



Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo



Escuela de
Ingeniería
Informática
Universidad de Oviedo



Proyecto de
Desarrollo

GRADO EN INGENIERÍA INFORMÁTICA DEL SOFTWARE

TRABAJO DE FIN DE GRADO

Análisis, diseño e implementación de herramienta Cost Deployment para la identificación de costes y pérdidas en procesos de las PYMES

AUTOR

Javier Martínez Álvarez

DIRECTOR

Rodolfo de Benito Arango

Julio 2020

Copyright (C) 2019 **JOSÉ MANUEL REDONDO LÓPEZ**. [1]

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

Agradecimientos

Este proyecto fin de grado ha sido el final de un camino que empezó hace cuatro años cuando decidí que quería estudiar en la universidad. Durante este tiempo, han pasado muchísimas cosas, buenas y malas, que he tenido que ir superando pero que me ha hecho evolucionar en muchos aspectos tanto personales como laborables. Me alegro de haber tomado la decisión de haber escogido algo relacionado con el software.

Puedo decir que me siento orgulloso de haber finalizado el último objetivo de la carrera, el trabajo fin de grado. Ha sido un camino largo, duro y que me ha permitido ampliar mucho mis conocimientos en materias que realmente me gustan.

Me gustaría dar las gracias a todos esos amigos que he conocido estos cuatro años. Puedo decir que he tenido la mayor suerte del mundo habiéndoles conocido. Estos años han pasado volando y tengo claro que, en gran parte, ha sido gracias a ellos.

Por otra parte, me gustaría dar las gracias a mi tutor del trabajo, Rodolfo, ya que me ha aportado la suficiente confianza y ayuda como para poder completar el objetivo que nos habíamos marcado en un principio: finalizar un sistema que pueda servir como demo para un proyecto real. Asimismo, remarcar la ayuda que me dieron todos los empleados de Seresco.

Por último, y no menos importante, me gustaría dar las gracias a mis padres, a mi hermana y a mis abuelos. Sin todos ellos, sé que no habría terminado, y quizás, ni haber empezado.

Este trabajo me gustaría dedicárselo en especial a mi abuelo. Siempre se ha preocupado muchísimo por mí, se ha interesado por mis notas y me ha ayudado económicamente los primeros años. Durante estos años, a pesar de la difícil situación que se encontraba, me ha sabido dar la fuerza para continuar y seguir adelante. Muchas gracias por todo, tras cuatro años aquí me estás viendo acabar la carrera.

Palabras claves

Cost Deployment, ahorros, pérdidas, beneficios, coste, manufactura, empresas, PYMEs, producción, energía, gestión, visión, procesos productivos, gestión empresarial.

Resumen

El proyecto tiene como objetivo alcanzar un sistema que permita maximizar los beneficios reduciendo las posibles pérdidas producidas por la realización de una actividad económica. Sus clientes potenciales son empresas de reducida dimensión – PYMEs.

Para alcanzarlo, se hará uso de técnicas de *Cost Deployment* que permitan estimar y aproximar de forma cuantitativa las diferentes líneas de producción. El usuario introducirá en el sistema posibles situaciones con las que comparar para escoger la que mejor se adapte. El resultado de la aplicación podrá ser entrada para la creación de un proyecto de reducción de costes en una empresa basado en los datos reflejados por el sistema.

El sistema estará diseñado para estar disponible a través de la nube de MS Azure. Se definirá una arquitectura basada en microservicios pensada para ser totalmente compatible con un *cluster* de *Kubernetes*.

Keywords

Cost Deployment, savings, losses, profits, cost, manufacture, company, SME, production, energy, management, vision, productive process, business management.

Abstract

The project aims to achieve an IT system that allows to maximise profit reducing all possible losses caused by the accomplishment of any economic activity. Its potential clients would be small companies – SMEs.

To achieve this, Cost Deployment techniques will be used to estimate and approximate all the different production lines in a quantitative way. Possible situations are inputted by the user so that they can compare them and choose the best one. The output of the system could be the input for a cost reduction project based on the data reflected by the system.

The system will be designed to run on MS Azure cloud service. A microservices architecture will be fully prepared for use in a Kubernetes cluster.

Índice de contenido

Capítulo 1 Planificación del Sistema de Información	23
PSI 1: Inicio del Plan de Sistemas de Información.....	25
PSI 1.1: Análisis de la Necesidad del PSI	26
PSI 1.2: Identificación del Alcance del PSI	28
PSI 1.3: Determinación de Responsables	28
PSI 2: Definición y Organización del PSI	30
PSI 2.1: Especificación del Ámbito y Alcance.....	34
PSI 2.2: Organización del PSI	35
PSI 2.3: Definición del Plan de Trabajo	36
PSI 3.1: Selección y Análisis de Antecedentes.....	37
PSI 7: Definición de la Arquitectura Tecnológica	38
PSI 7.1: Identificación de las Necesidades de Infraestructura Tecnológica	38
PSI 7.2: Selección de la Arquitectura Tecnológica.....	39
Capítulo 2 Estudio de Viabilidad del Sistema	41
EVS 4, 5 y 6: Estudio, Valoración de Alternativas de Solución y Selección de Alternativa Final.....	43
Descripción de las alternativas	43
Selección de la solución elegida	47
Capítulo 3 Análisis del Sistema de Información	49
ASI 1: Definición del Sistema	51
Determinación del Alcance del Sistema	51
ASI 2: Establecimiento de Requisitos	53
Obtención de los Requisitos del Sistema	53
Identificación de Actores del Sistema	56
Especificación de Casos de Uso	57
ASI 3: Identificación de Subsistemas de Análisis.....	59
Descripción de los Subsistemas.....	59
Descripción de los Interfaces entre Subsistemas	62
ASI 4: Análisis de los Casos de Uso.....	64
Inicio de sesión	64

Gestión de Usuarios.....	65
Gestión de Visiones	68
Gestión de Máquinas.....	71
Gestión de FactoryModel	74
Costes	77
Energía	79
Producción	81
Calidad	83
ASI 5: Análisis de Clases.....	85
Descripción del Modelo de Dominio	85
Diagrama de Clases.....	86
Descripción de las Clases	87
ASI 8: Definición de Interfaces de Usuario	95
Descripción de la Interfaz	95
Definición de los componentes del sistema	103
Diagrama de Navegabilidad.....	105
ASI 10: Especificación del Plan de Pruebas	106
Pruebas Unitarias.....	106
Pruebas de Comportamiento	114
Pruebas de Usabilidad y Accesibilidad	123
Pruebas de Carga y Rendimiento	124
Pruebas de Código	125
Capítulo 4 Diseño del Sistema de Información	127
DSI 4: Diseño de Clases	129
Clases comunes del sistema	129
Subsistema de identificación	134
Subsistema de visiones.....	137
Subsistema de máquinas	138
Subsistema de FactoryModels.....	142
Subsistemas de componentes de FactoryModel	145
DSI 5: Diseño de la Arquitectura de Módulos del Sistema	150
Diagramas de Paquetes	150

Diagrama de Arquitectura Modular	152
Diagramas de Despliegue y Componentes.....	152
DSI 6: Diseño Físico de Datos	154
Descripción del SGBD usado.....	154
Integración del SGBD en nuestro sistema	154
Diagrama E-R	155
DSI 10: Especificación Técnica del Plan de Pruebas.....	163
Pruebas Unitarias.....	163
Pruebas de Comportamiento	185
Pruebas de Usabilidad y Accesibilidad	193
Capítulo 5 Construcción Del Sistema de Información	197
CSI 1: Preparación del Entorno de Generación y Construcción	199
Estándares y normas seguidos	199
Lenguajes de programación	199
Herramientas y programas usados para el desarrollo	201
CSI 2: Generación del Código de los Componentes y Procedimientos.....	205
Dependencias entre módulos.....	205
CSI 3: Ejecución de las Pruebas Unitarias.....	207
CSI 4: Ejecución de las Pruebas de Integración.....	208
CSI 5: Ejecución de las Pruebas del Sistema.....	216
Pruebas de carga y rendimiento.....	216
Prueba de Usabilidad y accesibilidad	217
Prueba de Código	231
CSI 6: Elaboración de los Manuales de Usuario	232
Manual de Instalación-Ejecución.....	232
Manual de Usuario	232
CSI 8: Construcción de los Componentes y Procedimientos de Migración y Carga Inicial de Datos	233
Capítulo 6 Implantación y Aceptación del Sistema	235
IAS 1: Establecimiento del Plan de Implantación.....	237
IAS 4: Carga de Datos al Entorno de Operación.....	238
IAS 5: Pruebas de Implantación del Sistema	239

IAS 7: Preparación del Mantenimiento del Sistema	241
Capítulo 7 Apéndices.....	243
Problemas Encontrados Durante el Desarrollo.....	245
Comunicación entre microservicios	245
Clonación de entidades	245
Ampliaciones	247
Nuevos componentes de FactoryModel	247
Vista detalle de una visión y exportación de los datos finales.....	247
Soporte para <i>ActiveDirectory</i>	248
Separación en módulos del Cliente Web	248
Nueva implementación de Tablas de Hojas de Calculo.....	248
Mejora en la clonación de Entidades	248
<i>Dashboard</i>	248
Conclusiones.....	249
Contenido entregado en los anexos	250
Referencias Bibliográficas	252
GNU Free Documentation License	254
ADDENDUM: How to use this License for your documents.....	261
Anexos.....	263
Anexo I: Presupuesto	265
Anexo II: Requisitos funcionales	267
Inicio de sesión	267
Gestión de usuarios	267
Gestión de visiones.....	269
Gestión de maquinas	272
Gestión de FactoryModels.....	274
Gestión de los componentes de un FactoryModels.....	277
Componente de Costes	278
Componente de Energía	281
Componente de Producción.....	283
Componente de Calidad	286

Índice de Figuras

FIG 1. FASES DE TÉCNICAS DE COST-DEPLOYMENT [3]	27
FIG 2. PLANIFICACIÓN: TAREAS DE INICIO DE SESIÓN	30
FIG 3. PLANIFICACIÓN: ANÁLISIS DEL SISTEMA DE INFORMACIÓN	31
FIG 4. PLANIFICACIÓN: DISEÑO DEL SISTEMA DE INFORMACIÓN	31
FIG 5. PLANIFICACIÓN: CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN	32
FIG 6. PLANIFICACIÓN: IMPLANTACIÓN Y ACEPTACIÓN DEL SISTEMA	32
FIG 7. PLANIFICACIÓN INICIAL COMPLETA	33
FIG 8. EJEMPLO DE ARQUITECTURA USADA EN UN SISTEMA QUE USA AKS [7]	39
FIG 9. LOGO DE .NET FRAMEWORK	43
FIG 10. LOGO DE .NET CORE	44
FIG 11. LOGO DE BLAZOR	45
FIG 12. LOGO DE ANGULAR	46
FIG 13. LOGO DE VUE.JS	46
FIG 14. LOGO DE REACT	47
FIG 15. DIAGRAMA DE CLASES DEL DOMINIO	51
FIG 16. DIAGRAMA DE ACTORES Y SUS RELACIONES	57
FIG 17. DIAGRAMA DE CASOS DE USOS	58
FIG 18. DIAGRAMA DE QUEUE EVENT [9]	62
FIG 19. DIAGRAMA DE API REST [10]	63
FIG 20. DIAGRAMA DE MVVM [11]	63
FIG 21. DIAGRAMA DE CASOS DE USO: INICIO DE SESIÓN	64
FIG 22. DIAGRAMA DE CASOS DE USO: GESTIÓN DE USUARIOS	65
FIG 23. DIAGRAMA DE CASOS DE USO: GESTIÓN DE VISIONES	68
FIG 30. ANÁLISIS DE CLASES: DIAGRAMA DE CLASES	86
FIG 31. PROTOTIPO: INICIO DE SESIÓN	95
FIG 32. PROTOTIPO: LISTADO DE USUARIOS	96
FIG 33. PROTOTIPO: DIALOGO DE CREACIÓN	97
FIG 34. PROTOTIPO: DIALOGO DE CLONACIÓN	98
FIG 35. PROTOTIPO: LISTADO DE VISIONES	99
FIG 36. PROTOTIPO: VISTA DETALLES VISIÓN	100
FIG 37. PROTOTIPO: LISTADO COMPONENTES	100
FIG 38. PROTOTIPO: COMPONENTE COSTES	101
FIG 39. PROTOTIPO: ENERGÍA	102
FIG 40. PROTOTIPO: PRODUCCIÓN	102
FIG 41. PROTOTIPO: CALIDAD	103
FIG 42. DIAGRAMA DE NAVEGABILIDAD	105
FIG 43. DIAGRAMA DE CLASES COMUNES (ENTIDADES Y REPOSITORIOS)	129
FIG 44. DIAGRAMA DE CLASES COMUNES (DTO)	133
FIG 45. DISEÑO DE CLASES: SUBSISTEMA DE IDENTIFICACIÓN	135
FIG 46. DISEÑO DE CLASES: SUBSISTEMA DE VISIONES	137
FIG 47. DISEÑO DE CLASES: SUBSISTEMA DE MÁQUINAS (I)	139
FIG 48. DISEÑO DE CLASES: SUBSISTEMA DE MÁQUINAS (II)	140
FIG 49. DISEÑO DE CLASES: SUBSISTEMA DE FACTORYMODELS (I)	142
FIG 50. DISEÑO DE CLASES: SUBSISTEMA DE FACTORYMODELS (II)	144
FIG 51. DISEÑO DE CLASES: SUBSISTEMA DE COMPONENTES DE FACTORYMODEL (I)	145
FIG 52. DISEÑO DE CLASES: SUBSISTEMA DE COMPONENTES DE FACTORYMODEL (II)	147

FIG 53.	DISEÑO DE CLASES: SUBSISTEMA DE COMPONENTES DE FACTORYMODEL (III).....	148
FIG 54.	ARQUITECTURA BASADA EN EL PATRÓN CQRS [14]	150
FIG 55.	DIAGRAMA DE PAQUETES DE UN MÓDULO	150
FIG 56.	DIAGRAMA DE ARQUITECTURA CLOUD	152
FIG 57.	DIAGRAMA DE DESPLIEGUE	153
FIG 58.	DIAGRAMA E-R: MÓDULO DE IDENTIDAD.....	155
FIG 59.	DIAGRAMA E-R: MÓDULO DE VISION	156
FIG 60.	DIAGRAMA E-R: MÓDULO DE MÁQUINAS	157
FIG 61.	DIAGRAMA E-R: MÓDULO DE FACTORYMODELS.....	158
FIG 62.	DIAGRAMA E-R: MÓDULO DE COSTES	159
FIG 63.	DIAGRAMA E-R: MÓDULO DE ENERGÍA	160
FIG 64.	DIAGRAMA E-R: MÓDULO DE PRODUCCIÓN	161
FIG 65.	DIAGRAMA E-R: MÓDULO DE CALIDAD	162
FIG 66.	LOGO DE VISUASTUDIO.....	201
FIG 67.	LOGO: VISUAL STUDIO CODE	201
FIG 68.	LOGO: MICROSOFT SQL SERVER MANAGEMENT STUDIO.....	202
FIG 69.	LOGO: DOCKER.....	202
FIG 70.	LOGO: KUBERNETES	203
FIG 71.	LOGO: AZURE CLI.....	203
FIG 72.	LOGO: GIT.....	203
FIG 73.	DIAGRAMA DE DEPENDENCIAS ENTRE LOS DIFERENTES MÓDULOS	206
FIG 74.	RESULTADOS DE LAS PRUEBAS UNITARIAS EJECUTADAS.....	207
FIG 75.	DIAGRAMA DE FLUJO DEL PRIMER DISEÑO PARA LA CLONACIÓN	246

