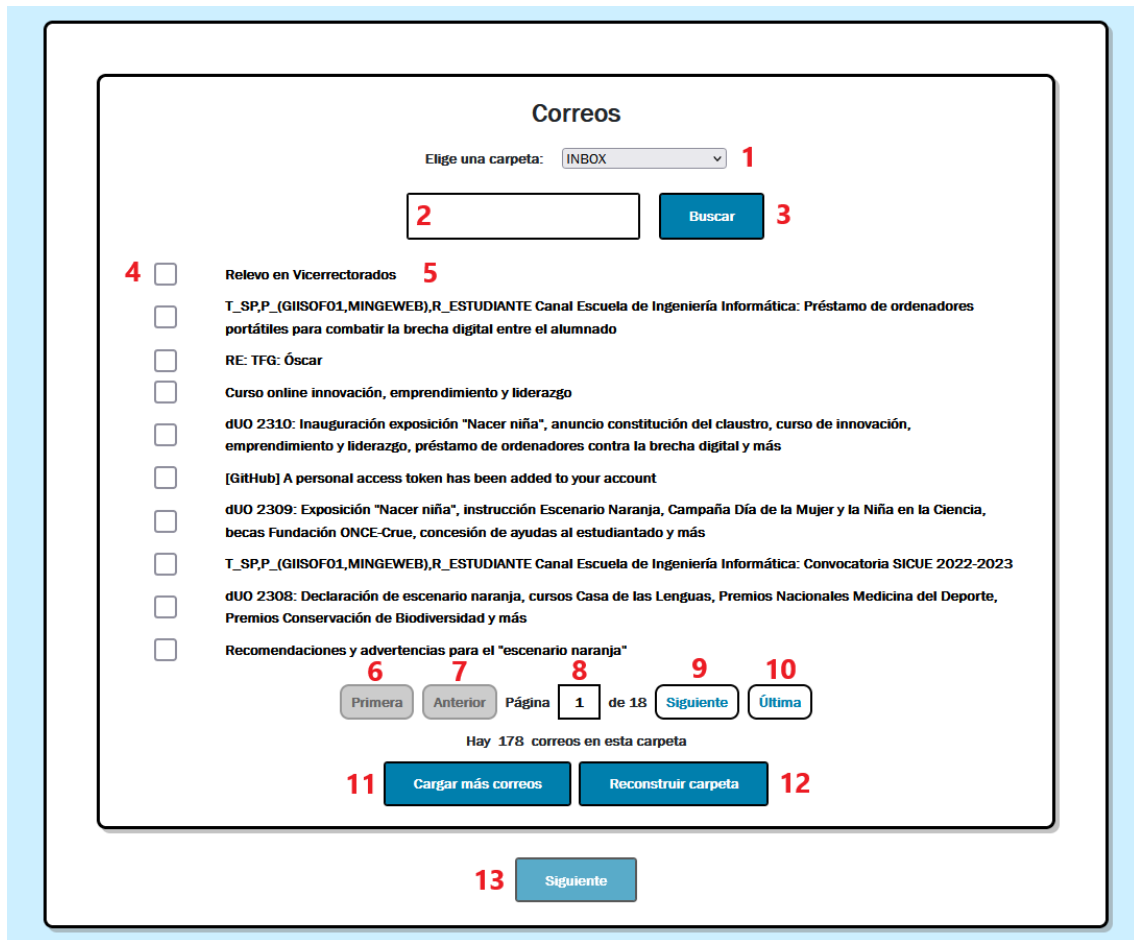


Figura 93 – Componente Listado de Correos

[1] -> Este selector permite elegir la carpeta que se quiere mostrar. El desplegable muestra las carpetas ordenadas por orden alfabético. Pueden generarse actas con correos de diferentes carpetas.

[2] y [3] -> Esta parte de la interfaz es el buscador. Permite filtrar los correos de la carpeta seleccionada utilizando palabras que aparezcan en el asunto. No distingue mayúsculas de minúsculas ni letras acentuadas.

[4] -> El *checkbox* que hay junto a cada uno de los correos permite seleccionar aquellos que formarán parte del acta. Si está marcado, se entenderá que se quiere añadir. No hace falta seleccionar todos los correos que formen parte de una conversación. Una vez seleccionado el último, se añadirán en el acta todos los correos relacionados.

[5] -> Clicando sobre el asunto de los correos, aparecen los datos del mismo, del modo que se ve en la Figura 94. Se muestra la fecha de envío, las direcciones de envío y recepción, las cuentas a las que se envió copia de carbón y el contenido del correo. Este se mostrará como texto plano si así se ha enviado o como HTML si es el caso. Si se intentase cargar un correo eliminado se indicará este hecho y no se cargará.

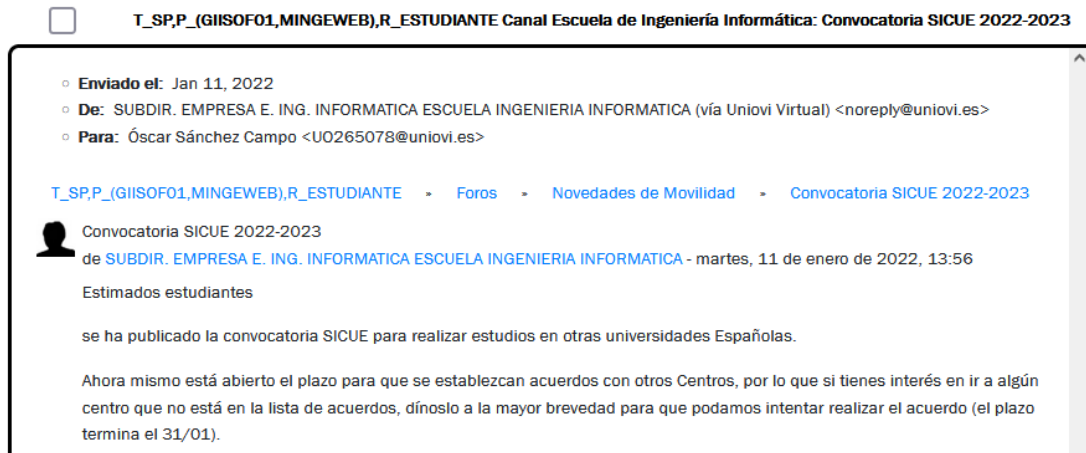


Figura 94 – Datos del correo (extendido)

[7] y [9] -> Los correos se muestran en páginas de diez elementos. Usando estos botones se puede mover uno a la página siguiente o a la página anterior. Cada vez que se cargue por primera vez una página, habrá unos segundos de espera. La segunda vez que esta se cargue será mucho más rápido.

[6] y [10] -> Estos botones permiten ir a la primera y a la última página de entre todas las de los correos.

[8] -> Aquí puede introducirse un número para ir a una página no contigua, p.ej., estando en la página 3, ir a la página 7.

[11] -> Como se advierte en el primer párrafo, en cada carpeta se cargarán los cien primeros correos. Si el usuario quiere acceder a correos más antiguos podrá usar este botón para continuar cargando correos de cien en cien.

[12] -> En contadas ocasiones en las que sucedan problemas temporales de conexión los correos pueden quedar en un estado inconsistente. También puede ser que queramos trabajar con menos correos o limpiar todos los que se hayan eliminado de manera automática. En estas situaciones pueden recargarse los correos de la carpeta seleccionada usando este botón.

[13] -> Permite continuar con el proceso de generación del acta, pasando a la pantalla de exportación. Sólo se podrá avanzar si hay al menos un correo seleccionado.

9.3.5 Exportación

El componente de exportación permite personalizar el acta, para ello deben seleccionarse una serie de opciones.

←

Selecciona el formato en que se generará el acta

PDF (Portable Document Format) 1

Datos que se añadirán al acta:

2 3 Título 4 5

Fecha 23/11/2021, 15:30:31

6

Configuración

7 Incluir direcciones (Remitente, destinatario, CC)

8 Mostrar la fecha de los correos exportados

9 Incluir nombre de los archivos adjuntos

10 Incluir resumen participantes

Se van a exportar los siguientes correos:

11 • T_SP,P_(GIISOFO1,MINGEWEB),R_ESTUDIANTE Canal Escuela de Ingeniería Informática: Oferta prácticas desarrollador de software

12

13

Figura 95 – Componente de exportación

[1] -> Este selector permite elegir el formato en el que se desea el acta. Hay dos opciones: PDF o CSV. El acta generada en PDF contiene todas las personalizaciones que se deseen y está diseñado para ser “leído” por otros usuarios. Si se genera en CSV, no se podrán añadir campos personalizados, este formato está diseñado para su procesamiento por otras máquinas.

[2] -> Este bloque contiene los campos personalizados a añadir. Estos se mostrarán en el acta generada como una serie de etiquetas y valores. Por ejemplo, si la etiqueta es “título” y el valor es “Acta de ejemplo”, al principio del acta se mostrará “título: Acta de Ejemplo”. Los *checkboxes* permitirán desactivar cada uno de los campos añadidos, esto puede ser útil si se desean generar actas con los mismos correos, pero diferentes campos.

[3] -> Aquí se especificará la etiqueta del campo personalizado.

[4] -> Aquí se especificará el valor del campo personalizado.

[5] -> Clicando este botón podrá eliminarse el campo personalizado.

[6] -> Con este botón se añadirá un nuevo campo.

[7] -> En este bloque se pueden elegir diferentes opciones para añadir al acta. Esta en concreto permite especificar si se quieren incluir las direcciones de correo.

[8] -> Marcando esta opción se incluirá en el acta generada, junto al cuerpo de los correos, la fecha de envío de cada uno.

[9] -> Aquí se selecciona si Junto al cuerpo de los correos se quiere añadir el nombre de los archivos adjuntos que tiene.

[10] -> Con esta opción se puede añadir una tabla al principio del acta resumiendo los participantes de los correos exportados.

[11] -> Esta lista resume los correos que se añadirán. Además de los aquí listados, todos los relacionados con estos se añadirán también.

[12] -> Clicando aquí se generará el acta y se descargará el navegador. En función del número de correos seleccionados puede tardar unos pocos segundos en generarse o algunos minutos.

[13] -> Si los correos contienen archivos adjuntos se podrán descargar clicando en este botón. Estos también pueden tardar en descargarse. Será un archivo ZIP que contendrá, además de los archivos adjuntos, el cuerpo de los correos en formato HTML.

9.3.6 Acta

El acta en sí es un archivo PDF que puede verse en la Figura 96. Está compuesta de varias partes:

[1] -> Aquí se incluyen los campos personalizados especificados en la exportación. Además de la fecha de inicio y fin de los correos.

[2] -> Esta tabla es resultado de haber marcado la opción de exportación “Incluir resumen participantes”. Incluye los correos que participan en la conversación y cómo lo hacen.

[3] -> Después vienen los correos con los datos que se hayan querido incluir.

Acta 1**Título:** Acta de Ejemplo**Fecha del Acta:** 13 de enero de 2021**Fechas de inicio y fin de los correos:** miércoles 8 de diciembre de 2021 11H00' GMT al miércoles 8 de diciembre de 2021 11H04' GMT**Participantes 2**

Participante	Remitente	Destinatario	CC
"Óscar Sánchez Campo" <sc134@gmail.com>	2	2	0
Pruebas CorreosTFG <pruebascorreostfg@gmail.com>	2	2	0
<ez134@gmail.com>	0	1	0

Correos 3**Re: Correo de Prueba****- Remitente:** "Óscar Sánchez Campo" <sc134@gmail.com>**- Destinatario:** Pruebas CorreosTFG <pruebascorreostfg@gmail.com>**- Fecha de envío:** miércoles 8 de diciembre de 2021 11H04' GMT

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris eleifend massa dictum turpis consectetur, vitae finibus ipsum convallis. Nunc sit amet augue at nisi vehicula fermentum. Nulla leo nisl, pellentesque nec aliquam eu, ornare vestibulum tortor. Quisque tincidunt ligula vel ex facilisis fermentum. Sed id vestibulum tellus. Ut porta leo egestas, consequat nunc in, aliquam nulla. Vivamus volutpat, enim quis aliquam maximus, urna lectus dapibus nunc, eget mattis velit lacus et justo. Nullam tortor enim, suscipit quis justo et, euismod condimentum nisl. Interdum et malesuada fames ac ante ipsum primis in faucibus. Mauris odio lacus, bibendum at efficitur at, euismod at quam. Proin fringilla interdum ligula, a faucibus risus dictum a. Nulla vel maximus purus, eu congue quam. Aenean id tempus tellus. Duis luctus pellentesque turpis eget pharetra.

**Re: Correo de Prueba****- Remitente:** Pruebas CorreosTFG <pruebascorreostfg@gmail.com>**- Destinatario:** sc134@gmail.com, ez134@gmail.com**- Fecha de envío:** miércoles 8 de diciembre de 2021 11H04' GMT

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris eleifend massa dictum turpis consectetur, vitae finibus ipsum convallis. Nunc sit amet augue at nisi vehicula fermentum. Nulla leo nisl, pellentesque nec aliquam eu, ornare vestibulum tortor. Quisque tincidunt ligula vel ex facilisis fermentum. Sed id vestibulum tellus. Ut porta leo egestas, consequat nunc in, aliquam nulla. Vivamus volutpat, enim quis aliquam maximus, urna lectus dapibus nunc, eget mattis velit lacus et justo. Nullam tortor enim, suscipit quis justo et, euismod condimentum nisl. Interdum et malesuada fames ac ante ipsum primis in faucibus. Mauris odio lacus, bibendum at efficitur at, euismod at quam. Proin fringilla interdum ligula, a faucibus risus dictum a. Nulla vel maximus purus, eu congue quam. Aenean id tempus tellus. Duis luctus pellentesque turpis eget pharetra.

Figura 96 - Acta

9.3.7 Cookies

La aplicación sólo almacena las cookies esenciales para su funcionamiento. Este aviso se muestra la primera ocasión en la que se entra a la aplicación y puede ocultarse clicando sobre el icono de la galleta. Si se quiere volver a ver, este icono permanecerá siempre en la parte inferior izquierda, pudiendo clicarse sobre él para volver a mostrar el aviso.



UNA no usa más cookies que las necesarias para su funcionamiento.

9.3.8 Para programadores

La API de la aplicación es accesible para todos los usuarios, pudiendo realizar peticiones directamente a ella. Se podrá acceder a través del puerto 8080, en local <https://localhost:8080>. Los *endpoints* que tiene son los siguientes:

POST	/connection/open	Respuesta		Parámetros	
		<i>value</i>	String	<i>email</i>	String
				<i>password</i>	String
				<i>host</i>	String
				<i>puerto</i>	String
				<i>protocolo</i>	String
				<i>ssl</i>	boolean
GET	/connection/close	Respuesta		Parámetros	
		<i>value</i>	String		
POST	/connection/sincronizar	Respuesta		Parámetros	
		<i>value</i>	String	<i>carpeta</i>	String
				<i>deCero</i>	boolean
POST	/mails	Respuesta		Parámetros	
		<i>List <Correo></i>		<i>inicio</i>	int
				<i>fin</i>	int
				<i>carpeta</i>	String
				<i>busqueda</i>	String
POST	/mail	Respuesta		Parámetros	
		<i><Correo> c</i>		<i>idCorreo</i>	String
		<i>c.id</i>	String	<i>carpeta</i>	String
		<i>c.nombreLargoCarpeta</i>	String		
		<i>c.subject</i>	String		
		<i>c.fecha</i>	Date		
		<i>c.cuerpo</i>	String		
		<i>c.remitente</i>	String		
		<i>c.destinatario</i>	String		
		<i>c.cc</i>	String		
		<i>c.nombresAdjuntos</i>	String[]		
		<i>c.completo</i>	boolean		
GET	/mail/count	Respuesta		Parámetros	
		<i>value</i>	Integer	<i>carpeta</i>	String
				<i>busqueda</i>	String
GET	/mail/folders	Respuesta		Parámetros	
		<i>List<Carpeta></i>			
		<i><Carpeta> f</i>			
		<i>f.nombreCorto</i>	String		
		<i>f.nombreLargo</i>	String		
GET	/export	Respuesta		Parámetros	
		<i>value</i>	byte[]	<i>format</i>	String
				Export Bean	
				<i>idsCorreos</i>	Integer[]

GET		/export/attachments		Respuesta		Parámetros	
		value	byte[]	ExportBean			
				idsCorreos	Integer[]	idsCarpetas	String[]
				labels	String[]	values	String[]
				opciones	String[]	idioma	String

Tabla 20 – Referencia de la API

Para poder realizar de manera satisfactoria las peticiones, debe haberse llamado primero a /connection/open. El resto deben hacerse con el mismo valor para la cookie JSESSIONID.