Índice de Tablas

TABLA 1.	EQUIPOS DE TRABAJO Y USUARIOS	35
TABLA 2.	RESUMEN POR CAPÍTULOS DE LA PLANIFICACIÓN	36
TABLA 3.	ESTADÍSTICAS DE HORAS A TRABAJAR POR PERSONAL	37
TABLA 4.	RESUMEN DE REQUISITOS DEL SISTEMA FUNCIONES	54
TABLA 5.	CASO DE USO: INICIAR SESIÓN	65
TABLA 6.	CASO DE USO: DAR DE ALTA A USUARIOS	66
TABLA 7.	CASO DE USO: CONSULTA TODOS LOS USUARIOS.....	66
TABLA 8.	CASO DE USO: MODIFICAR USUARIO.....	67
TABLA 9.	CASO DE USO: CONSULTAR DETALLES DE UN USUARIO	67
TABLA 10.	CASO DE USO: CONSULTA VISIONES.....	69
TABLA 11.	CASO DE USO: AÑADIR VISIÓN	69
TABLA 12.	CASO DE USO: MODIFICAR VISIÓN	70
TABLA 13.	CASO DE USO: CLONAR VISIÓN.....	70
TABLA 14.	CASO DE USO: CONSULTA MÁQUINAS	72
TABLA 15.	CASO DE USO: AÑADIR MÁQUINA	72
TABLA 16.	CASO DE USO: MODIFICAR MÁQUINA.....	73
TABLA 17.	CASO DE USO: CLONAR MÁQUINA	73
TABLA 18.	CASO DE USO: CONSULTA FACTORYMODELS	75
TABLA 19.	CASO DE USO: CONSULTA COMPONENTES	75
TABLA 20.	CASO DE USO: AÑADIR FACTORYMODELS.....	76
TABLA 21.	CASO DE USO: MODIFICAR FACTORYMODELS.....	76
TABLA 22.	CASO DE USO: CLONAR FACTORYMODELS	77
TABLA 23.	CASO DE USO: AÑADIR COSTES	78
TABLA 24.	CASO DE USO: CONSULTA DE COSTES	79
TABLA 25.	CASO DE USO: AÑADIR POTENCIA	80
TABLA 26.	CASO DE USO: AÑADIR CONSUMO	80
TABLA 27.	CASO DE USO: CONSULTA DE TÉRMINOS	81
TABLA 28.	CASO DE USO: AÑADIR PRODUCTOS	82
TABLA 29.	CASO DE USO: CONSULTA DE PRODUCCIÓN	82
TABLA 30.	CASO DE USO: AÑADIR DEFECTOS	83
TABLA 31.	CASO DE USO: CONSULTA DE CALIDAD	84
TABLA 32.	DESCRIPCIÓN DE CLASES: USER	87
TABLA 33.	DESCRIPCIÓN DE CLASES: ROLE	87
TABLA 34.	DESCRIPCIÓN DE CLASES: <i>VISION</i>	88
TABLA 35.	DESCRIPCIÓN DE CLASES: <i>MACHINE</i>	88
TABLA 36.	DESCRIPCIÓN DE CLASES: <i>GROUP</i>	88
TABLA 37.	DESCRIPCIÓN DE CLASES: <i>FACTORYMODEL</i>	89
TABLA 38.	DESCRIPCIÓN DE CLASES: <i>FACTORYMODELSTATUS</i>	89
TABLA 39.	DESCRIPCIÓN DE CLASES: <i>COSTCOMPONENT</i>	90
TABLA 40.	DESCRIPCIÓN DE CLASES: <i>COST</i>	90
TABLA 41.	DESCRIPCIÓN DE CLASES: <i>FIXEDCOST</i>	90
TABLA 42.	DESCRIPCIÓN DE CLASES: <i>VARIABLECOST</i>	91
TABLA 43.	DESCRIPCIÓN DE CLASES: <i>ENERGYCOMPONENT</i>	91
TABLA 44.	DESCRIPCIÓN DE CLASES: <i>PERIOD</i>	91
TABLA 45.	DESCRIPCIÓN DE CLASES: <i>TIME</i>	92
TABLA 46.	DESCRIPCIÓN DE CLASES: <i>PRODUCTIONCOMPONENT</i>	92

TABLA 47.	DESCRIPCIÓN DE CLASES: <i>QUALITYCOMPONENT</i>	93
TABLA 48.	DESCRIPCIÓN DE CLASES: <i>PRODUCT</i>	94
TABLA 49.	DESCRIPCIÓN DE CLASES: <i>FAULT</i>	94
TABLA 50.	PRUEBAS UNITARIAS: INICIAR SESIÓN	106
TABLA 51.	PRUEBAS UNITARIAS: CONSULTA DE USUARIOS	106
TABLA 52.	PRUEBAS UNITARIAS: DAR DE ALTA A USUARIOS.....	106
TABLA 53.	PRUEBAS UNITARIAS: MODIFICAR USUARIO	107
TABLA 54.	PRUEBAS UNITARIAS: CONSULTAR DETALLES DE UN USUARIO	107
TABLA 55.	PRUEBAS UNITARIAS: CONSULTA DE VISIONES	107
TABLA 56.	PRUEBAS UNITARIAS: AÑADIR VISIÓN	107
TABLA 57.	PRUEBAS UNITARIAS: MODIFICAR VISIÓN	107
TABLA 58.	PRUEBAS UNITARIAS: CLONAR VISIÓN	107
TABLA 59.	PRUEBAS UNITARIAS: CONSULTA DE MÁQUINAS.....	108
TABLA 60.	PRUEBAS UNITARIAS: AÑADIR MÁQUINA	108
TABLA 61.	PRUEBAS UNITARIAS: MODIFICAR MÁQUINAS.....	108
TABLA 62.	PRUEBAS UNITARIAS: CLONAR MÁQUINAS	108
TABLA 63.	PRUEBAS UNITARIAS: CONSULTA DE FACTORYMODELS	108
TABLA 64.	PRUEBAS UNITARIAS: CONSULTA DE COMPONENTES.....	109
TABLA 65.	PRUEBAS UNITARIAS: AÑADIR FACTORYMODELS	109
TABLA 66.	PRUEBAS UNITARIAS: MODIFICAR FACTORYMODELS	109
TABLA 67.	PRUEBAS UNITARIAS: CLONAR FACTORYMODELS.....	109
TABLA 68.	PRUEBAS UNITARIAS: VISTA DE DETALLES Y EDICIÓN DE FM COSTES.....	110
TABLA 69.	PRUEBAS UNITARIAS: CREACIÓN DE FM COSTES	110
TABLA 70.	PRUEBAS UNITARIAS: CLONACIÓN DE FM COSTES.....	110
TABLA 71.	PRUEBAS UNITARIAS: VISTA DETALLES Y EDICIÓN DE FM ENERGÍA.....	110
TABLA 72.	PRUEBAS UNITARIAS: CREACIÓN DE FM CALIDAD.....	111
TABLA 73.	PRUEBAS UNITARIAS: CLONACIÓN DE FM ENERGÍAS.....	111
TABLA 74.	PRUEBAS UNITARIAS: VISTA DE DETALLES Y EDICIÓN DE FM PRODUCCIÓN.....	111
TABLA 75.	PRUEBAS UNITARIAS: CREACIÓN DE FM PRODUCCIÓN.....	112
TABLA 76.	PRUEBAS UNITARIAS: CLONACIÓN DE FM PRODUCCIÓN.....	112
TABLA 77.	PRUEBAS UNITARIAS: VISTA DE DETALLES Y EDICIÓN DE FM CALIDAD.....	112
TABLA 78.	PRUEBAS UNITARIAS: CREACIÓN DE FM CALIDAD.....	112
TABLA 79.	PRUEBAS UNITARIAS: CLONACIÓN DE FM CALIDAD.....	113
TABLA 80.	PRUEBAS DE COMPORTAMIENTO: INICIO DE SESIÓN	114
TABLA 81.	PRUEBAS DE COMPORTAMIENTO: DAR DE ALTA A USUARIOS	114
TABLA 82.	PRUEBAS DE COMPORTAMIENTO: CONSULTA TODOS LOS USUARIOS	115
TABLA 83.	PRUEBAS DE COMPORTAMIENTO: MODIFICAR USUARIO	115
TABLA 84.	PRUEBAS DE COMPORTAMIENTO: CONSULTA TODOS LOS USUARIOS	115
TABLA 85.	PRUEBAS DE COMPORTAMIENTO: CONSULTA DE VISIONES.....	115
TABLA 86.	PRUEBAS DE COMPORTAMIENTO: AÑADIR VISIÓN.....	116
TABLA 87.	PRUEBAS DE COMPORTAMIENTO: MODIFICAR VISIÓN.....	116
TABLA 88.	PRUEBAS DE COMPORTAMIENTO: CLONAR VISIÓN	116
TABLA 89.	PRUEBAS DE COMPORTAMIENTO: CONSULTA DE MÁQUINAS	117
TABLA 90.	PRUEBAS DE COMPORTAMIENTO: AÑADIR MÁQUINAS	117
TABLA 91.	PRUEBAS DE COMPORTAMIENTO: MODIFICAR MÁQUINA.....	117
TABLA 92.	PRUEBAS DE COMPORTAMIENTO: CLONAR VISIÓN	118
TABLA 93.	PRUEBAS DE COMPORTAMIENTO: CONSULTA DE FACTORYMODEL	118
TABLA 94.	PRUEBAS DE COMPORTAMIENTO: CONSULTA DE COMPONENTES	118
TABLA 95.	PRUEBAS DE COMPORTAMIENTO: AÑADIR FACTORYMODEL.....	119

TABLA 96.	PRUEBAS DE COMPORTAMIENTO: MODIFICAR FACTORYMODEL.....	119
TABLA 97.	PRUEBAS DE COMPORTAMIENTO: CLONAR FACTORYMODEL	120
TABLA 98.	PRUEBAS DE COMPORTAMIENTO: AÑADIR COSTES.....	120
TABLA 99.	PRUEBAS DE COMPORTAMIENTO: CONSULTA DE COSTES.....	120
TABLA 100.	PRUEBAS DE COMPORTAMIENTO: AÑADIR POTENCIA	121
TABLA 101.	PRUEBAS DE COMPORTAMIENTO: AÑADIR CONSUMO.....	121
TABLA 102.	PRUEBAS DE COMPORTAMIENTO: CONSULTA DE ENERGÍA.....	121
TABLA 103.	PRUEBAS DE COMPORTAMIENTO: AÑADIR PRODUCTOS	122
TABLA 104.	PRUEBAS DE COMPORTAMIENTO: CONSULTA DE PRODUCCIÓN	122
TABLA 105.	PRUEBAS DE COMPORTAMIENTO: AÑADIR DEFECTOS.....	122
TABLA 106.	PRUEBAS DE COMPORTAMIENTO: CONSULTA DE CALIDAD	122
TABLA 107.	PRUEBAS DE CÓDIGO: COMPLEJIDAD CICLOMÁTICA.....	125
TABLA 108.	PRUEBAS DE CÓDIGO: PROFUNDIDAD DE HERENCIA	125
TABLA 109.	PRUEBAS DE CÓDIGO: LÍNEAS DE CÓDIGO.....	125
TABLA 110.	PRUEBAS DE CÓDIGO: MANTENIBILIDAD.....	125
TABLA 111.	DISEÑO DE CLASES: INTERFAZ ICLONE	130
TABLA 112.	DISEÑO DE CLASES: INTERFAZ IAUDITABLEENTITY.....	130
TABLA 113.	DISEÑO DE CLASES: INTERFAZ IENTITY.....	130
TABLA 114.	DISEÑO DE CLASES: CLASE ABSTRACTA ENTITYBASE.....	131
TABLA 115.	DISEÑO DE CLASES: CLASE ABSTRACTA ENTITY.....	131
TABLA 116.	DISEÑO DE CLASES: INTERFAZ IREPOSITORY	132
TABLA 117.	DISEÑO DE CLASES: CLASE ABSTRACTA REPOSITORYBASE.....	133
TABLA 118.	DISEÑO DE CLASES: CLASE ABSTRACTA AUDITABLECONTEXT.....	133
TABLA 119.	DISEÑO DE CLASES: INTERFAZ IAUDITABLEDTO	134
TABLA 120.	DISEÑO DE CLASES: INTERFAZ IDTO	134
TABLA 121.	DISEÑO DE CLASES: CLASE ABSTRACTA DTOBASE	134
TABLA 122.	DISEÑO DE CLASES: CLASE USERCONTROLLER	135
TABLA 123.	DISEÑO DE CLASES: CLASE SIGNINCONTROLLER	135
TABLA 124.	DISEÑO DE CLASES: CLASE ROLECONTROLLER	136
TABLA 125.	DISEÑO DE CLASES: CLASE TOKENGENERATOR.....	136
TABLA 126.	DISEÑO DE CLASES: CLASE USERHANDLER	136
TABLA 127.	DISEÑO DE CLASES: CLASE ROLEHANDLER	136
TABLA 128.	DISEÑO DE CLASES: CLASE VISIONCONTROLLER	137
TABLA 129.	DISEÑO DE CLASES: CLASE MACHINEQUERYHANDLER	137
TABLA 130.	DISEÑO DE CLASES: CLASE VISIONQUERYHANDLER	138
TABLA 131.	DISEÑO DE CLASES: CLASE VISIONHANDLER	138
TABLA 132.	DISEÑO DE CLASES: CLASE CLONEVISIONHANDLER	138
TABLA 133.	DISEÑO DE CLASES: CLASE MACHINECONTROLLER.....	139
TABLA 134.	DISEÑO DE CLASES: CLASE GROUPHANDLER.....	139
TABLA 135.	DISEÑO DE CLASES: CLASE MACHINEHANDLER.....	140
TABLA 136.	DISEÑO DE CLASES: CLASE CLONEMACHINEHANDLER.....	140
TABLA 137.	DISEÑO DE CLASES: CLASE GROUPCONTROLLER.....	141
TABLA 138.	DISEÑO DE CLASES: CLASE MACHINEQUERYHANDLER	141
TABLA 139.	DISEÑO DE CLASES: CLASE VISIONQUERYHANDLER	141
TABLA 140.	DISEÑO DE CLASES: CLASE FACTORYMODELQUERYHANDLER	141
TABLA 141.	DISEÑO DE CLASES: CLASE VISIONHANDLER	142
TABLA 142.	DISEÑO DE CLASES: CLASE FACTORYMODELCONTROLLER.....	143
TABLA 143.	DISEÑO DE CLASES: CLASE FACTORYMODELCOMPONENTSQUERYHANDLER.....	143
TABLA 144.	DISEÑO DE CLASES: CLASE FACTORYMODELCOMPONENTSHANDLER	143