9.4 Manual del Programador

El sistema ha sido diseñado con vistas a su ampliación y contiene algunas partes del sistema fácilmente incrementables. Por ejemplo, las opciones de exportación o los hosts. En las próximas líneas se darán unas directrices de cómo se tendrían que realizar modificaciones, en todo el sistema, y en estas partes específicas.

El sistema se presenta en dos proyectos diferentes, uno de Angular para el *front-end* y otro de SpringBoot para el *back-end*.

Normalmente para incluir una funcionalidad se deben realizar una serie de modificaciones:

- Incluir los nuevos datos en las clases del modelo del *back-end*. Por ejemplo, en `MailServer` o en alguno de los *parsers*.
- Añadir un nuevo *endpoint* en los controladores.
- Añadir en los servicios del *front-end* la llamada al *endpoint* creado.
- Modificar los componentes de la interfaz o crear uno nuevo si fuese necesario.

Para el traspaso de datos se usan `Beans`, en ambas partes del sistema. En la aplicación SpringBoot también hay entidades, que se mapean a `Beans` antes de ser devueltos.

Las funcionalidades de las que hablábamos antes, que pueden ser fácilmente ampliables se van a ver en los siguientes apartados.

9.4.1 Hosts

De algunas direcciones de correo la aplicación obtiene automáticamente las credenciales: host, puerto, protocolo... Estos valores se definen en un archivo contenido en el *front-end*, en la ruta `/assets/files/hosts.json`. Es un archivo que contiene un objeto JSON con un array "hosts" en el que se definen las credenciales. Añadiendo un nuevo valor a esta lista soportaremos un proveedor más:

```
{
  "hosts": [
    {
      "domain": "uniovi.es",
      "host": "outlook.office365.com",
      "port": "993",
      "protocol": "imap",
      "ssl": true
    },
    {
      "domain": "gmail.com",
      "host": "imap.gmail.com",
      "port": "993",
      "protocol": "imap",
      "ssl": true
    }
  ]
}
```

9.4.2 Opciones de exportación

Las opciones de exportación tienen un poco más de complejidad, pues deben definirse tanto en el *front-end* como en el *back-end*. En la primera capa se define una lista de opciones en el método `ngOnInit` del componente de exportación. Cada una se añade utilizando el método `push` y añadiendo un objeto como el siguiente, donde `id` es el identificador que usamos para opción, `label` tiene que ser una etiqueta definida en los archivos de internacionalización para los diferentes idiomas y `selected` será `true` si se desea mostrar marcada por defecto en la interfaz.

```
this.opciones.push({
  id: 'incluir_direcciones',
  label: 'p_incluir_direcciones',
  selected: true
});
```

Una vez definida aquí deberemos añadirla en el *back-end* para poder utilizarla. El objeto `ExportBean` que es el que se usa para realizar la petición del acta tiene un parámetro que es un array de strings. Contiene los identificadores de las opciones seleccionadas en la exportación. Estos se utilizan en la construcción del generador, concretamente a través de la utilidad `OptionsManager`. En esta clase hay que realizar cuatro modificaciones:

1. Añadir una constante que defina el identificador de la nueva opción.
2. Incluir una variable booleana que indique si se ha seleccionado o no.
3. Modificar el constructor para establecer correctamente la variable creada.
4. Crear un método accesor a esa variable.

Una vez realizados estos pasos, ya podrá utilizarse la nueva opción de exportación en cualquier parte del generador de actas en la que la necesitemos.

9.4.3 Eliminador de Respuestas

Otra de las funcionalidades creadas para ser fácilmente ampliada o modificada es el Eliminador de Respuestas. Su objetivo es limpiar el cuerpo de los correos, quitando las referencias a otros de la misma conversación. Cada proveedor tiene su propia manera de definirlos, por lo que en el paquete `parsers/elimina_respuestas` del *back-end* hay que definir una nueva clase por proveedor.

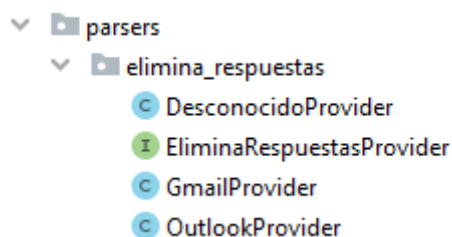


Figura 97 – Paquete `elimina_respuestas`

Se usa el patrón de diseño Estrategia (*Strategy*). La estrategia sería la interfaz `EliminaRespuestasProvider` y debe implementarse con una estrategia concreta por cada proveedor soportado. La interfaz tiene un solo método `eliminaRespuestas` que debe sobreescribirse.

Capítulo 10. Conclusiones y Ampliaciones

Tras todo el trabajo desarrollado, este capítulo es el lugar conveniente para hacer balance del producto resultante y plantear ciertas mejoras que podrían hacerse si se continuase desarrollando el proyecto. Lo dividiremos en dos partes, primero las conclusiones y posteriormente las posibles ampliaciones.

10.1 Conclusiones

Tras varios meses de trabajo, muchas funcionalidades desarrolladas y un buen paquete de pruebas que desembocaron en la implantación de varias mejoras, las conclusiones no pueden ser más que buenas.

Respecto al producto final, este cumple en mayor o menor medida con los objetivos planteados al principio del proyecto. Consideramos que es una herramienta sencilla de utilizar y muy útil para las necesidades que está pensada. Las pruebas realizadas con usuarios y la apuesta por unos componentes pulidos y una interfaz cuidada, en lugar de muchas funcionalidades poco trabajadas, garantizan la adaptabilidad a diferentes usuarios que buscábamos.

Aspectos como la compatibilidad con muchos proveedores se han logrado de manera satisfactoria, al menos en cuanto a su uso básico, aunque las implementaciones propias de cada uno puedan ocasionar algún problema de apariencia.

Algunas funcionalidades que se buscaban han resultado sencillas de lograr. Por ejemplo, la internacionalización no ha supuesto ningún problema. Sin embargo, otros aspectos han ganado o perdido importancia a lo largo del desarrollo. La exportación a otros formatos o el uso de la API es un ejemplo de lo segundo, que si bien se incluyen entre las funcionalidades no brillan como se esperaba al principio. Por el contrario, la orientación a la generación en formato PDF si ha centrado la mayor importancia. Las actas presentan un aspecto muy cuidado y se presentan como producto final y principal de todo el flujo de la aplicación.

También es importante hacer balance del proceso de construcción. El proyecto se planteó con un par de metodologías de trabajo, por un lado, Scrum para el desarrollo del software y por otro un desarrollo en cascada para la documentación. El planteamiento no fue malo, si bien la aplicación de Scrum y muchos de sus artefactos, se quedó algo descafeinada por aquello de no contar con un Product Owner. No obstante, el desarrollo por historias, el uso de *sprints* y demás artefactos, fue muy beneficioso y permitió aplicar algunas herramientas en el ámbito de la dirección de proyectos de este tipo. Sin embargo, el punto en el que menos acertado se estuvo fue en la planificación de las pruebas con los usuarios, que se realizaron demasiado tarde por problemas de recursos y que de haberse hecho antes quizás el desarrollo no se hubiese alargado tantas horas.