TABLA 145.	DISEÑO DE CLASES: CLASE CLONEFACTORYMODELHANDLER.....	144
TABLA 146.	DISEÑO DE CLASES: CLASE MACHINEHANDLER.....	144
TABLA 147.	DISEÑO DE CLASES: CLASE FACTORYMODELCOMPONENTSCONTROLLER	145
TABLA 148.	DISEÑO DE CLASES: CLASE MACHINEQUERYHANDLER	145
TABLA 149.	DISEÑO DE CLASES: CLASE FACTORYMODELQUERYHANDLER	145
TABLA 150.	DISEÑO DE CLASES: CLASE ECHOCONTROLLER	146
TABLA 151.	DISEÑO DE CLASES: CLASE COSTCONTROLLER	146
TABLA 152.	DISEÑO DE CLASES: CLASE COSTCOMPONENTCONTROLLER	146
TABLA 153.	DISEÑO DE CLASES: CLASE FACTORYMODELCOMPONENTQUERYHANDLER	147
TABLA 154.	DISEÑO DE CLASES: CLASE FACTORYMODELQUERYHANDLER	147
TABLA 155.	DISEÑO DE CLASES: CLASE REFRESHCOSTHANDLER.....	147
TABLA 156.	DISEÑO DE CLASES: CLASE FACTORYMODELHANDLER	148
TABLA 157.	DISEÑO DE CLASES: CLASE FACTORYMODELCOMPONENTSHANDLER	148
TABLA 158.	DISEÑO DE CLASES: CLASE FACTORYMODELHANDLER	149
TABLA 159.	DISEÑO DE CLASES: CLASE CLONEFACTORYMODELCOMPONENTSHANDLER	149
TABLA 160.	DISEÑO PRUEBAS UNITARIAS: INICIAR SESIÓN	163
TABLA 161.	DISEÑO PRUEBAS UNITARIAS: CONSULTA DE VISIONES	164
TABLA 162.	DISEÑO PRUEBAS UNITARIAS: AÑADIR VISIÓN	164
TABLA 163.	DISEÑO PRUEBAS UNITARIAS: MODIFICAR VISIÓN	164
TABLA 164.	DISEÑO PRUEBAS UNITARIAS: CLONAR VISIÓN.....	165
TABLA 165.	DISEÑO PRUEBAS UNITARIAS: CONSULTA DE MÁQUINAS.....	165
TABLA 166.	DISEÑO PRUEBAS UNITARIAS: AÑADIR MÁQUINA.....	166
TABLA 167.	DISEÑO PRUEBAS UNITARIAS: MODIFICAR MÁQUINAS.....	166
TABLA 168.	DISEÑO PRUEBAS UNITARIAS: CLONAR MÁQUINAS	167
TABLA 169.	DISEÑO PRUEBAS UNITARIAS: CONSULTA DE FACTORYMODELS.....	167
TABLA 170.	DISEÑO PRUEBAS UNITARIAS: CONSULTA DE COMPONENTES	167
TABLA 171.	DISEÑO PRUEBAS UNITARIAS: AÑADIR FACTORYMODELS.....	168
TABLA 172.	DISEÑO PRUEBAS UNITARIAS: MODIFICAR FACTORYMODELS.....	168
TABLA 173.	DISEÑO PRUEBAS UNITARIAS: CLONAR FACTORYMODELS	169
TABLA 174.	DISEÑO PRUEBAS UNITARIAS: CONSULTA DE USUARIOS.....	169
TABLA 175.	DISEÑO PRUEBAS UNITARIAS: DAR DE ALTA A USUARIOS	169
TABLA 176.	DISEÑO PRUEBAS UNITARIAS: MODIFICAR USUARIO	169
TABLA 177.	DISEÑO PRUEBAS UNITARIAS: CONSULTAR DETALLES DE UN USUARIO.....	170
TABLA 178.	DISEÑO PRUEBAS UNITARIAS: VISTA DE DETALLES Y EDICIÓN DE FM COSTES	172
TABLA 179.	DISEÑO PRUEBAS UNITARIAS: CREACIÓN DE FM COSTES.....	173
TABLA 180.	DISEÑO PRUEBAS UNITARIAS: CLONACIÓN DE FM COSTES	174
TABLA 181.	DISEÑO PRUEBAS UNITARIAS: VISTA DETALLES Y EDICIÓN DE FM ENERGÍA	176
TABLA 182.	DISEÑO PRUEBAS UNITARIAS: CREACIÓN DE FM CALIDAD	177
TABLA 183.	DISEÑO PRUEBAS UNITARIAS: CLONACIÓN DE FM ENERGÍAS	178
TABLA 184.	DISEÑO PRUEBAS UNITARIAS: VISTA DE DETALLES Y EDICIÓN DE FM PRODUCCIÓN	180
TABLA 185.	DISEÑO PRUEBAS UNITARIAS: CREACIÓN DE FM PRODUCCIÓN	180
TABLA 186.	DISEÑO PRUEBAS UNITARIAS: CLONACIÓN DE FM PRODUCCIÓN	181
TABLA 187.	DISEÑO PRUEBAS UNITARIAS: VISTA DE DETALLES Y EDICIÓN DE FM CALIDAD	183
TABLA 188.	DISEÑO PRUEBAS UNITARIAS: CREACIÓN DE FM CALIDAD	183
TABLA 189.	DISEÑO PRUEBAS UNITARIAS: CLONACIÓN DE FM CALIDAD	184
TABLA 190.	DISEÑO PRUEBAS DE COMPORTAMIENTO: INICIO DE SESIÓN.....	185
TABLA 191.	DISEÑO PRUEBAS DE COMPORTAMIENTO: DAR DE ALTA A USUARIOS.....	185
TABLA 192.	DISEÑO PRUEBAS DE COMPORTAMIENTO: CONSULTA TODOS LOS USUARIOS	186
TABLA 193.	DISEÑO PRUEBAS DE COMPORTAMIENTO: MODIFICAR USUARIO	186

TABLA 194.	DISEÑO PRUEBAS DE COMPORTAMIENTO: CONSULTA DETALLES DE UN USUARIO.....	186
TABLA 195.	DISEÑO PRUEBAS DE COMPORTAMIENTO: CONSULTA VISIONES	186
TABLA 196.	DISEÑO PRUEBAS DE COMPORTAMIENTO: AÑADIR VISIÓN	187
TABLA 197.	DISEÑO PRUEBAS DE COMPORTAMIENTO: MODIFICAR VISIÓN	187
TABLA 198.	DISEÑO PRUEBAS DE COMPORTAMIENTO: CLONAR VISIÓN	187
TABLA 199.	DISEÑO PRUEBAS DE COMPORTAMIENTO: CONSULTA MÁQUINAS	187
TABLA 200.	DISEÑO PRUEBAS DE COMPORTAMIENTO: AÑADIR MÁQUINA	188
TABLA 201.	DISEÑO PRUEBAS DE COMPORTAMIENTO: MODIFICAR MÁQUINA	188
TABLA 202.	DISEÑO PRUEBAS DE COMPORTAMIENTO: CLONAR MÁQUINA.....	188
TABLA 203.	DISEÑO PRUEBAS DE COMPORTAMIENTO: CONSULTA FACTORYMODELS.....	188
TABLA 204.	DISEÑO PRUEBAS DE COMPORTAMIENTO: CONSULTA COMPONENTES.....	189
TABLA 205.	DISEÑO PRUEBAS DE COMPORTAMIENTO: AÑADIR FACTORYMODEL	189
TABLA 206.	DISEÑO PRUEBAS DE COMPORTAMIENTO: MODIFICAR FACTORYMODEL	189
TABLA 207.	DISEÑO PRUEBAS DE COMPORTAMIENTO: CLONAR FACTORYMODEL.....	190
TABLA 208.	DISEÑO PRUEBAS DE COMPORTAMIENTO: AÑADIR COSTES	190
TABLA 209.	DISEÑO PRUEBAS DE COMPORTAMIENTO: CONSULTA DE COSTES.....	190
TABLA 210.	DISEÑO PRUEBAS DE COMPORTAMIENTO: AÑADIR POTENCIA.....	191
TABLA 211.	DISEÑO PRUEBAS DE COMPORTAMIENTO: AÑADIR CONSUMO.....	192
TABLA 212.	DISEÑO PRUEBAS DE COMPORTAMIENTO: CONSULTA DE ENERGÍA.....	192
TABLA 213.	DISEÑO PRUEBAS DE COMPORTAMIENTO: AÑADIR PRODUCTOS	192
TABLA 214.	DISEÑO PRUEBAS DE COMPORTAMIENTO: CONSULTA DE PRODUCCIÓN	192
TABLA 215.	DISEÑO PRUEBAS DE COMPORTAMIENTO: AÑADIR DEFECTO	193
TABLA 216.	DISEÑO PRUEBAS DE COMPORTAMIENTO: CONSULTA DE CALIDAD	193
TABLA 217.	MEDIDAS DE SEGURIDAD TOMADAS EN EL CÓDIGO DE LA APLICACIÓN.....	201
TABLA 218.	RESULTADO PRUEBAS DE COMPORTAMIENTO: INICIO DE SESIÓN.....	208
TABLA 219.	RESULTADO PRUEBAS DE COMPORTAMIENTO: DAR DE ALTA A USUARIOS.....	208
TABLA 220.	RESULTADO PRUEBAS DE COMPORTAMIENTO: CONSULTA TODOS LOS USUARIOS	209
TABLA 221.	RESULTADO PRUEBAS DE COMPORTAMIENTO: MODIFICAR USUARIO	209
TABLA 222.	RESULTADO PRUEBAS DE COMPORTAMIENTO: CONSULTA DETALLES DE UN USUARIO	209
TABLA 223.	RESULTADO PRUEBAS DE COMPORTAMIENTO: CONSULTA VISIONES	209
TABLA 224.	RESULTADO PRUEBAS DE COMPORTAMIENTO: AÑADIR VISIÓN	210
TABLA 225.	RESULTADO PRUEBAS DE COMPORTAMIENTO: MODIFICAR VISIÓN	210
TABLA 226.	RESULTADO PRUEBAS DE COMPORTAMIENTO: CLONAR VISIÓN	210
TABLA 227.	RESULTADO PRUEBAS DE COMPORTAMIENTO: CONSULTA MÁQUINAS.....	211
TABLA 228.	RESULTADO PRUEBAS DE COMPORTAMIENTO: AÑADIR MÁQUINA	211
TABLA 229.	RESULTADO PRUEBAS DE COMPORTAMIENTO: MODIFICAR MÁQUINA	211
TABLA 230.	RESULTADO PRUEBAS DE COMPORTAMIENTO: CLONAR MÁQUINA.....	212
TABLA 231.	RESULTADO PRUEBAS DE COMPORTAMIENTO: CONSULTA FACTORYMODELS.....	212
TABLA 232.	RESULTADO PRUEBAS DE COMPORTAMIENTO: CONSULTA COMPONENTES.....	212
TABLA 233.	RESULTADO PRUEBAS DE COMPORTAMIENTO: AÑADIR FACTORYMODEL	213
TABLA 234.	RESULTADO PRUEBAS DE COMPORTAMIENTO: MODIFICAR FACTORYMODEL	213
TABLA 235.	RESULTADO PRUEBAS DE COMPORTAMIENTO: CLONAR FACTORYMODEL.....	213
TABLA 236.	RESULTADO PRUEBAS DE COMPORTAMIENTO: AÑADIR COSTES	213
TABLA 237.	RESULTADO PRUEBAS DE COMPORTAMIENTO: CONSULTA DE COSTES.....	214
TABLA 238.	RESULTADO PRUEBAS DE COMPORTAMIENTO: AÑADIR POTENCIA.....	214
TABLA 239.	RESULTADO PRUEBAS DE COMPORTAMIENTO: AÑADIR CONSUMO.....	214
TABLA 240.	RESULTADO PRUEBAS DE COMPORTAMIENTO: CONSULTA DE ENERGÍA.....	214
TABLA 241.	RESULTADO PRUEBAS DE COMPORTAMIENTO: AÑADIR PRODUCTOS.....	214
TABLA 242.	RESULTADO PRUEBAS DE COMPORTAMIENTO: CONSULTA DE PRODUCCIÓN	215

TABLA 243.	RESULTADO PRUEBAS DE COMPORTAMIENTO: AÑADIR DEFECTO	215
TABLA 244.	RESULTADO PRUEBAS DE COMPORTAMIENTO: CONSULTA DE CALIDAD	215
TABLA 245.	CHECKLIST DE PRUEBAS DE USABILIDAD	222
TABLA 246.	RESULTADO PRUEBAS DE CÓDIGO	231
TABLA 247.	MEDIDAS DE SEGURIDAD DEL SERVIDOR INSTALADO	239
TABLA 248.	ESTRUCTURA GENERAL DE LOS FICHEROS ANEXOS ENTREGADOS	250
TABLA 249.	FICHEROS ANEXOS DEL PROYECTO.....	251
TABLA 250.	PRESUPUESTO RESUMEN DEL CLIENTE	265
TABLA 251.	PRESUPUESTO DEL CLIENTE	266



Capítulo 1 PLANIFICACIÓN DEL SISTEMA DE INFORMACIÓN

FASE DE PLANIFICACIÓN

PSI



PSI 1: INICIO DEL PLAN DE SISTEMAS DE INFORMACIÓN

El objetivo principal que se busca alcanzar con este trabajo fin de grado es analizar, diseñar y construir un sistema software completo para la mejora continua [2] empresarial que nos permita identificar costes, perdidas y ahorros resultantes de una actividad económica, con objeto de maximizar los ahorros y minimizar los costes y pérdidas. Es importante señalar que este sistema de software para la mejora continua se centra en la identificación de los costes y ahorros sin entrar en cómo ejecutarlos o planificarlos.

Para lograr el objetivo, se diseñará un sistema que permitirá a gestores y directivos adquirir una visión de su empresa identificando que **oportunidades de mejora** hay en sus procesos productivos desde el punto de vista del coste. Normalmente, la identificación de costes y ahorros se realizará anualmente, teniendo como referencia el ejercicio anterior e interviniendo en ella múltiples usuarios, responsables de cada unidad productiva. Esta construcción agregada de la visión permitirá además contemplar diferentes escenarios de mejora, por ejemplo, ahorro de costes agresivo frente a un escenario de mejora más leve, buscando obtener la mejor visión posible. De todos los escenarios planteados, solo uno de ellos será el que se consolidará y agregará para conformar la visión global de la empresa.

El sistema proporcionará una aplicación web orientado a la mejora continua de empresas de pequeño/media tamaño. Debido al uso extendido en este ámbito de aplicaciones de hojas de cálculo, fundamentalmente MS Excel, se diseñará la aplicación teniendo en cuenta la necesidad de mantener una similitud visual con este tipo de aplicaciones.

Como prueba de concepto, el sistema se implantará en la empresa cliente, Seresco S.A. lo que proporcionará un valioso *feedback* que permita refinar y mejorar el software en futuras iteraciones.

Para lograr que el proyecto sea un éxito, será necesario que se cumplan con requisitos especificados por el Cliente en primera instancia, al tratarse de los más importantes. Son los siguientes:

1. Portabilidad
2. Configurabilidad
3. Comprensibilidad

Para completar con éxito el proyecto que se detallará en las siguientes páginas, ha sido necesario el trabajo de todos los participantes que han intervenido de forma directa o indirecta por parte de la empresa. Recalcar en especial a mi director del proyecto y mis compañeros de trabajo.



PSI 1.1: Análisis de la Necesidad del PSI

La empresa Seresco S.A. ha decidido desarrollar una aplicación web que permita maximizar beneficios de forma continua a empresas de reducida dimensión (PYME).

A partir de este momento, Seresco S.A. se le conocerá como el Cliente del propio software, cuyo objetivo es la venta del producto resultante de este proyecto a diferentes empresas.

A través de técnicas de **Cost Deployment** [3], se harán los diferentes cálculos que permitan la maximización de beneficios, buscando la reducción de coste, o una mejora en los procesos productivos. Inicialmente, partimos de unos resultados y gastos derivados de realizar procesos. Una vez completado todas las fases de *Cost Deployment* [4], la salida que deberemos obtener es un plan de reducción de costes en un proyecto determinado.

Cost Deployment [5] se define como una técnica que permitan evaluar efectivamente las diferentes áreas que forman el proceso de fabricación de un producto dividiéndolo en piezas pequeñas que permitan identificar las causas que producen esas pérdidas. Este método fue propuesto por **Yamashina y Kubo** en 2002 [6] para detectar las principales causas de las pérdidas y poder organizar y priorizar los procesos para mejorarlo. Para hacerlo, se propuso hacer uso de cinco matrices de soporte:

- Matriz A: Identificar y cuantificar las pérdidas.
- Matriz B: Clasificar las causas y los efectos de las relaciones.
- Matriz C: Conectar perdidas y coste de fabricación.
- Matriz D: Conectar pérdidas y las técnicas de mejora.
- Matriz E: Identificar los beneficios y establecer un programa de reducción.

Aunque la técnica parece sencilla, se considera una herramienta avanzada que necesita mucho conocimiento y habilidades para encontrar buenas y completas soluciones al ser una técnica de ayuda a la decisión, permite la toma de éstas, de una forma más efectiva y profesional, pero hay que tomarlas con cautela, al estar basadas en simplificaciones del mundo real. Por ejemplo, estas técnicas no tienen en cuenta factores humanos o del espacio de trabajo pudiendo producir divergencias. Aunque suelen ser útiles para alcanzar una solución óptima, es necesario disponer de acceso a mucha información lo que suele provocar normalmente unos costes añadidos. Por todas las anteriores razones, las técnicas de *Cost Deployment* hay que tomarlas con cautela y hacer uso de ellas como método para cuantificar y facilitar las decisiones, pero nunca como una solución real. Se consideran útiles en sistema estables como la fabricación en la industria.