Como en todo proyecto, nos quedamos con una serie de aspectos y cosas que haríamos de diferente manera, valiosas lecciones que nos permitirían afrontar otro de este estilo con mayor eficacia y en un mucho menos tiempo.

En definitiva, consideramos UNA un gran éxito y su desarrollo un trabajo, que, aunque requirió muchos esfuerzos, ha proporcionado también gran cantidad de habilidades y experiencia práctica de cara a futuros proyectos.

10.2 Ampliaciones

Al contrario que en otros campos de la ingeniería, en informática es difícil dar un producto por terminado, sobre todo uno que está de cara un público. Siempre pueden surgir aspectos a mejorar basados en su experiencia o nuevas funcionalidades que estos demanden. Por otro lado, nuestra aplicación no vive en un mundo aislado, se integra con otras tecnologías, como el correo electrónico. Este tipo de integraciones puede obligar a cambios en nuestros sistemas si estas evolucionan, de cara a mejorar compatibilidades o a aprovechar sus nuevos avances. En definitiva, si este proyecto continuase en un entorno productivo, sin ninguna duda debería ir actualizándose. En las próximas líneas describiremos algunas de las funcionalidades que podría tener y que no se han implementado en esta versión, bien sea por ser muy complejas en cuanto su realización o por no disponer del tiempo suficiente para ello.

Una de las ampliaciones que podría realizarse es ampliar el soporte a más proveedores de correo. Si bien la aplicación está diseñada para facilitar la introducción de nuevos hosts, en el tema de la eliminación de las referencias a otros correos en el cuerpo de los mismos se queda un poco corta. Esta es una problemática que se describe en el apartado 8.8.5.2. Solucionarlo no es trivial y requeriría de la construcción de un módulo propio. Debería recopilarse un corpus de correos en muchos idiomas, formatos y proveedores que posteriormente se usaría para realizar un análisis comparativo que permita detectar las partes que son referencia. De conseguir esto, las actas mejorarían aún más su apariencia.

Otra posible mejora sería la creación de un proceso en segundo plano que se dedique a limpiar los correos eliminados del proveedor y que realice la carga de los restantes sin causar perjuicios al usuario. Esto tendría más sentido si la aplicación estuviese desplegada en un servidor, para que pudiese funcionar fuera de las horas de uso. Actualmente esto se realiza de manera síncrona, obligando al usuario a soportar tiempos de espera. No obstante, estos se han intentado reducir al mínimo logrando un equilibrio entre tiempo y funcionalidad.

Respecto a la exportación, también podría trabajarse de cara a la personalización de las actas. Sería fácil crear nuevas opciones de exportación que añadiesen campos calculados al acta, aunque estos suelen ser específicos del dominio y del contexto de la reunión que se refleje. Una opción más ambiciosa y que daría cabida a todos ellos sería la creación de un DSL (*Domain Specific Language*), que permita realizar cálculos con los datos de los correos y para que estos se vean reflejados en el acta. Sin embargo, el desarrollo de esta funcionalidad sería casi un proyecto propio y estaría más bien destinada a unos usuarios muy expertos, que a un usuario básico de correo electrónico como planteamos aquí.

Capítulo 11. Planificación del Proyecto y Presupuesto finales

En este capítulo mostraremos la planificación y presupuestos finales del proyecto, estableciendo comparativa con aquellos planteados como iniciales en el Capítulo 4. Veremos si nos hemos ajustado en mayor o menor medida a ellos y justificaremos las diferencias entre lo planteado y lo ejecutado. Además, incluiremos algunos artefactos de Scrum para medir el avance.

11.1 Planificación Final

La planificación inicial se resumía en la Figura 4 por medio de un cronograma. Allí se planteaban una serie de tareas a realizar y su duración prevista. La Figura 98 representa el mismo esquema con las tareas una vez finalizadas. Vamos a analizar las diferencias entre ambos.

La más evidente de todas ellas es la duración del proyecto. Habíamos previsto su finalización para principios de noviembre de 2021, sin embargo, tras las pruebas con los usuarios, se decidió continuar unas semanas más el desarrollo para implementar las mejoras que se comentan en el apartado 8.8. Al final, el proyecto finalizó la segunda semana de febrero de 2022. Alargándose unas 12 semanas más de lo previsto.

En la planificación inicial se contemplaba un periodo vacacional las dos últimas semanas de julio. Tras la decisión de alargar el desarrollo, se añadió otro en las dos últimas semanas de diciembre. Ambos aparecen marcados en gris en los cronogramas.

Vamos a ir viendo las tareas cronológicamente. La primera diferencia fue el alcance y los objetivos del proyecto, pues se pretendía partir de las EPIC definidas en la planificación inicial, sin embargo, tras una reflexión inicial se consideró importante y se realizó esa tarea.

Esto retrasó ligeramente el inicio del primer *sprint* y se optó por reducir su tiempo para no variar las fechas del resto. Aunque se había previsto construir el sistema de manera continua, es decir, un *sprint* detrás de otro, el periodo vacacional de julio dividía en dos este segundo *sprint*. No era deseable dejar historias abiertas y, por otro lado, con el primer *sprint* finalizado, ya se podían ir redactando cosas de la documentación para avanzar en los dos frentes. En las semanas pre y postvacacionales se realizó una primera fase de la documentación creando un borrador del análisis y el diseño, que, aunque luego se modificó, ya tenía unas bases seguras.