Usualmente, para llevar a cabo *Cost Deployment*, se suele seguir las siguiente siete fases:

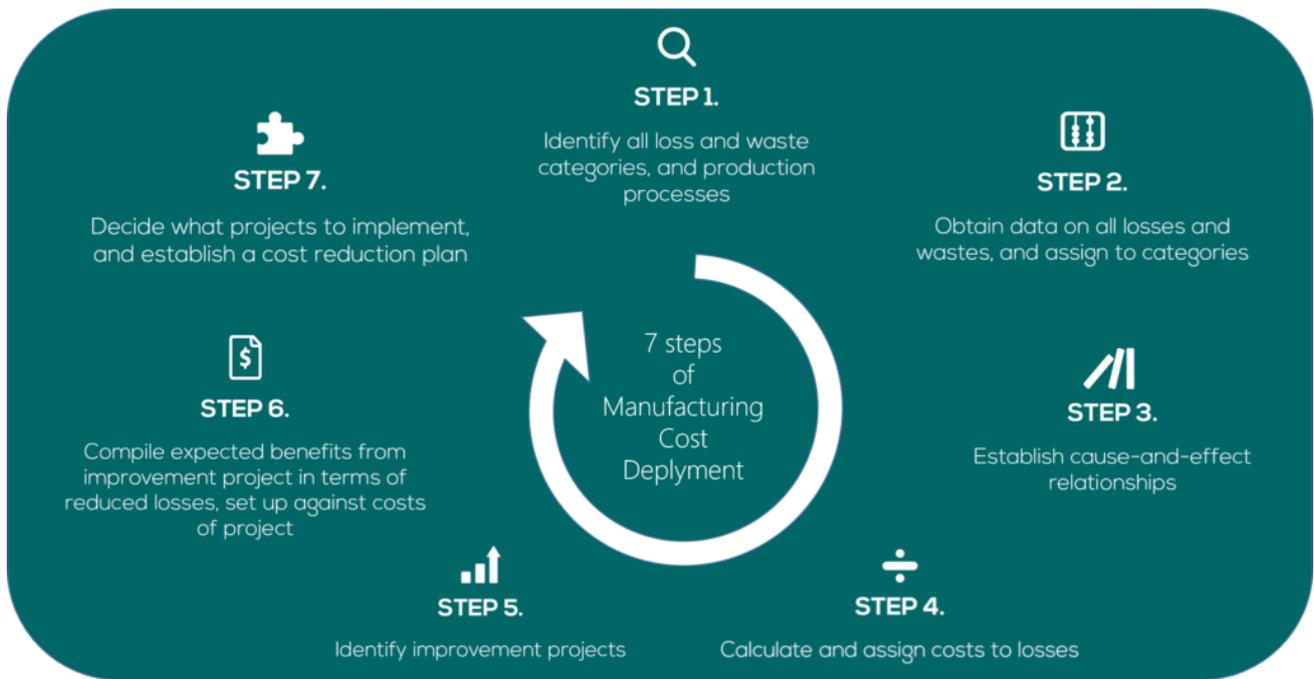


Fig 1. Fases de técnicas de Cost-Deployment [3]

1. Identificar las pérdidas y gastos.
2. Obtener datos de las pérdidas catalogadas.
3. Establecer una causa y efecto.
4. Calcular las pérdidas.
5. Identificar los proyectos a mejorar.
6. Realizar una visión de cómo mejorarían nuestros costes y pérdidas con los datos ajustados.
7. Decidir qué visión es la más adecuada según las necesidades del usuario.

En resumen, nos permite poner énfasis en áreas que producen pérdidas y que tienen oportunidades para eliminarlas y reducirlas permitiendo realizar una evaluación continua de los costes y beneficios.



PSI 1.2: Identificación del Alcance del PSI

Actualmente, el software se corresponde con un nuevo desarrollo enfocado a empresas de pequeño y mediana tamaño. Los objetivos de esta aplicación serán los siguientes:

- Una empresa suele tener varias actividades que producen pérdidas. Actualmente, la mayor parte, hace uso de técnicas muy rudimentarias para la reducción de los costes ocasionados por la realización de una actividad económica. Con la siguiente aplicación, se busca obtener un software web personalizable para un cliente cuyo objetivo es realizar una mejora continua aglutinando en una sola aplicación diferentes actividades que producen unos costes y unas pérdidas.
- Dependiendo del sector económico al que nos refiramos, tendremos unos costes específicos inherentes a esa actividad. La aplicación trata de adquirir una visión general de los diferentes sectores, a los cuales, esta aplicación podrá ser puesta a su disposición.
- Lograr disponer de una aplicación configurable y modular cuyo soporte sea realizable de forma sencilla para el Cliente. Esto implica, disponer de un sistema de módulos independientes que trabajan de forma simultánea.
- El sistema dispondrá de visiones – posibles casos futuros – cuyo resultado será el cálculo de la reducción de costes en diferentes módulos. El objetivo de esto es alcanzar una aplicación donde sea posible analizar las pérdidas que provocan los diferentes áreas o secciones que dispone una empresa.
- Los datos a introducir por el usuario, en su mayoría, serán suministrados en un FactoryModel. Un FactoryModel es una entidad que permite simplificar los cálculos de los sistemas reales buscando ayudar e incrementar la racionalidad de las decisiones necesarias. Éste estará formado por componentes que almacenarán los campos y cálculos que arrojarán los resultados. Posibles ejemplos podrían ser: Costes, Energía, Producción, Calidad, OEE...
- Dado la volatilidad en los datos, y la necesidad por parte de un usuario de hacer pruebas repetidas veces, se dispondrá de la opción de clonar todos los grupos de entidades.
- Tras la recogida de los costes y sus respectivos cálculos, los FactoryModel podrán ser publicados para entrar en el cómputo de ahorros potenciales en una visión, permitiendo añadir o quitar según el criterio del usuario.

Por tanto, en resumen, los objetivos que trata lograr son:

1. Análisis y pruebas para la reducción de costes.
2. Reducción de costes producidos por una actividad.
3. Diseñar una aplicación destinada a PYMEs de diferentes sectores de actividad.

PSI 1.3: Determinación de Responsables

Este proyecto es un desarrollo interno del Cliente. En dicho desarrollo, intervienen los siguientes responsables:



- **Jefe de proyecto:** Se encargará de la supervisión de todas las fases del proyecto, seguimiento del proyecto y su aceptación.
- **Equipo de desarrollo:** Se encargará del análisis, diseño e implementación del sistema resultado del proyecto.
- **Usuarios escogidos de forma aleatoria:** Encargados de realizar pruebas de la aplicación para posibles evoluciones.

PSI 2: DEFINICIÓN Y ORGANIZACIÓN DEL PSI

Para lograr completar el proyecto, se ha definido una planificación donde se detalla la descripción, la duración, el inicio, fin y los predecesores que pudieran existir.

El día uno de octubre de 2019 se mantuvo de la reunión de arranque del proyecto; se detalló el alcance y los objetivos que se quería obtener desarrollando el sistema. Se llegó al acuerdo de que el trabajo debía de ser semanal, con revisión del trabajo realizado cada quince días, o antes si fuera necesario por el alumno.

El proyecto se ha estimado en la cantidad de **356,5 horas**. Comprende cuarenta y cuatro tareas y once hitos cuya finalización darán como resultado el producto final del proyecto.

A continuación, se comentará de forma breve cada etapa del proyecto:

1. **Inicio del proyecto:** El objetivo de esta parte es principalmente:
 - a. Adquirir un conocimiento sobre las técnicas de *Cost Deployment*.
 - b. Identificar el sistema de información que estamos tratando. Comprobar si lo que fue pedido en la reunión inicial con el Cliente se puede llevar a cabo.
 - c. Definir una planificación del desarrollo del proyecto.

	Task Name	Duration (h)				
			1	2	3	4
2	Inicio del proyecto	59				
3	Adquisición del dominio	8				
4	Inicio de Plan de Sistemas de Información	12				
5	Definición de la organización	7				
6	Especificación del Ámbito y Alcance	10				
7	Estudio de la Arquitectura Tecnológica	12				
8	Estudio y valoración de Alternativas	10				

Fig 2. Planificación: Tareas de Inicio de sesión

2. **Análisis del sistema de información:** El objetivo de esta parte es principalmente:
 - a. Definir que debe de hacer el sistema dado su alcance.
 - b. Analizar los requisitos que nos ha pedido el Cliente.
 - c. Analizar y definir los casos de uso.
 - d. Análisis las clases necesarias para que el sistema puede ser diseñado.
 - e. Definir prototipos de las pantallas que deberá tener la aplicación.
 - f. Definición del presupuesto de coste y de cliente, cuya entrada es la planificación definida.
 - g. Identificar y analizar e las pruebas a realizar al sistema.

	Task Name	Duration (h)	Gantt Chart Columns					
			4	5	6	7	8	9
9	Analisis del Sistema de Información	73	█	█	█	█	█	█
10	Definición del sistema	4	█	█				
11	Analisis de requisitos	15		█	█			
12	Analisis de casos de uso	10			█	█		
13	Analisis de Clases	12			█	█		
14	Identificación de Subsistemas de Analisis	6				█	█	
15	Definición de Interfaces de usuario	5					█	█
16	Especificación del Plan de Pruebas	12					█	█
17	Definición del presupuesto	9						█
18	Costes directos	7						█
19	Costes indirectos	2						█

Fig 3. Planificación: Análisis del Sistema de Información

3. **Diseño del sistema de información:** El objetivo de esta parte es principalmente:
- Diseño funcional de la persistencia de la aplicación.
 - Diseño final de las clases involucradas en el sistema.
 - Diseño de la arquitectura necesaria para poder ejecutar el proyecto.
 - Diseño del plan de pruebas a ejecutar sobre el sistema a construir.

	Task Name	Duration (h)	Gantt Chart Columns			
			9	10	11	12
20	Diseño del sistema de información	40	█	█	█	█
21	Diseño físico de datos	5	█	█		
22	Diseño de los microservicios	35		█	█	█
23	Diseño de clases	15		█	█	
24	Diseño de la Arquitectura de Módulos del sistema	10			█	█
25	Especificación Técnica del Plan de Pruebas	10			█	█

Fig 4. Planificación: Diseño del Sistema de Información

4. **Construcción del sistema de información:** El objetivo de esta parte es principalmente:
- Construir el esqueleto de la aplicación web y del servidor.
 - Desarrollo de los diferentes módulos del sistema.
 - Integración y correcto funcionamiento del sistema.
 - Construir los contenedores necesarios para el despliegue.
 - Creación y ejecución de las pruebas necesarias para comprobar el correcto funcionamiento de la aplicación.

	Task Name	Duration (h)	WEEKS														
			12	13	14	15	16	17	18	19	20	21	22	23			
26	Construcción del Sistema de Información	170															
27	Back-end Startup	20															
28	Front-end Startup	15															
29	Desarrollo de la funcionalidad	100															
30	Construcción de módulos	80															
31	Integración tecnologías	10															
32	Construir contenedores	10															
33	Pruebas	35															
34	Unitarias	20															
35	Comportamiento	8															
36	Usabilidad y accesibilidad	5															
37	Carga y Rendimiento	2															

Fig 5. Planificación: Construcción del Sistema de Información

5. **Implantación y Aceptación del sistema:** El objetivo de esta parte es principalmente:
- Elaborar los manuales necesarios para el despliegue y la utilización de la aplicación por parte de un usuario.
 - Ejecutar las pruebas de implantación del sistema.
 - Hacer una validación final con el Cliente para comprobar si el software cumple con los requisitos dados.

	Task Name	Duration (h)	WEEKS	
			23	24
38	Implantación y Aceptación del sistema	14,5		
39	Elaboración de Manuales	6,5		
40	Manual de Instalación	2		
41	Manual de Ejecución	0,5		
42	Manual de Usuario	4		
43	Pruebas de Implantación del Sistema	3		
44	Validación del software	5		

Fig 6. Planificación: Implantación y Aceptación del Sistema

A continuación, se muestra la vista completa de la planificación:

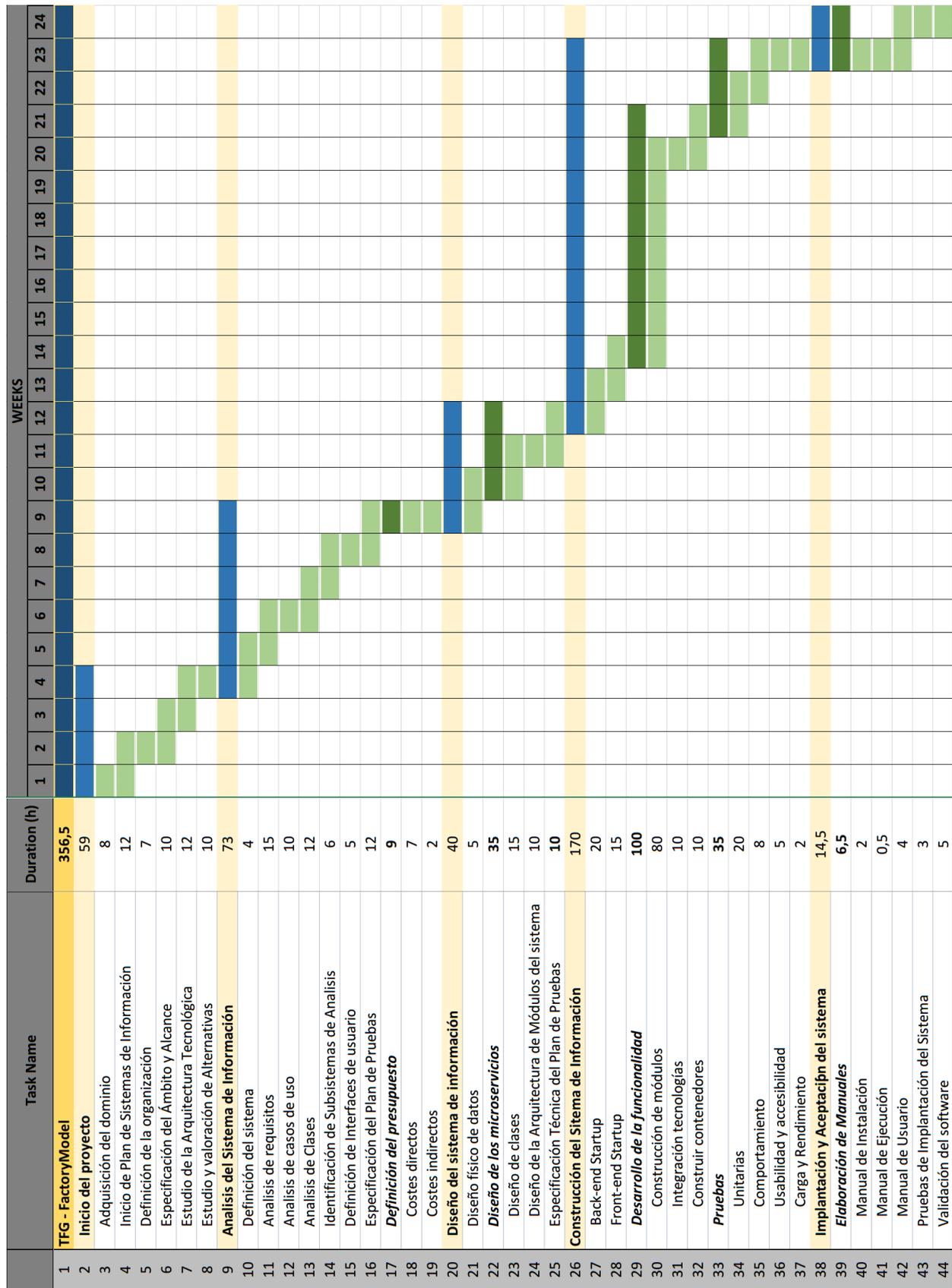


Fig 7. Planificación inicial completa



PSI 2.1: Especificación del Ámbito y Alcance

El objetivo del Cliente será el desarrollo de un software que pueda introducido y comercializado en el mercado específico de aplicaciones de gestión destinadas a PYMEs.

Como se ha indicado, el proyecto es un nuevo desarrollo y se corresponde con una nueva aplicación dentro del abanico disponible del Cliente.

En función de los objetivos tratados, el proyecto tendrá las siguientes fases:

Fase 1: Planificación y definición del proyecto

En esta fase, se sentarán las bases para el desarrollo de una primera versión, sujeta a posteriores modificaciones, que cumpla con las especificaciones mínimas dadas por el Cliente. Se desarrollará una versión inicial del documento que acompañará al software con la finalización del proyecto. Se definirán todos los objetivos a alcanzar en las fases posteriores, análisis del sistema y requisitos iniciales.

Fase 2: Gestión de usuarios

En esta fase, se busca obtener un software que permita crear, borrar y editar usuarios. Se permitirá la generación de tokens JWT para la autenticación en el software. Los objetivos por tratar, por tanto, son:

- Inicio de sesión.
- Creación de usuario.
- Borrado de usuarios.
- Edición de usuarios.

Fase 3: Gestión de Visión, Máquina y FactoryModel

El objetivo en esta fase es alcanzar un software que disponga de los servicios necesarios para el funcionamiento mínimo esperado de la aplicación. Se buscará realizar los motores de gestión de Visiones, Máquinas y *FactoryModels*.

Los objetivos finales son:

- Gestión de Visión.
- Gestión de Máquinas.
- Gestión de *FactoryModels*.

Fase 4: Módulos independientes de Componentes FM

El objetivo que busca alcanzar esta fase es completar la aplicación con la gestión de los diferentes módulos de componentes. Esta fase, se alargará en el tiempo mientras se mantenga el soporte de software; ampliando y modificando los módulos desarrollados.

Inicialmente, se considerará finalizado cuando se cumplan los requisitos mínimos especificados por el Cliente.

Los objetivos, por tanto, son:

- Vista de detalles/edición de los componentes FM.
- Cálculos de los diferentes módulos para permitan disponer de una herramienta de *Cost Deployment*.

PSI 2.2: Organización del PSI

Una vez definido las fases, disponemos de la siguiente organización:

USUARIO	PERFIL	FUNCIÓN
EQUIPO DE SUPERVISIÓN		
JEFE DE PROYECTO	Informático	Supervisión del proyecto y aceptación del Cliente.
EQUIPO DE DESARROLLO		
DESARROLLADOR	Informático	Equipo encargado del software detallado en el apartado de análisis y diseño de la aplicación.
ANALISTA-PROGRAMADOR	Informático	Equipo encargado de análisis de los requisitos que necesita el Cliente.
ARQUITECTO SOFTWARE	Informático	Personal encargado del diseño de la parte de arquitectura software del proyecto.
CONSULTOR DE TECNOLOGÍAS	Informático	Personal encargado del análisis de las distintas opciones aportando la solución más eficiente.
DISEÑADOR GRÁFICO	Informático	Personal encargado de diseñar y adaptar las necesidades a la UI.
EQUIPO DE PRUEBAS		
USUARIO INTERNO	Informático	Pruebas internas de que se cumplen con los requisitos especificados.
USUARIO EXTERNO	Informático / No Informático	Pruebas de uso sobre personas externas que no tienen relación con el software.

Tabla 1. Equipos de trabajo y usuarios

Para poder ejecutar la aplicación, es necesario que se disponga de una cuenta de MS Azure con crédito suficiente para soportar los costes derivados por el desarrollo. Además de esto, es necesario disponer de licencias de Visual Studio para los programadores y demás personal involucrados en el desarrollo.

PSI 2.3: Definición del Plan de Trabajo

Dado el plan de trabajo definido por el Cliente, lo estipulado en el trabajo fin de grado y las necesidades del alumno, se ha desarrollado el siguiente contexto:

- El alumno trabajará de forma autónoma.
- Se podrá paralizar el proyecto en cualquier momento a petición del alumno al Equipo de Supervisión.
- Se debe de mantener actualizada una planificación que siga con los requisitos del Plan de Trabajo.
- El responsable principal en todas ellas será el alumno encargado de ejecutar el proyecto requerido por el cliente.

En octubre de 2019, se realizó la presentación y puesta en marcha del proyecto. Se detallaron los principales planes y se decidió definir una planificación inicial que permitiera tener un control del avance del proyecto. Aunque inicialmente se estimó un periodo de tiempo, durante el desarrollo se consideró interesante ampliar el sistema con varios nuevos componentes, repercutiendo en la duración final del proyecto. Adicionalmente, el proyecto se paralizó entre las fechas de once de diciembre al dieciocho de enero por petición del alumno, haciendo alusión a los exámenes del primer semestre. El proyecto se ha considerado finalizado el día **veintiocho de junio de 2020**.

La duración para dar por concluido el proyecto ha sido de **500,5 horas** con el siguiente desglose:

	Hito	Duración (h)
1	TFG - FactoryModel	500,5
2	Inicio del proyecto	65
9	Análisis del Sistema de Información	109
20	Diseño del Sistema de Información	40
26	Construcción del Sistema de Información	272
38	Implantación y Aceptación del Sistema	14,5

Tabla 2. Resumen por capítulos de la planificación

A continuación, se muestran las horas planificadas según el personal:

I 1	I 2	Descripción	Número de horas
01		Organizativos	
	001	Jefe de proyecto	12,50



	002	Analista de requisitos	137,00
02		Especialistas	
	001	Arquitecto software	89,50
	002	Consultor de tecnologías	19,50
03		Desarrollo	
	002	Programador Senior	116,00
	004	Programador Junior	66,00
	010	Diseñador gráfico	13,00
	011	Tester	47,00
			500,50

Tabla 3. Estadísticas de horas a trabajar por personal

Se puede consultar toda la información relativa a la **planificación del proyecto** en el archivo anexo externo denominado “**Planificación**” – en formato MS Excel y PDF.

PSI 3.1: Selección y Análisis de Antecedentes

Para la realización de este proyecto se ha seguido métrica V3. Dado la naturaleza de este formato y la disponibilidad limitada de personal para trabajar de forma paralela, la mayor parte de las tareas tienen un carácter secuencial, siguiendo con lo estipulado por métrica V3.

Dado el carácter técnico en materia económica que tiene el proyecto, es necesario conocer la normativa seguidos por los diferentes sectores de actividad que se van a tratar, como, por ejemplo:

- **Ley 9/2017**, de 8 de noviembre, de Contratos del Sector Público.
- **RD 1098/2001**, de 12 de octubre, por el que se aprueba el Reglamento general de la Ley de Contratos de las Administraciones Públicas.
- **Ley 24/2013**, de 26 de diciembre, del Sector Eléctrico.
- **Ley 5/2011**, de 29 de marzo, de Economía Social.



PSI 7: DEFINICIÓN DE LA ARQUITECTURA TECNOLÓGICA

Dado la necesidad que el Cliente ha manifestado de realizar un producto reutilizable, parametrizable y mantenible, es necesario definir una arquitectura compleja que nos permita conseguir los objetivos de la forma óptima.

PSI 7.1: Identificación de las Necesidades de Infraestructura Tecnológica

Las necesidades que nos exige el Cliente son las siguientes:

1. Configurable.
2. Distribuible.
3. Modular.
4. Ampliable.

Al definir la infraestructura, nos especifica que es necesario que el desarrollo sea desarrollado utilizando la tecnología .NET en un servicio alojado en la nube de MS Azure.

Las diferentes tecnologías que a las que nos podríamos enfrentar son las siguientes:

- **Arquitectura basada en monolitos:** Tipo de arquitectura usada principalmente en proyectos de un pequeño tamaño que nos obliga a desplegar la aplicación completa en todos los servidores. Complica el tema de distribución y modularidad necesaria.
- **Arquitectura basada en microservicio:** Tipo de arquitectura que define una aplicación como un conjunto de servicios que funcionan independientes, sin tener relación real entre ellos. Nos aporta una arquitectura modular, mantenible, configurable y distribuible.

PSI 7.2: Selección de la Arquitectura Tecnológica

La configuración del sistema que se ha decidido es una arquitectura basada en microservicios. El sistema será desplegado en un servidor Azure que nos aportará de la seguridad necesaria ante posibles ataques. Además, se dispondrá de una capa adicional de protección aportada por el sistema de alojamiento de **Azure Kubernetes Service**.

Un diagrama donde se modela la infraestructura será el siguiente:

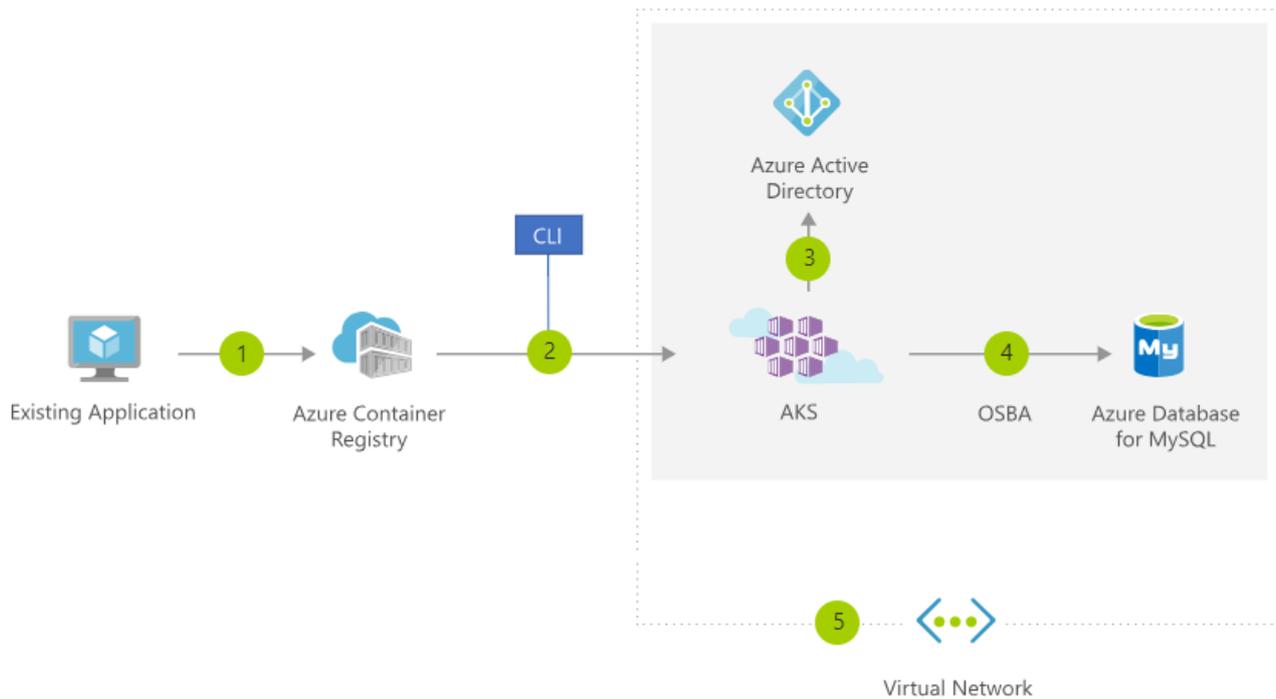


Fig 8. Ejemplo de arquitectura usada en un sistema que usa AKS [7]

En la sección [DSI 5: Diseño de la Arquitectura de Módulos del Sistema](#) se detalla en mayor detalle la arquitectura elegida.



Capítulo 2 ESTUDIO DE VIABILIDAD DEL SISTEMA

FASE DE DESARROLLO

EVS

EVS 4, 5 Y 6: ESTUDIO, VALORACIÓN DE ALTERNATIVAS DE SOLUCIÓN Y SELECCIÓN DE ALTERNATIVA FINAL

En este capítulo, se busca hacer un proceso de estudio de alternativas para escoger la que mejor se adapte al sistema.

Descripción de las alternativas

Se hará un análisis amplio de las diferentes alternativas que existen en el mercado para este proyecto. Se van a contemplar las mejores opciones y más usadas en la actualidad.

Back-end

Debido a las restricciones dadas por el Cliente, en esta parte de desarrollo solo vamos a poder hacer uso de alguna de las tecnologías que usa .NET. Dentro de ellas, podemos diferenciar varias:

.NET Framework (4.x)



Fig 9. Logo de .NET Framework

.NET Framework es la primera versión del Framework desarrollado por Microsoft con posibilidad de usar Visual Basic, C# o F#. Actualmente, sigue siendo la opción natural para muchos escenarios existentes, no pudiendo reemplazarse.

Nos podemos encontrar las siguientes ventajas con respecto a otros sistemas:

- La mayor parte de aplicaciones en C# están basadas en .NET Framework.
- La superficie de API es mucho mayor que su competencia. Prácticamente todas las librerías son compatibles.
- Algunas librerías de Azure SDK solo son compatibles con .NET Framework.
- Algunas tecnologías no están disponibles en sus competidores:
 - **Aplicaciones de formularios Web Forms ASP.NET**
 - **Aplicaciones de ASP.NET Web Pages**
 - **Implementación de servicios WCF**

- **Compatibilidad con lenguajes:** Compatibilidad total de los tres lenguajes de programación.

En el momento de hacer el proyecto, nos encontramos con la que será la última versión de este Framework. Posteriormente, se pasará a .NET 5.0 (versión actualizada de Core).

.NET Core (3.x)

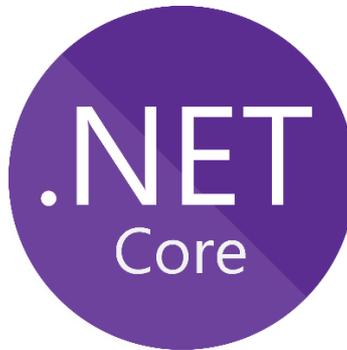


Fig 10. Logo de .NET Core

.NET Core se corresponde con la nueva versión desarrollada por Microsoft con el fin de mejorar la productividad. Es una versión modificada de .NET Framework cuyo objetivo es tener una versión más rápida, ligera y actualizada a las tendencias de estos días.

Las características que nos aporta son:

- Aplicación **multiplataforma**.
- Haga uso de una **arquitectura basada en Microservicios**. Permite combinar varios microservicios haciendo uso de diferentes tecnologías.
- **Uso de contenedores** de Docker. Uso normal junto a una arquitectura de microservicios.
- Necesidad de sistemas **escalables** y de **alto rendimiento**.
- Necesidad de diferentes versiones de .NET en paralelo.

Se corresponde con una tecnología más nueva que sustituirá a .NET Framework el año actual con su versión 5. Es su evolución natural.

Front-end

En el lado del *Front-end* no hay restricciones por parte del cliente en cuanto a la tecnología a utilizar. Vamos a considerar una serie de *Frameworks* actuales que usan diferentes lenguajes de programación [8].

Razor/Blazor



Fig 11. Logo de Blazor

Blazor es un marco de trabajo para la creación de **interfaces de usuario web interactivas** del lado cliente con .NET:

- Crea interfaces de usuario completamente interactivas con **C#** en lugar de JavaScript.
- Represente la interfaz de usuario como HTML y CSS para la compatibilidad con todos los exploradores, incluidos los móviles.
- El uso de .NET para el desarrollo web en el lado cliente ofrece las siguientes ventajas:
 - Escribe el código en **C#** en lugar de JavaScript.
 - Aprovechamiento del ecosistema y las **bibliotecas .NET existentes**.
 - Uso compartido de la lógica de aplicación en el servidor y el cliente.
 - Beneficios de **rendimiento, confiabilidad y seguridad de .NET**.
 - Conjunto común de lenguajes, marcos y herramientas que son estables, completos y fáciles de usar.

Las aplicaciones de **Blazor** se basan en componentes. Un componente en Blazor, es un elemento de la interfaz de usuario, como una página, un cuadro de diálogo o un formulario de entrada de datos. La clase del componente se escribe normalmente en forma de una página de marcado de Razor, con extensión de archivo `.razor`.

Razor es una sintaxis para combinar marcado HTML con código de C# diseñada para aumentar la productividad del desarrollador. Los componentes de Blazor son interoperables con los componentes Razor.

Angular



Fig 12. Logo de Angular

Angular es un framework de código abierto creado por Google en 2010 y desarrollado en **Typescript** frecuentemente utilizado para aplicaciones web de una **sola página**.

Aporta todas las necesidades básicas de un *framework* para creación de páginas web, al igual que sus competidores.

Las características que tiene son:

- Hace uso de un **árbol DOM real**.
- Framework más usado en la industria actual.
- **Mayor diversidad de componentes** para construir la aplicación.
- Permite todo lo necesario para routing, plantillas para hacer pruebas.
- Diseñado para aplicaciones de **gran escala**.

Es decir, tenemos una plataforma muy fiable, que aunque no sea la más rápida, es la más usada para aplicaciones de uso a gran escala. Debido a ser el más usado, se convierte en el más fácil de obtener información relativa a errores.

Vue.js



Fig 13. Logo de Vue.js

Vue.js es un *framework* escrito en **Javascript** modelo de código abierto usado para contruir aplicaciones de una sola página.

Las características que tiene son:

- Hace uso de un **árbol DOM virtual**. Permite una significativa mejora en la velocidad en conjunción a una experiencias sin bugs.

- *Framework* menos pesada.
- Mayor facilidad de desarrollo (menor curva de aprendizaje) que su competencia.

En conclusión, está es la versión más ligera y más actual de los *framework* de desarrollo web. Es una versión muy optimizada pero que su uso está limitado por el bajo índice de mercado que tiene. Es buena opción para desarrollos pequeños que no necesitan toda una infraestructura.

React

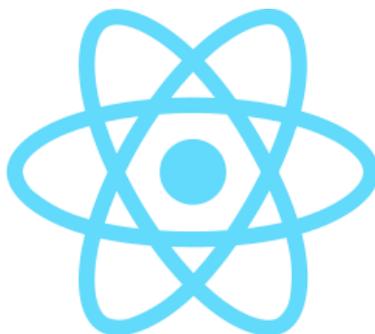


Fig 14. Logo de React

React es un framework creado por Facebook en 2013 escrito en **Javascript** de código abierto diseñado para crear interfaces de usuario con el objetivo de facilitar el desempeño de aplicaciones de una sola página.

Las características que tiene son:

- Hace uso de un **árbol DOM virtual**. Permite una significativa mejora en la velocidad de carga de las librerías.
- Ofrece la mayor flexibilidad del mercado.
- Facilidad de **mantenimiento de código**.