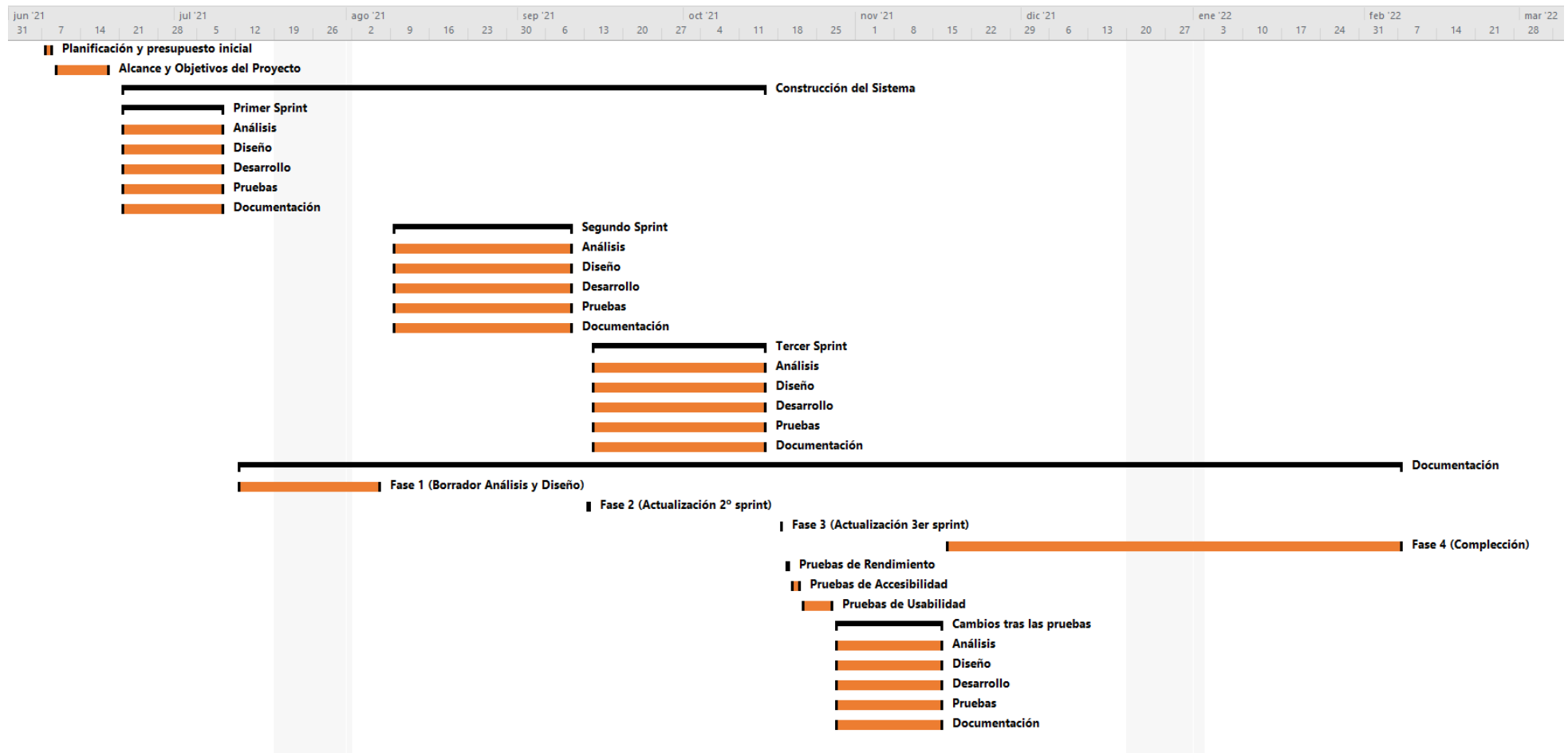


Figura 98 – Planificación Final del Proyecto

No obstante, al realizar esta tarea, la construcción del sistema se alargó dos semanas más, finalizando el 15 de octubre. Por otro lado, también se actualizó ese análisis y diseño entre el segundo y tercer *sprint*, reduciendo en un día el tercero para realizar esa tarea. En definitiva, las variaciones de tiempo de los *sprints* entre lo planificado y lo ejecutado es la siguiente.

	Comienzo		Fin		Horas	
	Previsto	Real	Previsto	Real	Previsto	Real
<i>Sprint 1</i>	9 de junio de 2021	21 de junio de 2021	9 de julio de 2021	9 de julio de 2021	69	45
<i>Sprint 2</i>	12 de julio 2021	9 de agosto de 2021	27 de agosto de 2021	10 de septiembre de 2021	75	75
<i>Sprint 3</i>	30 de agosto 2021	14 de septiembre de 2021	1 de octubre de 2021	15 de octubre de 2021	75	72

Tabla 21 – Resumen final de fechas del *sprint*

Tras la finalización del tercer *sprint* se dedicó una jornada a actualizar la documentación y se realizaron las pruebas previstas. El resultado de estas obligó a programar un nuevo *sprint* que duró 40,5 horas y resolvió los problemas encontrados, como bien comentábamos antes.

Por último, completar la documentación fue la tarea peor estimada de todas, pues se habían planteado 54 horas para ella y acabaron resultando 210 horas, incluyendo los borradores y actualizaciones intermedias.

Podemos ver que la redacción de la documentación se desvió bastante de las previsiones, pero de momento la construcción del sistema es una caja negra. En las próximas líneas intentaremos explicar por medio del uso de Burndown Charts, uno de los artefactos que Scrum plantea para la medición del trabajo completado, cómo se desarrollaron los diferentes *sprints* en cuanto a tiempos.

11.1.1 Primer Sprint

La primera semana del desarrollo comenzó según lo previsto, se implementaron las historias que abrían y cerraban la conexión, con mayores o menores inconvenientes principalmente fruto de la inexperiencia con las librerías y la necesidad de ciertas configuraciones. Esto ya se había previsto en la estimación de las historias, siendo unos tiempos conservadores los establecidos. En la segunda semana, una vez solucionados los problemas, el desarrollo fue más rápido de lo previsto, aunque en el cambio de mes hubo un pequeño bloqueo. El día 2 de julio las horas restantes igualaron a las estimadas, no yendo a más los problemas y desarrollándose la última semana del *sprint* sin sobresaltos.

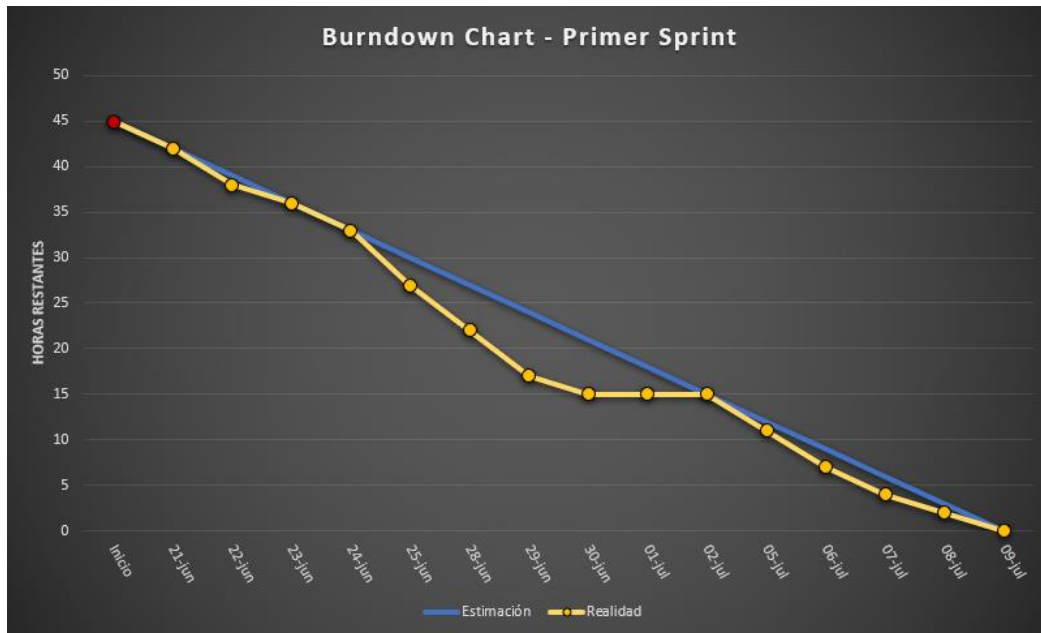


Figura 99 – Burndown Chart del primer sprint.

11.1.2 Segundo Sprint

El inicio del segundo *sprint* vaticinaba unas semanas difíciles, pues surgió el problema de la Previsualización de correos que se describe en el apartado Problemas Encontrados. Sin embargo, una vez tomadas las decisiones oportunas, se fue recuperando el tiempo perdido, llegando a estar bastantes horas por debajo de lo previsto, por ejemplo, el 24 de agosto quedaban casi 17 horas menos de las que se habían previsto. Algún que otro problema surgió en la última semana del *sprint*, entre ellos el de El locale asturiano que se describe en la implementación, pero que no causó graves problemas gracias al margen acumulado.