Recalipitulando, tenemos a un framework fiable, mantenido por Facebook con un alto nivel de uso en la industria. Aporta la mayor flexibilidad y un árbol DOM virtual.

Selección de la solución elegida

Una vez explicado los diferentes *frameworks* que se podrían usar, se seleccionarán los que mejor se adapten al sistema a desarrollar.

Primeramente, se va a comentar la elección de la tecnología de Back-end. Se ha decidido usar **.NET Core** por las siguientes razones:

- Multiplataforma
- Diseñado para microservicios
- Compatible con contenedores y orquestadores de contenedores.
- Más ligera y modular.
- Versión más actual a .NET Framework. Se corresponde con la evolución natural de la plataforma.



En el lado del Front-End, tal como se ha visto en las soluciones dispuestas, es muy importante hacer uso de un *framework* actual que nos proporcione de forma sencilla y óptima las necesidades dadas por el usuario.

Desde el punto de vista del desarrollo, es muy importante disponer de una comunidad amplia que nos permita encontrar soluciones a posibles problemas que aparezcan, soporte para librerías necesarias o disponer de actualizaciones de forma asidua.

Desde el punto de vista del usuario y de un cliente, es importante que sea una plataforma mantenible, con el mínimo coste en el mantenimiento y que permita compartir funciones entre sistemas independientes, buscando evitar rehacer una misma funcionalidad varias veces.

En conclusión, el sistema de módulos que nos proporciona **Angular** y la facilidad de mantenimiento que nos aporta son las principales razones que hacen que la mejor solución para este proyecto sea Angular, en su última versión 9. Nos aporta todo lo necesario para poder crear un cliente completo de forma sencilla.



Capítulo 3 ANÁLISIS DEL SISTEMA DE INFORMACIÓN

FASE DE DESARROLLO

ASI

ASI 1: DEFINICIÓN DEL SISTEMA

Determinación del Alcance del Sistema

El objetivo del sistema a desarrollar es analizar, diseñar e implementar un sistema que permita detectar y reducir las posibles pérdidas derivadas de ejercer una actividad económica.

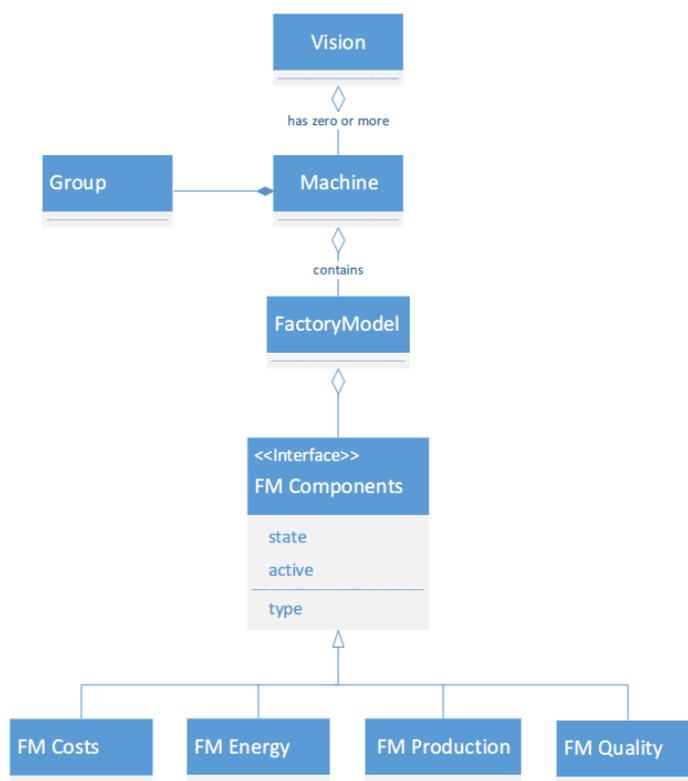


Fig 15. Diagrama de clases del dominio

El sistema dispondrá de una serie de visiones. Una **visión** es una entidad que almacena los resultados de un periodo de tiempo finito, normalmente un año. Deberá de disponer de un nombre identificativo y del área a desarrollar.

Una visión contendrá máquinas, es decir, sistemas productivos. Vendrán dadas por las asociaciones o grupos por la que está formada una empresa. Por ejemplo, si una empresa tiene dos secciones independientes, cada una sería una máquina que pertenece a una la visión (Empresa). Una máquina pertenecerá a una agrupación cuyo objetivo es agrupar máquinas características similares.

A su vez, una máquina almacenará posibles situaciones creadas por el usuario cuyo objetivo será generar supuestos que nos permitan compararlos con el fin de quedarnos con el mejor resultado. Esto se conocerá como FactoryModel. Dado el contexto, los FactoryModel tendrán un estado de Borrador o Publicado, teniéndose en cuenta solo los elementos que se encuentren en publicado para los cálculos. Solo podrá existir un FactoryModel publicado por máquina.



Un FactoryModel estará formado por componentes que almacenarán la información de los datos introducidos por el usuario. Los componentes tendrán la lógica necesaria para realizar todos los pertinentes cálculos. Según los componentes desplegados en el sistema, la aplicación debería poder adaptarse. Todos los componentes tendrán una relación entre el valor de referencia y el valor que nos gustaría alcanzar, referencia ajustada. Los componentes que se proponen en una primera instancia son los siguientes:

- **Costes:** Componente que nos permite reducir los costes fijos, variables y mano de obra que puedan existir en un proceso productivo. Según el tipo de coste, los cálculos serán diferentes.
- **Producción:** Componente que nos permite mejorar la producción por hora de un determinado producto. La mejora puede residir en fabricar más en menos tiempo y/o reducir los costes de fabricación.

Adicionalmente, se han analizado los siguientes componentes:

- **Energía:** Componente que nos permite reducir los costes derivados de la energía eléctrica. En una primera instancia, se contempla tener en cuenta la potencia y el consumo eléctrico producido. Debe de mantener una similitud con una factura de electricidad para permitir un paso rápido de datos a la aplicación. Debe de considerarse que puedan existir diferentes discriminaciones horarias en la tarifa contratada por una empresa (tarifas 3.X o tarifas 6.x).
- **Calidad:** Componente que nos permite reducir las pérdidas derivadas por “defectos” de un producto/servicio generado en un proceso productivo. Los diferentes productos son analizados individualmente identificando las posibles pérdidas producidas y su influencia en el coste final de producción.

ASI 2: ESTABLECIMIENTO DE REQUISITOS

Obtención de los Requisitos del Sistema

Requisitos Funcionales

CÓDIGO	NOMBRE REQUISITO	DESCRIPCIÓN DEL REQUISITO
RF.LG.1	Inicio de sesión	Permitir a un usuario entrar en sesión.
RF.US.1	Dar de alta a usuarios	Permitir crear nuevos usuarios en la aplicación.
RF.US.2	Consultar lista usuarios	Consultar la lista de usuarios disponibles en la aplicación.
RF.US.3	Modificar usuarios	Permitir modificar usuarios existentes en la aplicación.
RF.US.4	Consultar detalles de un usuario	Consultar los detalles de los usuarios dados de alta en la aplicación.
RF.VS.1	Consultar visiones	Consultar un listado de visiones disponibles en el sistema.
RF.VS.2	Añadir visiones	Permitir añadir nuevas visiones al sistema.
RF.VS.3	Modificar visiones	Permitir modificar visiones ya existentes en el sistema.
RF.VS.4	Clonar visiones	Permitir clonar las visiones y todas las dependencias.
RF.MQ.1	Consultar maquinas	Consultar un listado de máquinas disponibles en el sistema.
RF.MQ.2	Añadir maquinas	Permitir añadir nuevas máquinas al sistema.
RF.MQ.3	Modificar maquinas	Permitir modificar máquinas ya existentes en el sistema.
RF.MQ.4	Clonar maquinas	Permitir clonar las máquinas y todas las dependencias.
RF.FM.1	Consultar <i>FactoryModels</i>	Consultar un listado de <i>FactoryModels</i> disponibles en el sistema.
RF.FM.2	Añadir <i>FactoryModels</i>	Permitir añadir nuevas <i>FactoryModels</i> al sistema.
RF.FM.3	Modificar <i>FactoryModels</i>	Permitir modificar <i>FactoryModels</i> ya existentes en el sistema.
RF.FM.4	Clonar <i>FactoryModels</i>	Permitir clonar las <i>FactoryModels</i> y todas las dependencias.
RF.FM.5	Publicar <i>FactoryModels</i>	Permitir convertir a un <i>FactoryModels</i> en modo lectura y que se tenga en cuenta en el cómputo total.
RF.FMC.1	Seleccionar los componentes disponibles en un <i>FactoryModel</i>	Permitir activar y desactivar un componente.
RF.FMC.2	Consultar los componentes disponibles en un <i>FactoryModel</i>	Permitir consultar la vista en detalles de los componentes que tiene un <i>FactoryModel</i> .
RF.CS.1	<i>Costes</i> Vista de detalles y edición	Permitir consultar la vista con la información relativa a <i>Costes</i> .
RF.ES.1	<i>Energía</i> Vista de detalles y edición	Permitir consultar la vista con la información relativa a <i>Energía</i> .



RF.PS.1	<i>Producción</i> Vista de detalles y edición	Permitir consultar la vista con la información relativa a Producción.
RF.QS.1	<i>Calidad</i> Vista de detalles y edición	Permitir consultar la vista con la información relativa a Calidad.

Tabla 4. Resumen de requisitos del sistema funciones

La lista de requisitos funcionales es extensa y se ha decidido moverla al [Anexo II: Requisitos funcionales](#) para facilitar la lectura del trabajo fin de grado.

Requisitos Específicos

Interfaces Externas

Interfaces de Usuario

- RNF.IE.1.** El sistema deberá ser accesible a través de cualquier dispositivo con conexión a internet.
- RNF.IE.2.** El sistema deberá permitir la internalización del sistema.
 - RNF.IE.1.1.** El sistema deberá estar adaptado para la cultura española.
 - RNF.IE.2.1.** El sistema deberá estar adaptado para la cultura inglesa.
- RNF.IE.3.** El sistema deberá de permitir cambiar de idioma de forma automática.

Interfaces Hardware

- RNF.IH.1.** El sistema dispondrá de una base datos que se usará para almacenar toda la información relevante.
- RNF.IH.2.** El sistema dispondrá de una cola de eventos.
- RNF.IH.3.** El sistema dispondrá de un *Service Discovery*.

Interfaces de Comunicaciones

- RNF.IC.1.** El sistema dispondrá de un punto de acceso al servidor.
- RNF.IC.2.** El sistema dispondrá de un sistema de paso de datos entre los diferentes servicios de forma asíncrona.
- RNF.IC.3.** El sistema permitirá descubrir los servicios accesibles dentro del sistema.
 - RNF.IC.3.1.** Cada servicio estará registrado en un servicio que conocerá todos los servicios.
 - RNF.IC.3.2.** Los servicios no conocerán al resto.
 - RNF.IC.3.3.** El servicio se registrará en el momento que se encuentre disponible para responder mensajes.
- RNF.IC.4.** El sistema se podrá comunicará con el *Active Directory* de la empresa.
 - RNF.IC.4.1.** La comunicación de la contraseña debe de ser encriptada.
- RNF.IC.5.** El sistema contendrá enlaces a distintos sitios web.
- RNF.IC.6.** El sistema dispondrá de un certificado TLS en todas sus comunicaciones.



Requisitos de Rendimiento

- RNF.R.1.** El sistema tiene que responder en un tiempo inferior a 5 segundos.
- RNF.R.1.1.** Si no se cumple, la petición se desecha y se cancela la petición.

Requisitos Lógicos de BD

- RNF.BD.1.** El sistema dispondrá de varios sistemas de base de datos.
 - RNF.BD.1.1.** Estos deberán ser diferentes.
 - RNF.BD.1.2.** Cada microservicio dispondrá de su base de datos.
 - RNF.BD.1.2.1.** Solo dicho microservicio puede realizar operaciones sobre el sistema de base de datos.
 - RNF.BD.1.3.** Un microservicio puede tener varios sistemas para almacenar la información.

Requisitos de Desarrollo

- RNF.DEV.1.** Se deberá cumplir la ley 15/1999 con respecto a la protección de datos de carácter personal.
 - RNF.DEV.1.1.** El sistema hará uso de la política de Protección de Datos que estime el Cliente.
 - RNF.DEV.2.1.** La seguridad de los datos de la aplicación debe ser garantizada, de forma que se evite su alteración o pérdida.
 - RNF.DEV.3.1.** El usuario implicado debe tener acceso total a sus datos personales.

Atributos del sistema

Seguridad

- RNF.SEG.1.** El sistema deberá garantizar que la comunicación con el mismo sea segura.
 - RNF.SEG.1.1.** Se hará uso de un certificado SSL/TLS en todas las comunicaciones Cliente-Servidor.
 - RNF.SEG.2.1.** Se hará uso de una versión igual o superior a TLS1.2.
- RNF.SEG.2.** La información manejada por el sistema estará protegida de acceso no autorizado y divulgación.
 - RNF.SEG.2.2.** El sistema ocultará toda la información relativa al servidor.
 - RNF.SEG.3.2.**
- RNF.SEG.3.** El sistema garantizará que los usuarios autorizados se le garantizará el acceso a la información.
- RNF.SEG.4.** El sistema deberá de asegurarse de que el usuario se identifica y de que su identidad sea real y no suplantada.
 - RNF.SEG.1.4.** Se comprobará si la autenticación sigue siendo válida cada 1800 segundos.



Fiabilidad

- RNF.FI.1.** El sistema permitirá recuperarse ante posibles fallos del sistema.
- RNF.FI.2.** El sistema almacenará los cambios pendientes generados en otro microservicio para evitar posibles pérdidas de información ante posibles fallos del sistema.

Soporte

- RNF.SOP.1.** El sistema dispondrá de soporte directo durante 12 meses.
 - RNF.SOP.1.1.** Se garantizará el suministro de la infraestructura necesaria para el funcionamiento.
 - RNF.SOP.1.2.** Se garantizará un correcto funcionamiento del sistema.

Usabilidad

- RNF.US.1.** El sistema será un servicio web.
- RNF.US.2.** El sistema garantizará que será compatible con un monitor de X resolución.
 - RNF.US.2.1.** Se hará uso de colores que se ajusten a los estándares de usabilidad.
- RNF.US.3.** Deberá de mantener un formato similar al de una hoja de estilos.
- RNF.US.4.** El sistema permitirá debe de soportar varios lenguajes y culturas.
 - RNF.US.4.1.** Por defecto se seleccionará el valor del navegador.
- RNF.US.5.** El sistema permitirá configurar la moneda.
 - RNF.US.5.1.** Por defecto se seleccionará el valor del navegador.

Mantenibilidad

- RNF.MAN.1.** El sistema permitirá hacer una gestión del sistema de forma automática.
 - RNF.MAN.1.1.** Cada microservicio será tratado como un contenedor independiente al resto.
 - RNF.MAN.1.2.** Todo se manejará a través de un orquestador de contenedores.

Otros requisitos

N/A

Identificación de Actores del Sistema

Los actores interesados en el sistema son todos aquellos que tienen una interacción directa o indirecta con el proyecto. Los actores que, por tanto, tienen relación sobre las actividades o decisiones del proyecto, y cuyo interés es obtener un correcto funcionamiento del sistema son:

Usuario anónimo: Actor que interactúa con el sistema antes de iniciar sesión en la aplicación. Tiene solo acceso al *layout* público del sistema.

Usuario observador: Actor que interactúa con el sistema. Se corresponde con un rol que solo tiene permisos de solo lectura de la aplicación.

Usuario logueado/Usuario: Actor que interactúa con el sistema. Se corresponde con un rol que tiene permisos de escritura en el sistema. Hereda de Usuario observador.

Usuario administrador: Actor que interactúa con el sistema. Se corresponde con un rol con permisos de administrador. Hereda de Usuario/Usuario logueado.