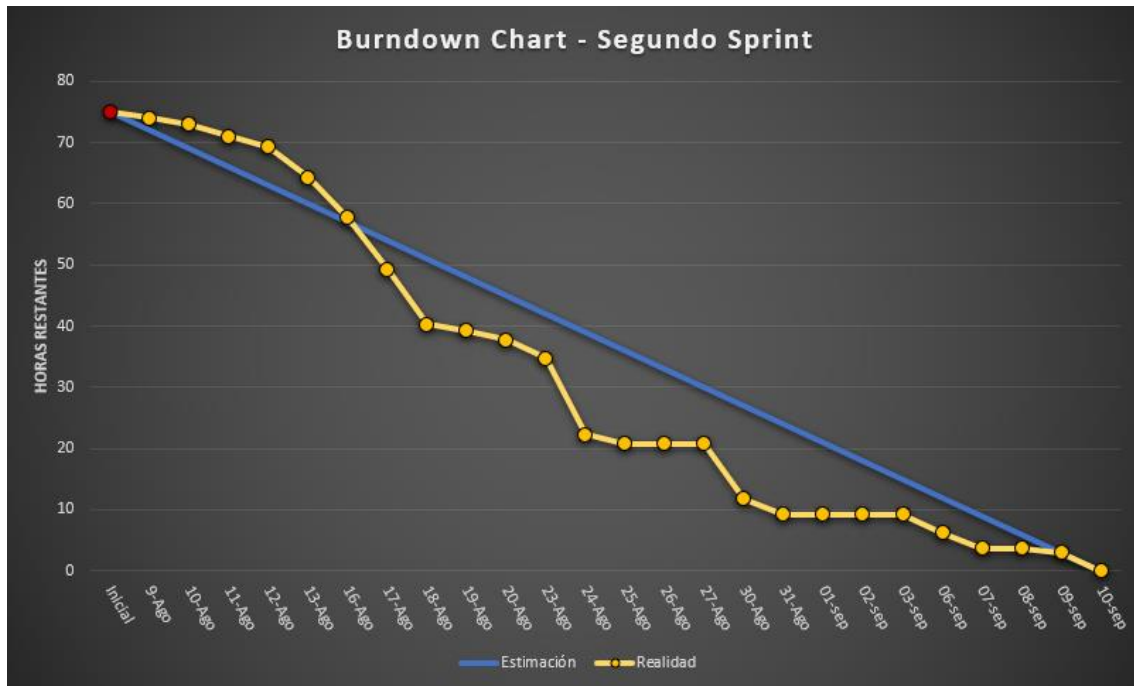


Figura 100 – Burndown Chart del segundo sprint

11.1.3 Tercer Sprint

Con la experiencia de los *sprints* anteriores, las estimaciones del último resultaron mucho más cercanas a la realidad. También hay que valorar aquí otros problemas que surgieron como los Correos HTML o los Tiempos de carga de adjuntos. En los últimos días de trabajo, se acumuló un pequeño retraso que afortunadamente pudo solventarse antes del final.

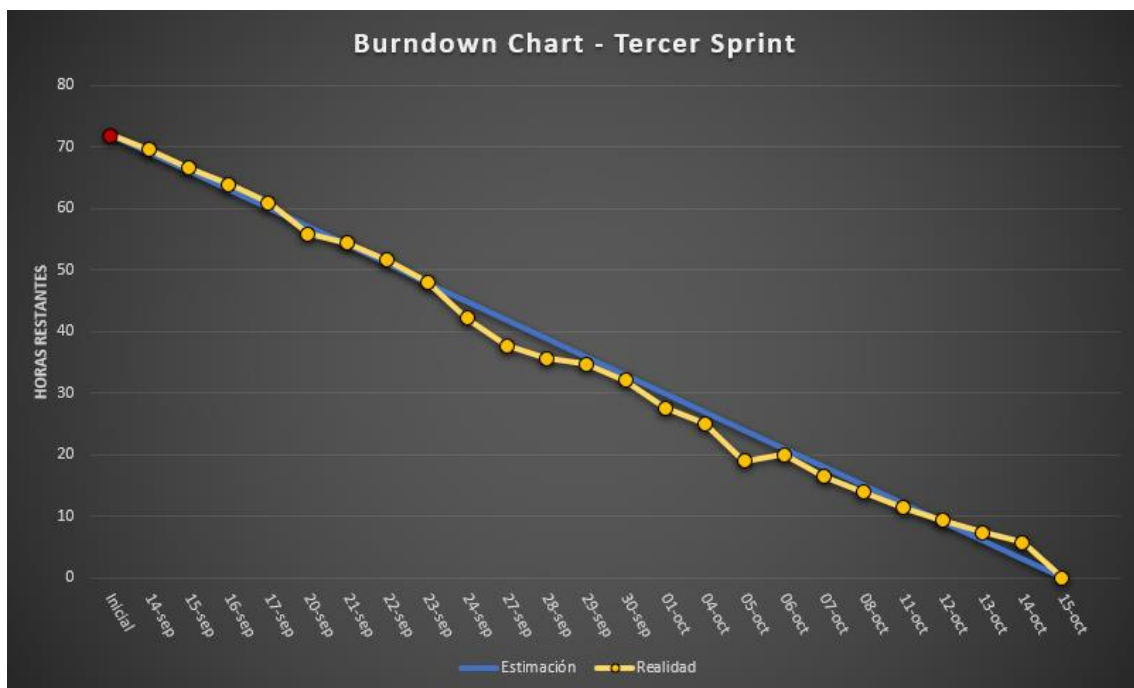


Figura 101 – Burndown Chart del tercer sprint

En definitiva, estas gráficas son un reflejo de los tiempos muy cercano al desarrollo, pues muestran la evolución día a día y ayudan a tomar decisiones importantes respecto al alcance antes de llegar a puntos comprometidos. En definitiva, esta parte se corresponde mucho mejor que la documentación con la planificación inicial, posiblemente por mi propia experiencia, más cercana al desarrollo de software que a la redacción documental, siendo más precisas las estimaciones y aumentando esta precisión con conocimiento de *sprints* previos.

11.2 Presupuesto Final

Las horas por encima de la planificación inicial explicadas en el punto anterior tienen su repercusión en el presupuesto. En las próximas líneas veremos los sobrecostes que producen haciendo una revisión de las cantidades planteadas en el apartado 4.2.

El primer punto de aquel apartado es la definición de la empresa, donde se plantan los costes indirectos de su actividad, los costes de los equipos y el personal, en definitiva, aquellos referentes al funcionamiento general de la empresa. Estos no hace falta revisarlos, pues se calculan sobre un año natural y no han variado (no han hecho falta más equipos, ni personal extra, ni ningún otro recurso). Por tanto, se siguen manteniendo los precios de 49,60€/h para desarrolladores y *testers* y 62€/h para el jefe de negocio.

Lo que sí varía es el presupuesto del proyecto, para la empresa y para el cliente, que es lo que analizaremos en los siguientes subapartados.

11.2.1 Presupuesto de Costes

En la siguiente tabla se muestran los nuevos costes y las horas de dedicación a cada *item*. Como comentábamos la documentación aumenta considerablemente el número de horas. Por otro lado, este incremento se combina con los cambios realizados tras las pruebas e incrementa en gran medida las horas de desarrollador y *tester*.

Part.	Item	Recurso	Horas	Precio/h	Total Item
1		Construcción del Sistema			27,869.00 €
	1	Primer Sprint			
		1 <i>Jefe de Negocio</i>	15 -8	62.00 €	930.00 €
		2 <i>Desarrollador</i>	45 -24	49.60 €	2,232.00 €
		3 <i>Tester</i>	45 -24	49.60 €	2,232.00 €
	2	Segundo Sprint			
		1 <i>Jefe de Negocio</i>	25 0	62.00 €	1,550.00 €
		2 <i>Desarrollador</i>	75 0	49.60 €	3,720.00 €
		3 <i>Tester</i>	75 0	49.60 €	3,720.00 €
	3	Tercer Sprint			
		1 <i>Jefe de Negocio</i>	24 -1	62.00 €	1,488.00 €
		2 <i>Desarrollador</i>	72 -3	49.60 €	3,571.20 €
		3 <i>Tester</i>	72 -3	49.60 €	3,571.20 €

4	Cambios tras las pruebas					
	1	<i>Jefe de Negocio</i>	13,5	Nuevo	62.00 €	837.00 €
	2	<i>Desarrollador</i>	40,5	Nuevo	49.60 €	2,008.80 €
	3	<i>Tester</i>	40,5	Nuevo	49.60 €	2,008.80 €
2	Documentación					9,523.20 €
	1	Planificación y presupuesto				
	1	<i>Jefe de Negocio</i>	6	0	62.00 €	297.60 €
	2	Redactar documentación				
	1	<i>Desarrollador</i>	186	+132	49.60 €	9,225.60 €
	3	Alcance y Objetivos del Proyecto				
	1	<i>Jefe de Negocio</i>	24	Nuevo	62.00 €	1,488.00 €
3	Pruebas del sistema					1,041.60 €
	1	Pruebas de accesibilidad				
	1	<i>Tester</i>	9	0	49.60 €	446.40 €
	2	Pruebas de usabilidad				
	1	<i>Tester</i>	12	0	49.60 €	595.20 €
TOTAL					38,433.80 €	

Tabla 22 – Presupuesto de costes final del proyecto

11.2.2 Presupuesto de Cliente

El presupuesto de costes se traduce en el del cliente cambiando los precios por hora para incluir el 25% de beneficio que se espera. La tabla que sigue es reflejo de la anterior. Aquí hemos decidido mostrar la variación del coste total de los *items*.