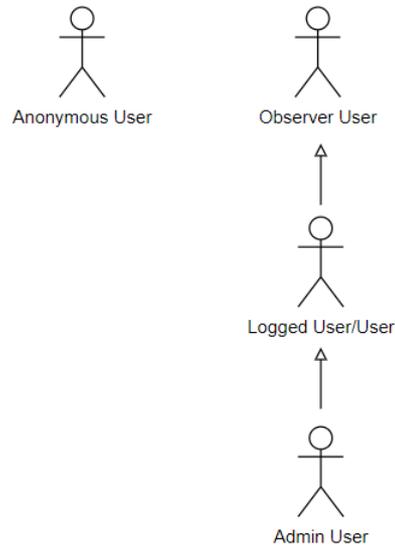


Fig 16. Diagrama de actores y sus relaciones

Especificación de Casos de Uso

En una primera instancia, se definirá un diagrama de casos de uso de que debe de hacer la aplicación en general. Posteriormente, se detallarán todos los casos de uso de forma más detallada definiendo el nombre, descripción, cometido y pasos que debe de seguir ese caso de uso en la aplicación.

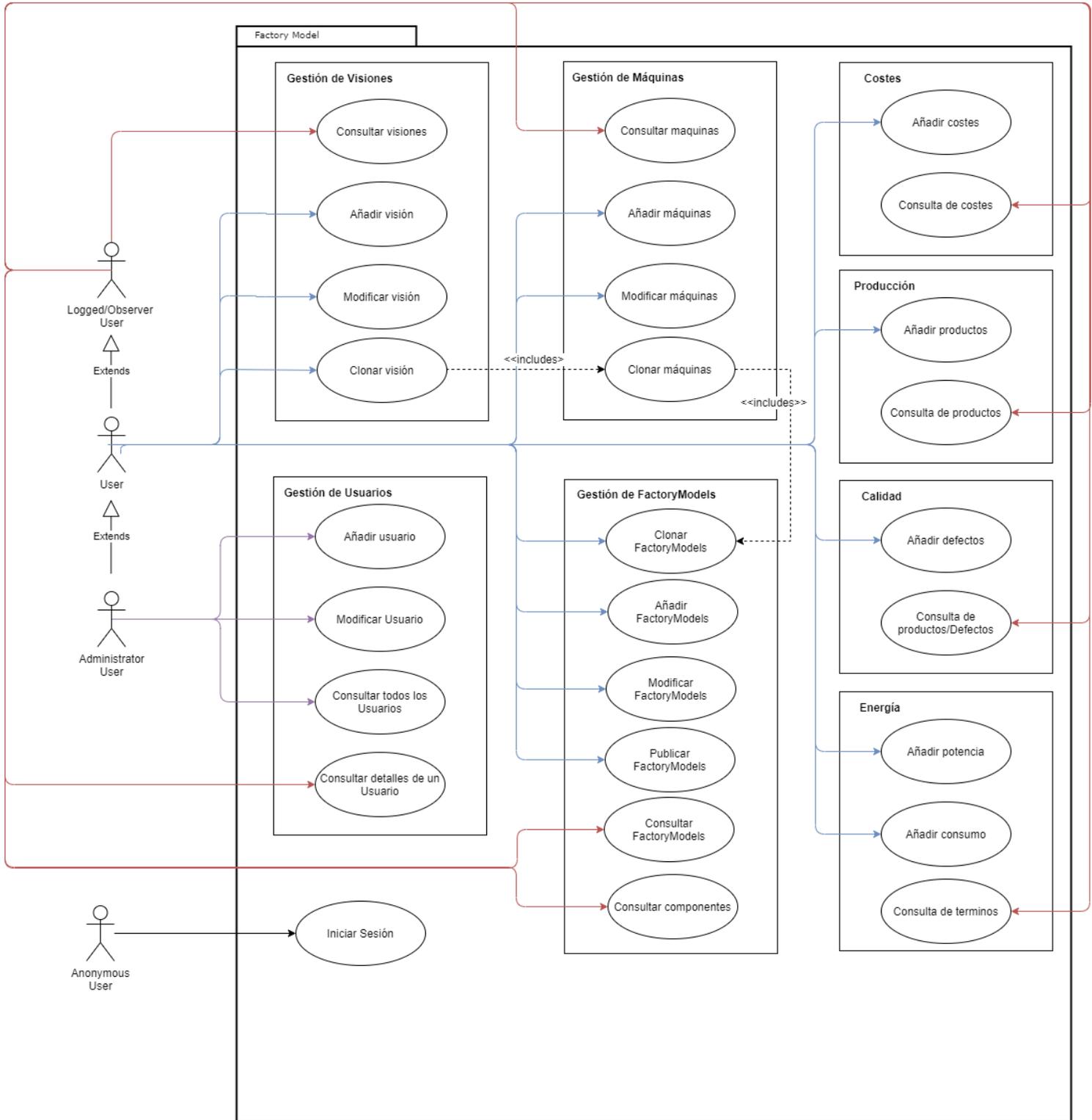


Fig 17. Diagrama de casos de usos



ASI 3: IDENTIFICACIÓN DE SUBSISTEMAS DE ANÁLISIS

En este apartado del análisis, se ilustrará los diferentes submódulos en los que se dividirá el sistema para realizar los casos de uso especificados por el Cliente. Debido a la naturaleza requerida por el Cliente, de ser un sistema parametrizable y modular, los módulos no interactuarán entre ellos de forma síncrona.

Descripción de los Subsistemas

En los próximos párrafos, se describirá la descomposición que se ha hecho del sistema en módulos o subsistemas. Cada módulo será independiente entre ellos, pudiendo hacer uso de librerías externas comunes entre ellas.

Subsistemas propios

Los subsistemas propios son los desarrollados en este proyecto para alcanzar los requisitos que nos especifica el Cliente.

A mayores de los siguientes subsistemas, el proyecto deberá incluir un subsistema que permita la interacción entre *back-end* con *front-end*.

Podemos identificar los siguientes subsistemas en la aplicación:

Subsistema del cliente

Subsistema que pertenece a la parte de interfaz de usuario. El usuario interactuará con la aplicación a través de dicho subsistema.

Debe de ser un subsistema único que se conecta por HTTP al subsistema de integración *back-front*.

Se hará uso de un patrón Modelo-Vista Vista-Modelo (MVVM) permitiendo tener sincronizados los datos entre la vista y el modelo-vista. En este nivel, no se realizará ningún tipo de lógica de negocio, solo la lógica necesaria por la interfaz.

Subsistema de identificación

Subsistema que pertenece a la lógica de negocio del proyecto encargada de:

1. Gestión de usuarios.
2. Identificación de usuarios.
3. Control de accesos.



Subsistema de visiones

Subsistema que pertenece a la lógica de negocio que permite la gestión de visiones. Es usado desde el subsistema del cliente.

Debe de contener todas las clases requeridas para implementar correctamente el tratamiento de visiones. Se hará uso de un sistema por capas donde habrá independencia entre controladores, servicios y repositorios para acceso a datos.

Tendrá dependencias del subsistema de base de datos SQL y cola de eventos.

Subsistema de máquinas

Subsistema que pertenece a la lógica de negocio que permite la gestión de máquinas. Es usado desde el subsistema del cliente.

Debe de contener todas las clases requeridas para implementar correctamente de un subsistema completo para el tratamiento de máquinas. Se hará uso de un sistema por capas donde habrá independencia entre controladores, servicios y repositorios para acceso a datos.

Tendrá dependencias del subsistema de base de datos SQL y cola de eventos.

Subsistema de *FactoryModels*

Subsistema que pertenece a la lógica de negocio que permite la gestión de *FactoryModels*. Es usado desde el subsistema del cliente.

Debe de contener todas las clases requeridas para implementar correctamente de un subsistema completo para el tratamiento de *FactoryModels*. Se hará uso de un sistema por capas donde habrá independencia entre controladores, servicios y repositorios para acceso a datos.

Tendrá dependencias del subsistema de base de datos SQL y cola de eventos.

Subsistema de costes

Subsistema que pertenece a la lógica de negocio que permite la gestión de costes. Es usado desde el subsistema del cliente.

Debe de contener todas las clases requeridas para implementar correctamente de un subsistema completo para el tratamiento de costes. Se hará uso de un sistema por capas donde habrá independencia entre controladores, servicios y repositorios para acceso a datos.

Tendrá dependencias del subsistema de base de datos SQL y cola de eventos.

Subsistema de energía

Subsistema que pertenece a la lógica de negocio que permite la gestión de energía eléctrica. Es usado desde el subsistema del cliente.



Debe de contener todas las clases requeridas para implementar correctamente de un subsistema completo para el tratamiento de energía. Se hará uso de un sistema por capas donde habrá independencia entre controladores, servicios y repositorios para acceso a datos.

Tendrá dependencias del subsistema de base de datos SQL y cola de eventos.

Subsistema de producción

Subsistema que pertenece a la lógica de negocio que permite la gestión de producción. Es usado desde el subsistema del cliente.

Debe de contener todas las clases requeridas para implementar correctamente de un subsistema completo para el tratamiento de producción. Se hará uso de un sistema por capas donde habrá independencia entre controladores, servicios y repositorios para acceso a datos.

Tendrá dependencias del subsistema de base de datos SQL y cola de eventos.

Subsistema de calidad

Subsistema que pertenece a la lógica de negocio que permite la gestión de calidad. Es usado desde el subsistema del cliente.

Debe de contener todas las clases requeridas para implementar correctamente de un subsistema completo para el tratamiento de calidad. Se hará uso de un sistema por capas donde habrá independencia entre controladores, servicios y repositorios para acceso a datos.

Tendrá dependencias del subsistema de producción, base de datos SQL y cola de eventos.

Subsistemas externos

Los subsistemas externos son los módulos externos ajenos a nuestra organización necesarias para un correcto funcionamiento del sistema. Se puede identificar:

Subsistema de Base de Datos SQL

Contiene todos los servicios necesarios para la conexión, el manejo y el tratamiento de la base de datos. Podría estar formado por una o más base de datos.

Subsistema de Cola de Eventos

Contiene todos los elementos necesarios para la interacción entre módulos del sistema. Pertenece a la parte de *back-end* y no estará disponible desde el exterior.

Su cometido es permitir publicar y suscribirse a eventos de los subsistemas. Podría ser sustituido por un subsistema propietario en un futuro.

Descripción de los Interfaces entre Subsistemas

Dado el carácter modular que tiene una arquitectura basada en microservicios, es imprescindible tener interfaces de comunicación entre ellas. La regla principal que se va a tratar es la de tener un sistema asíncrono que interactúe con el resto de los elementos.

Si hablamos de la parte de *back-end*, se ha decidió hacer uso de interfaces de comunicación basadas en colas de mensajes, donde un módulo manda una acción a realizar a otro módulo y este la completa. Nos aporta la independencia que queremos alcanzar y un control de consistencia mucho mejor al que nos aportaría el uso de un sistema sincrónico.

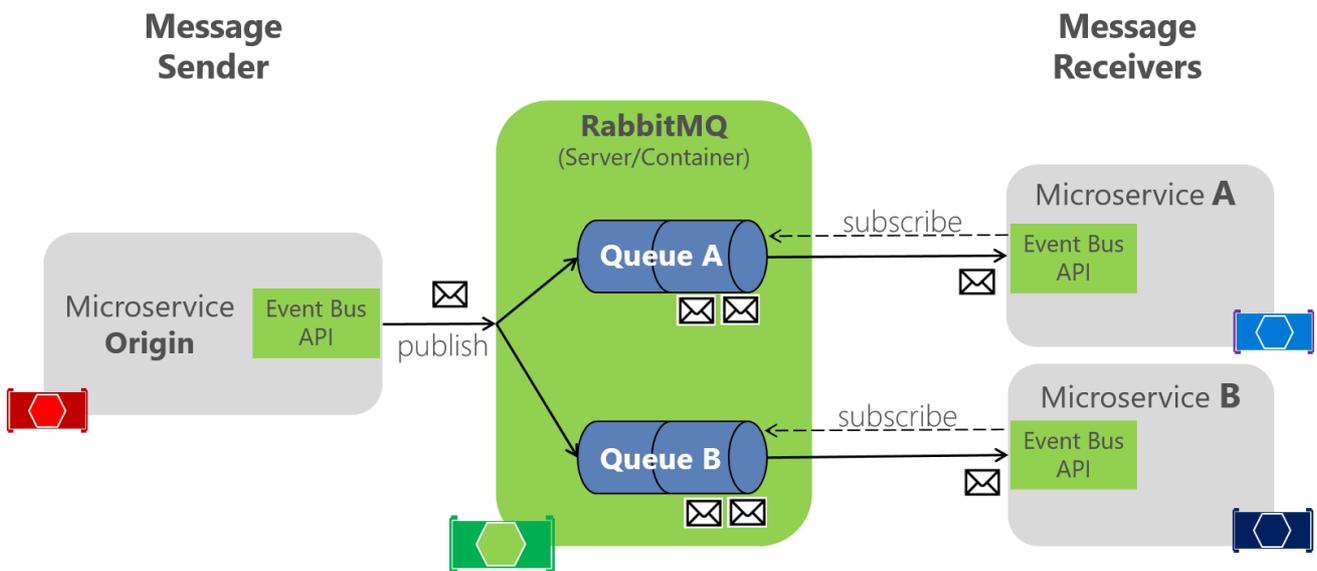


Fig 18. Diagrama de Queue Event [9]

Para la comunicación entre el servicio (o servicios) y la parte de cliente – *front-end*- se ha decidido hacer uso de una API REST basada en HTTP. Todos los subsistemas pertenecientes a la lógica de negocio no tendrán salida a excepción del punto de acceso.

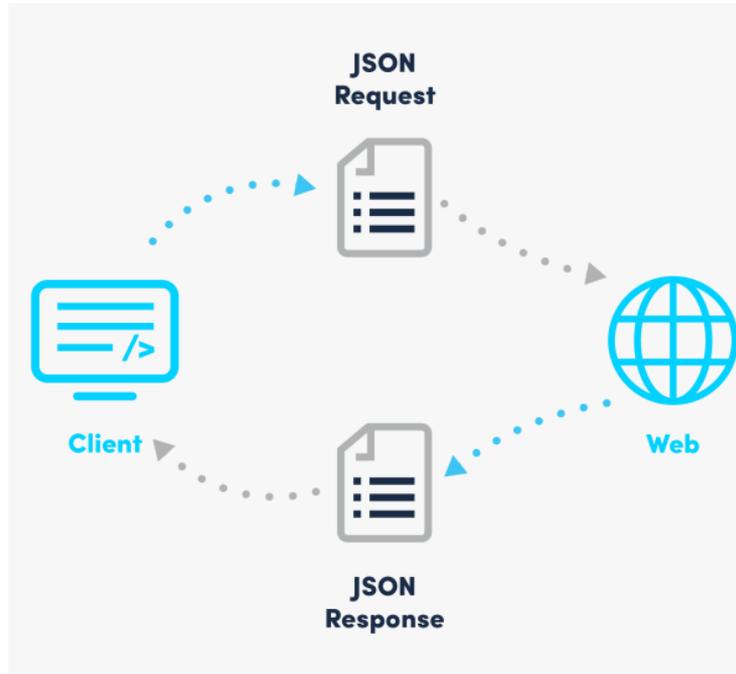


Fig 19. Diagrama de API REST [10]

Para finalizar, la comunicación que se realiza en el subsistema del cliente es diferente. Se tiene una comunicación basada MVVM que nos permitirá tener una actualización de la vista basada en el modelo.