P	Itm	Rec	Horas	Precio/h	Total Item		
1	Construcción del Sistema				34,836.25 €		
	1	Primer Sprint					
		1	<i>Jefe de Negocio</i>	15	77.50 €	1,162.50 €	-620.00 €
		2	<i>Desarrollador</i>	45	62.00 €	2,790.00 €	-1,488.00 €
		3	<i>Tester</i>	45	62.00 €	2,790.00 €	-1,488.00 €
	2	Segundo Sprint					
		1	<i>Jefe de Negocio</i>	25	77.50 €	1,937.50 €	0
		2	<i>Desarrollador</i>	75	62.00 €	4,650.00 €	0
		3	<i>Tester</i>	75	62.00 €	4,650.00 €	0
	3	Tercer Sprint					
		1	<i>Jefe de Negocio</i>	24	77.50 €	1,860.00 €	-77.50 €
		2	<i>Desarrollador</i>	72	62.00 €	4,464.00 €	-186.00 €
		3	<i>Tester</i>	72	62.00 €	4,464.00 €	-186.00 €
	4	Cambios tras las pruebas					
		1	<i>Jefe de Negocio</i>	13,5	77.50 €	1,046.25 €	Nuevo
		2	<i>Desarrollador</i>	40,5	62.00 €	2,511.00 €	Nuevo

	3	Tester	40,5	62.00 €	2,511.00 €	Nuevo
2		Documentación			11,904.00 €	
	1	Planificación y presupuesto				
	1	Jefe de Negocio	6	77.50 €	372.00 €	0
	2	Redactar documentación				
	1	Desarrollador	186	62.00 €	11,532.00 €	+8,184.00 €
	3	Alcance y Objetivos del Proyecto				
	1	Jefe de Negocio	24	77.50 €	1,860.00 €	+1,860.00 €
3		Pruebas del sistema			1,302.00 €	
	1	Pruebas de accesibilidad				
	1	Tester	9	62.00 €	558.00 €	0
	2	Pruebas de usabilidad				
	1	Tester	12	62.00 €	744.00 €	0
		TOTAL			48,042.25 €	

Tabla 23 – Presupuesto de cliente final del proyecto

Y para ver las variaciones en los costes de las diferentes partidas se incluye la siguiente tabla, que será el presupuesto simplificado.

Dato	Precio	Variación
Construcción del Sistema	34,836.25 €	+2,022.75 € (+6%)
Documentación	11,904.00 €	+8,184.00 € (+220%)
Pruebas del sistema	1,302.00 €	0
TOTAL	48,042.25 €	+10,206.75 € (+27%)

Tabla 24 – Resumen final de las partidas del proyecto

El alargue del proyecto ha supuesto un sobrecoste de 10.206,75 €, lo cual supone un 27% más de lo inicialmente presupuestado.

Capítulo 12. Referencias Bibliográficas

- [1] I. THE RADICATI GROUP, "Email Statistics Report," Londres, 2021.
- [2] Wanbil W. Lee, Wolfgang Zankl, and Henry Chang, "An Ethical Approach to Data Privacy Protection," *ISACA Journal*, vol. 6, Dec. 2016.
- [3] Arsys, "Diferencias entre POP3 e IMAP." <https://www.arsys.es/soporte/correo/diferencias-entre-pop3-e-imap> (accessed Dec. 20, 2021).
- [4] D. Crocker, "RFC #822. Standard for ARPA Internet Text Messages.," Newark, DE, Aug. 1982. Accessed: Dec. 20, 2021. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc822#section-3.2>
- [5] javinpaul, "Essential Java Libraries and APIs Every Programmer Should Learn in 2021," *Medium.com*. <https://medium.com/javarevisited/20-essential-java-libraries-and-apis-every-programmer-should-learn-5ccd41812fc7> (accessed Dec. 20, 2021).
- [6] VMWare Inc., "Spring." <https://spring.io/projects/spring-boot> (accessed Dec. 21, 2021).
- [7] Miguel Ángel Sánchez, "Inversión de Control - Principio de Hollywood: 'Don't call us, we'll call you,'" *medium.com*, Sep. 26, 2017.
- [8] Martin Fowler, "Inversion of Control Containers and the Dependency Injection pattern," *martinfowler.com*, Jan. 23, 2004.
- [9] Apache Maven Project, "What is Maven?," *maven.apache.org*, Dec. 20, 2021. <https://maven.apache.org/what-is-maven.html> (accessed Dec. 21, 2021).
- [10] Google, "Angular Features." <https://angular.io/features> (accessed Dec. 22, 2021).
- [11] M. Rehkopf, "Historias de usuario con ejemplos y plantilla," *Atlassian Agile Coach*. atlassian.com/es/agile/project-management/user-stories (accessed Jul. 13, 2021).
- [12] W3C, "Web Accessibility Tutorials." <https://www.w3.org/WAI/tutorials> (accessed Oct. 26, 2021).
- [13] wc testing, "La famosa Pirámide de Cohn y la dura realidad," Sep. 28, 2018. <https://medium.com/@wc.testing.qa/la-famosa-pir%C3%A1mide-de-cohn-y-la-dura-realidad-e1250dfbe5f3> (accessed Aug. 09, 2021).

- [14] Ankunda R. Kiremire, "THE APPLICATION OF THE PARETO PRINCIPLE IN SOFTWARE ENGINEERING," Oct. 2011.
- [15] IBM, "Formato JSON (JavaScript Object Notation)." <https://www.ibm.com/docs/es/baw/20.x?topic=formats-javascript-object-notation-json-format> (accessed Nov. 02, 2021).

Capítulo 13. Apéndices

Este capítulo contiene el glosario, el resumen de los contenidos que se adjuntan aparte, tales como el código fuente de la aplicación y el código de las diferentes pruebas. También se incluye el índice alfabético de términos.

13.1 Glosario y Diccionario de Datos

A continuación, se incluyen varios términos importantes que pueden ser útiles para comprender el funcionamiento de la aplicación.

- **Back-end:** En un sistema diseñado por capas es la parte que se encarga del procesamiento de datos.
- **Cookie:** Información pequeña que se almacena en el navegador de manera que puede ser leída por las aplicaciones para consultar la actividad previa del usuario.
- **CSV (archivo):** *Comma Separated Values*. Tipo de archivo cuyo tipo MIME es *text/csv* y que contiene una serie de valores separados por el carácter coma (,) o punto y coma (;) y sus filas por medio de saltos de línea (\n).
- **Front-end:** En un sistema diseñado por capas es la parte que se encarga de la interacción con los usuarios.
- **Historia de Usuario:** Artefacto de Scrum que permite la representación de un requisito en una o dos oraciones. Deben cumplir una serie de criterios como ser independientes, negociables o estimables.
- **Host:** De manera genérica un *host* (o anfitrión) es una máquina que ofrece servicios de múltiples tipos, p.ej. transferencia de archivos, servidores web. En el contexto de este documento se usa para referirse a la máquina donde el proveedor de correo del usuario almacena los correos.
- **IDE:** *Integrated Development Environment*. Es una aplicación informática que proporciona herramientas para facilitar el desarrollo de software. Algunos ejemplos son Eclipse, IntelliJ o WebStorm.
- **Interfaz Gráfica de Usuario:** También abreviado como IGU o más comúnmente GUI, del inglés *Graphic User Interface*. Es el conjunto de elementos tales como botones, imágenes o selectores que permite al usuario interactuar con una aplicación de manera usable y amigable.
- **PDF (archivo):** *Portable Document Format*. Tipo de archivo que permite el almacenamiento de información de múltiples tipos, tanto imágenes como texto. Es muy popular y su tipo MIME es *application/pdf*.
- **Remitente:** En una comunicación, y en este contexto en un correo electrónico, es la persona que envía el mensaje.
- **Requisito:** Se conoce como requisito a cada una de las capacidades que debe poseer un software o un sistema. El conjunto de ellos debe especificar qué debe hacer un sistema.
- **Tipo MIME:** La nomenclatura MIME de un conjunto de datos permite identificar el tipo de datos que son. Por ejemplo, *text/csv* identifica archivos de texto y en formato CSV.

13.2 Contenido Entregado en el Archivo adjunto

Esta documentación se entrega en un archivo que contiene varias cosas más. En las próximas líneas describiremos su estructura y qué contiene cada una de sus subcarpetas. Además de lo comentado, se incluye el código fuente, los archivos compilados, la documentación técnica y en definitiva, todo aquello necesario para evaluar el proyecto.

13.2.1 Contenidos

Por medio de una serie de tablas explicaremos el contenido de cada una de las carpetas que se entregan.

./Plataforma para la generación automática de actas a partir de la comunicación por correo electrónico	
Directorio	Contenido
<i>./documentacion</i>	Contiene este documento y los archivos de imagen utilizados para su creación.
<i>./documentacion/img</i>	Incluye las imágenes utilizadas en el documento.
<i>./documentacion/uml</i>	Aquí están los archivos <i>.drawio</i> de <i>diagrams.net</i> , que se han usado para realizar los diagramas <i>uml</i> y otras infografías.
<i>./UNA</i>	Contiene el resto de directorios del proyecto y la documentación técnica de cada uno. Se explican en la siguiente tabla.

Por motivos técnicos y organizativos, la carpeta *UNA* contenida en el directorio raíz contiene los archivos *build.bat* y *run.bat*, que se explican en el Manual de Ejecución. También la descripción de los contenedores a desplegar (*docker-compose.yml*). La siguiente tabla indica la estructura del resto de directorios contenidos.

./UNA	
Directorio	Contenido
<i>./UNA-API</i>	Aquí se incluyen los archivos relativos al <i>back-end</i> , el <i>Dockerfile</i> que construye el contenedor, el <i>pom.xml</i> que establece las dependencias y otros archivos de configuración.
<i>./UNA-API/docs</i>	Se incluye aquí el <i>Javadoc</i> de la aplicación. Puede verse en un navegador clicando sobre el archivo <i>"index.html"</i> .
<i>./UNA-API/src</i>	Contiene el código fuente de la aplicación. Su contenido se describe más adelante en el punto 13.2.2.
<i>./UNA-API/target</i>	Son los archivos compilados de la aplicación.
<i>./UNA-FrontEnd</i>	Contiene los archivos relativos al <i>front-end</i> , su <i>Dockerfile</i> correspondiente, la

	configuración del servidor (nginx.conf), el package.json con las dependencias y otros archivos de configuración.
<code>./UNA-FrontEnd/cert</code>	Contiene los certificados necesarios para que la aplicación use el protocolo HTTPS.
<code>./UNA-FrontEnd/dist</code>	Contiene los archivos listos para su compilación JIT (<i>Just In Time</i>).
<code>./UNA-FrontEnd/docs</code>	Contiene la documentación técnica del <i>front-end</i> . Puede verse en un navegador clicando sobre el archivo "index.html".
<code>./UNA-FrontEnd/e2e</code>	Son archivos necesarios para la ejecución de los tests unitarios.
<code>./UNA-FrontEnd/node_modules</code>	Este directorio contiene los paquetes dependientes, necesarios en la construcción de la aplicación.
<code>./UNA-FrontEnd/src</code>	Contiene el código fuente de la aplicación. Su contenido se describe más adelante en el punto 13.2.2.
<code>./UNA-PruebasFuncionales</code>	Contiene el código fuente de las pruebas funcionales.

13.2.2 Código Ejecutable e Instalación

Esta información se amplía en el Manual de Instalación y en el Manual de Ejecución. Puede resumirse en la necesidad de instalar Docker (y quizás Docker Compose). Una vez cumplido con este requisito puede usarse el archivo "run.bat" que se encuentra en la carpeta "/UNA". El código ya se proporciona compilado, por lo que no es necesario ningún paso más. Para acceder a la aplicación usar la URL <https://localhost:4200>.

./UNA-FrontEnd/src	
Directorio	Contenido
<code>./app</code>	Contiene el código de la aplicación.
<code>./app/beans</code>	Incluye los beans que se usan para el traspaso de datos.
<code>./app/common</code>	Incluye código común a todas las clases, como las variables globales.
<code>./app/componentes</code>	Contiene el código relativo a los componentes de la aplicación.
<code>./app/servicios</code>	Es el código relativo a los servicios, que realizan las llamadas al <i>back-end</i> .
<code>./assets</code>	Incluye los iconos, los archivos de i18n y los auxiliares de la aplicación.
<code>./environments</code>	Contiene las variables de entorno de la aplicación.

./UNA-API/src	
Directorio	Contenido

<i>./main/java/es/uniovi</i>	Contiene el código de la aplicación y sus paquetes.
<i>./main/resources</i>	Contiene recursos de la aplicación tales como los certificados o los archivos de i18n.
<i>./test/java/es/uniovi</i>	Contiene el código fuente de los tests de desarrollo e integración.
<i>./test/java/es/uniovi/common_logic</i>	Aquí van archivos comunes a todos los tests.
<i>./test/java/es/uniovi/tests</i>	Contiene las pruebas de la aplicación. Cada una de sus subcarpetas se corresponde con una del código.
<i>./test/resources</i>	Incluye los recursos para los tests, como los parámetros de la aplicación.

13.3 Índice Alfabético

Para facilitar la localización de párrafos referentes a determinados temas se incluye este índice alfabético, que contiene ciertos términos importantes y las páginas en la que se mencionan.

A

acta, 3, 8, 12, 14, 17, 18, 41, 42, 45, 46, 51, 53, 54, 55, 56, 58, 60, 61, 63, 64, 68, 69, 73, 74, 75, 77, 81, 82, 83, 90, 95, 102, 111, 128, 130, 135, 137, 138, 141, 142, 143, 145, 146, 147, 151, 152, 154, 155, 156, 157, 162, 174, 177, 183, 192, 194, 196, 201, 202, 204, 205, 206, 211, 216, 217, 220, 223, 224, 225, 226, 231, 235
Angular, 7, 21, 28, 69, 112, 123, 139, 164, 166, 168, 172, 230

B

base de datos, 12, 21, 22, 29, 69, 159, 199, 202, 207, 208, 209, 210, 214, 216

C

cookies, 8, 11, 12, 51, 65, 74, 98, 136, 137, 140, 155, 177, 184, 227

D

Docker, 7, 9, 11, 29, 30, 34, 56, 65, 112, 156, 159, 168, 175

G

Gmail, 12, 17, 19, 20, 24, 43, 52, 62, 74, 133, 140, 143, 150, 154, 155, 160, 170, 181

H

hosts, 50, 61, 72, 75, 172, 230, 235

I

idioma, 8, 12, 49, 67, 68, 73, 80, 81, 92, 102, 109, 124, 131, 139, 154, 162, 171, 181, 182, 217, 220, 221, 229

interfaz, 9, 11, 18, 25, 28, 41, 42, 43, 65, 72, 73, 74, 81, 84, 85, 87, 88, 89, 90, 92, 93, 94, 98, 100, 101, 102, 104, 106, 108, 111, 118, 124, 128, 129, 131, 136, 159, 162, 172, 177, 192, 194, 196, 198, 210, 214, 215, 220, 223, 230, 231, 232, 233

J

JavaMail, 7, 11, 23, 25, 26, 28, 111, 170, 172

N

Nginx, 56, 65, 112, 159

O

Outlook, 17, 19, 20, 24, 43, 143, 150, 154

P

pruebas de accesibilidad, 106, 159, 176
pruebas de rendimiento, 158, 175, 191
pruebas de usabilidad, 106, 139, 159, 160, 174, 198
pruebas funcionales, 151, 175
pruebas unitarias, 104, 139, 174

R

requisitos, 15, 31, 41, 42, 51, 56, 57, 59, 61, 65, 69, 84, 159, 202

S

Scrum, 31, 41, 42, 57, 84, 233, 236, 238
SpringBoot, 7, 21, 26, 27, 69, 112, 165, 171, 230
sprint, 13, 33, 34, 42, 43, 45, 49, 57, 59, 61, 106, 150, 151, 154, 170, 171, 175, 198, 236, 238, 239, 240