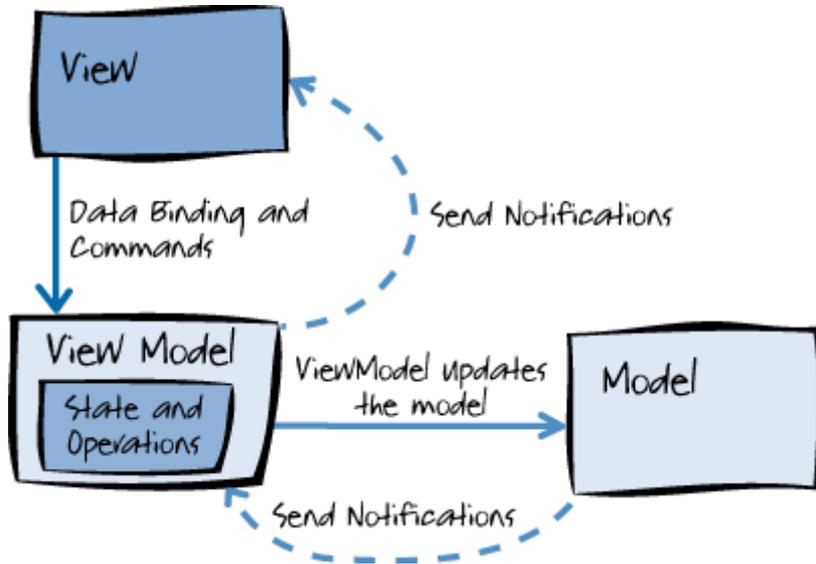


Fig 20. Diagrama de MVVM [11]

ASI 4: ANÁLISIS DE LOS CASOS DE USO

Inicio de sesión

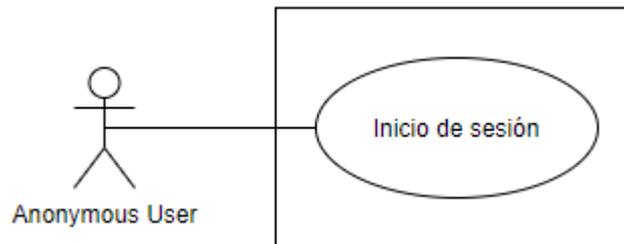


Fig 21. Diagrama de casos de uso: Inicio de sesión

Caso de Uso
CU.1. Inicio de sesión
Requisitos funcionales que satisface
RF.LG.1. El sistema permitirá iniciar sesión a los usuarios registrados previamente en el sistema.
Descripción Caso de Uso
El usuario podrá introducir sus credenciales personales para entrar en la zona privada de la aplicación. En caso de que los datos sean correctos, el usuario se convertirá en un usuario de tipo <i>Observer User</i> .
Precondiciones
Ninguna.
Postcondiciones
Ninguna.
Actores
Anonymous User
Descripción
<ol style="list-style-type: none"> 1. El usuario introducirá su correo electrónico. 2. El usuario introduce su contraseña. 3. El usuario selecciona recordar la contraseña. 4. El usuario presiona el botón de iniciar sesión.
Variaciones
<ul style="list-style-type: none"> ▪ <i>Escenario 1:</i> El usuario no introduce alguno de los campos. <ul style="list-style-type: none"> ○ El sistema mostrará un mensaje de error. ▪ <i>Escenario 2:</i> El usuario se encuentra desactivado en el sistema.

- El sistema mostrará un mensaje de error.

Tabla 5. Caso de uso: Iniciar Sesión

Gestión de Usuarios

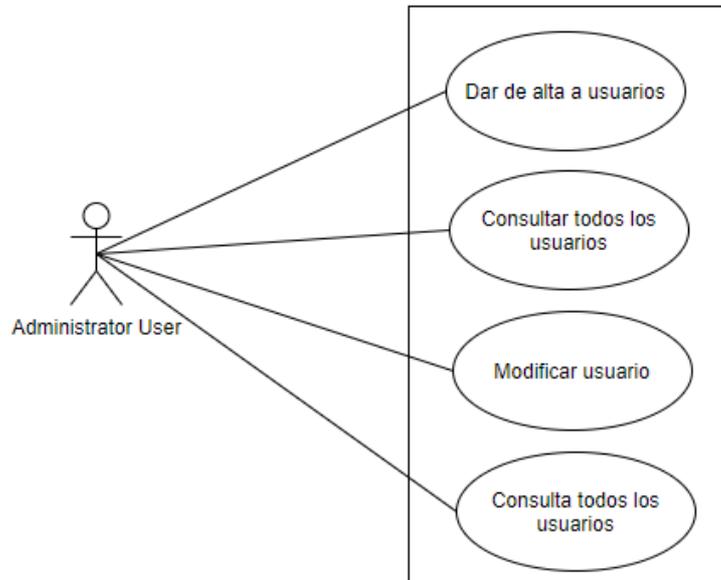


Fig 22. Diagrama de casos de uso: Gestión de Usuarios

Caso de Uso
CU.2. Dar de alta a usuarios
Requisitos funcionales que satisface
RF.US.1. Permitir crear nuevos usuarios en la aplicación.
Descripción Caso de Uso
El programa permitirá añadir un nuevo usuario al sistema. Se pedirá toda la información necesaria para generar una nueva entidad. Se permitirá a un usuario que tenga el permiso de administrador – <i>Administrator User</i> .
Precondiciones
Iniciar sesión
Postcondiciones
La entidad es guardada en el sistema.
Actores
Administrator User
Descripción
CU.1. El usuario introducirá los campos necesarios.

CU.2. El usuario presiona el botón de crear.

Variaciones

- *Escenario 1:* El usuario no introduce alguno de los campos.
 - El sistema mostrará un mensaje de error.
- *Escenario 2:* El usuario desea cancelar la operación.
 - El usuario presiona el botón de cancelar.
- *Escenario 3:* El usuario ya existe en el sistema.
 - El sistema reportará un mensaje de error.

Tabla 6. Caso de uso: Dar de alta a usuarios

Caso de Uso
CU.3. Consulta todos los usuarios
Requisitos funcionales que satisface
RF.US.2. El sistema permitirá consultar los diferentes usuarios dados de alta en el sistema.
Descripción Caso de Uso
El programa mostrará, en formato similar a una tabla, la información relativa de los usuarios disponibles en el sistema. Se permitirá a un usuario que tenga el permiso de administración – <i>Administrator User</i> .
Precondiciones
Iniciar sesión
Postcondiciones
Ninguna.
Actores
Administrator User
Descripción
1. El sistema mostrará toda la información, de forma tabular, los usuarios disponibles en el sistema.
Variaciones
Ninguna.

Tabla 7. Caso de uso: Consulta todos los usuarios

Caso de Uso
CU.4. Modificar usuario
Requisitos funcionales que satisface
RF.US.3. El sistema permitirá modificar datos de los usuarios en el sistema.
Descripción Caso de Uso

El programa permitirá editar un usuario existente en el sistema. Se podrán modificar los datos que no formen la identidad de un usuario. Se permitirá a un usuario que tenga el permiso de administración – <i>Administrator User</i> .
Precondiciones
Iniciar sesión, seleccionar una máquina
Postcondiciones
La entidad es modificada en el sistema.
Actores
Administrator User
Descripción
<ol style="list-style-type: none"> 1. El usuario activará el modo edición pulsando en el botón editar. 2. El usuario introducirá los cambios que considere oportunos. 3. El usuario presiona el botón de guardar cambios.
Variaciones
<ul style="list-style-type: none"> ▪ <i>Escenario 1:</i> El usuario no introduce alguno de los campos. <ul style="list-style-type: none"> ○ El sistema mostrará un mensaje de error.

Tabla 8. Caso de uso: Modificar usuario

Caso de Uso
CU.5. Consulta detalles de un usuario
Requisitos funcionales que satisface
RF.US.4. El sistema permitirá dar de alta a nuevos usuarios en el sistema.
Descripción Caso de Uso
El programa mostrará, en formato similar a una tabla, la información relativa de los usuarios disponibles en el sistema.
Precondiciones
Iniciar sesión
Postcondiciones
Ninguna.
Actores
Logged User, Seresco, Empleados Seresco, Certificadora TLS
Descripción
<ol style="list-style-type: none"> 1. El sistema mostrará la información del usuario activo.
Variaciones
Ninguna.

Tabla 9. Caso de uso: Consultar detalles de un usuario

Gestión de Visiones

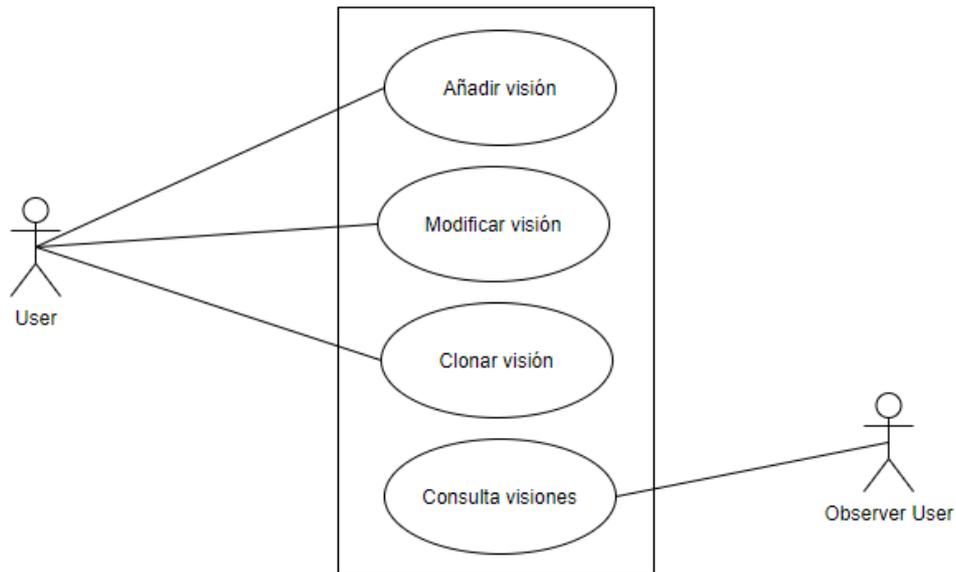


Fig 23. Diagrama de casos de uso: Gestión de Visiones

Caso de Uso	
CU.6. Consulta visiones	
Requisitos funcionales que satisface	
RF.VS.1.	El sistema permitirá consultar las diferentes visiones dados de alta en el sistema.
RF.VS.5.	El sistema permitirá consultar las diferentes visiones dados de alta en el sistema.
RF.VS.6.	El sistema permitirá consultar los FactoryModel publicados de la visión seleccionada.
Descripción Caso de Uso	
El programa mostrará, en formato similar a una tabla, la información pertinente de las visiones disponibles en el sistema. Se debe de permitir obtener una vista detalles.	
Precondiciones	
Iniciar sesión	
Postcondiciones	
Ninguna.	
Actores	
Observer User	
Descripción	



1. El sistema mostrará toda la información, de forma tabular, de las visiones disponibles en el sistema.

Variaciones

Ninguna.

Tabla 10. Caso de uso: Consulta visiones

Caso de Uso
CU.7. Añadir visión
Requisitos funcionales que satisface
RF.VS.2. El sistema permitirá dar de alta nuevas visiones.
Descripción Caso de Uso
El programa permitirá añadir una nueva visión al sistema. Se pedirá toda la información necesaria para generar una nueva entidad. Se permitirá a un usuario que tenga el permiso de edición – <i>User</i> .
Precondiciones
Iniciar sesión
Postcondiciones
La entidad es guardada en el sistema.
Actores
User
Descripción
<ol style="list-style-type: none"> 1. El usuario introducirá los campos necesarios. 2. El usuario presiona el botón de crear.
Variaciones
<ul style="list-style-type: none"> ▪ <i>Escenario 1:</i> El usuario no introduce alguno de los campos. <ul style="list-style-type: none"> ○ El sistema mostrará un mensaje de error. ▪ <i>Escenario 2:</i> El usuario desea cancelar la operación. <ul style="list-style-type: none"> ○ El usuario presiona el botón de cancelar.

Tabla 11. Caso de uso: Añadir visión

Caso de Uso
CU.8. Modificar visión
Requisitos funcionales que satisface
RF.VS.3. El sistema permitirá modificar datos de las visiones del sistema.
Descripción
El programa permitirá editar una visión existente en el sistema. Se pedirá la información que no implique la unidad de la entidad. Se permitirá a un usuario que tenga el permiso de edición – <i>User</i> .
Precondiciones
Iniciar sesión

Postcondiciones
La entidad es modificada en el sistema.
Actores
User
Descripción
<ol style="list-style-type: none"> 1. El usuario activará el modo edición pulsando en el botón editar. 2. El usuario introducirá los cambios que considere oportunos. 3. El usuario presiona el botón de guardar cambios.
Variaciones
<ul style="list-style-type: none"> ▪ <i>Escenario 1:</i> El usuario no introduce alguno de los campos. <ul style="list-style-type: none"> ○ El sistema mostrará un mensaje de error.

Tabla 12. Caso de uso: Modificar visión

Caso de Uso
CU.9. Clonar visión
Requisitos funcionales que satisface
RF.VS.4. El sistema permitirá dar clonar visiones ya existentes.
Descripción Caso de Uso
El programa permitirá clonar una visión junto a todos los elementos que tengan relación con él. Se permitirá modificar la identidad de la entidad. Se permitirá a un usuario que tenga el permiso de edición – <i>User</i> .
Precondiciones
Iniciar sesión, existencia de una entidad original
Postcondiciones
La entidad clonada es guardada en el sistema.
Actores
User
Descripción
<ol style="list-style-type: none"> 1. El usuario presionará en el botón de clonación de una entidad original. 2. El usuario introducirá los nuevos valores de identificación 3. El usuario presiona el botón de clonar.
Variaciones
<ul style="list-style-type: none"> ▪ <i>Escenario 1:</i> El usuario no modifica alguno de los campos. <ul style="list-style-type: none"> ○ El sistema mostrará un mensaje de error. ▪ <i>Escenario 2:</i> El usuario desea cancelar la operación. <ul style="list-style-type: none"> ○ El usuario presiona el botón de cancelar.

Tabla 13. Caso de uso: Clonar visión

Gestión de Máquinas

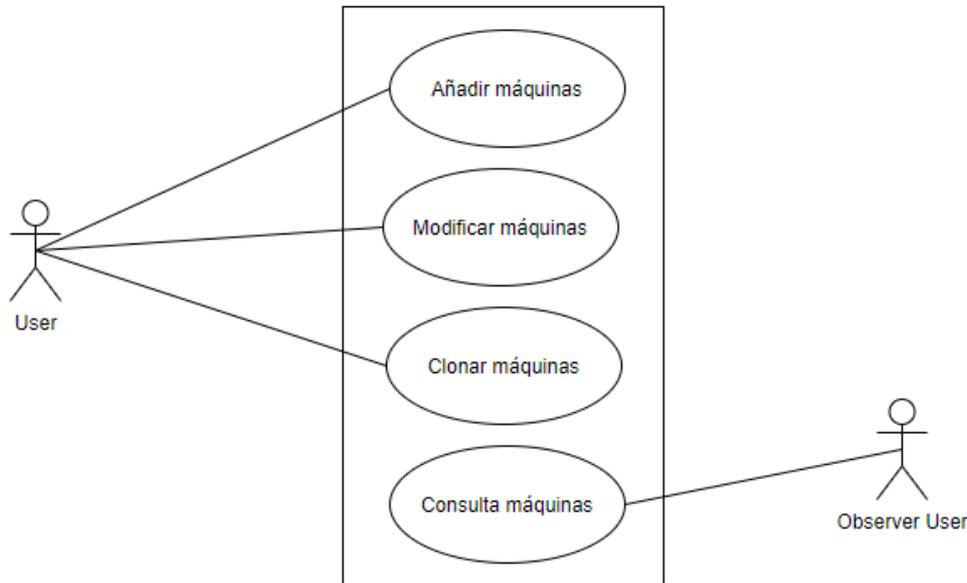


Fig 24. Diagrama de casos de uso: Gestión de Máquinas

Caso de Uso
CU.10. Consulta máquinas
Requisitos funcionales que satisface
RF.VS.5. El sistema permitirá consultar las diferentes visiones dados de alta en el sistema.
RF.MQ.1. El sistema permitirá consultar las diferentes máquinas dados de alta en el sistema de una visión.
Descripción Caso de Uso
El programa mostrará, en formato similar a una tabla, la información pertinente de las máquinas disponibles en el sistema. Se debe de permitir obtener una vista detalles.
Precondiciones
Iniciar sesión, seleccionar una visión
Postcondiciones
Ninguna.
Actores
Observer User
Descripción
1. El sistema mostrará toda la información, de forma tabular, de las visiones disponibles en el sistema.
Variaciones

Ninguna.

Tabla 14. Caso de uso: Consulta máquinas

Caso de Uso
CU.11. Añadir máquina
Requisitos funcionales que satisface
RF.MQ.2. El sistema permitirá dar de alta nuevas máquinas.
Descripción Caso de Uso
El programa permitirá añadir una nueva máquina al sistema. Se pedirá toda la información necesaria para generar una nueva entidad. Se permitirá a un usuario que tenga el permiso de edición – <i>User</i> .
Precondiciones
Iniciar sesión, seleccionar una visión
Postcondiciones
La entidad es guardada en el sistema.
Actores
User
Descripción
<ol style="list-style-type: none">1. El usuario introducirá los campos necesarios.2. El usuario presiona el botón de crear.
Variaciones
<ul style="list-style-type: none">▪ <i>Escenario 1:</i> El usuario no introduce alguno de los campos.<ul style="list-style-type: none">○ El sistema mostrará un mensaje de error.▪ <i>Escenario 2:</i> El usuario desea cancelar la operación.<ul style="list-style-type: none">○ El usuario presiona el botón de cancelar.

Tabla 15. Caso de uso: Añadir máquina

Caso de Uso
CU.12. Modificar máquina
Requisitos funcionales que satisface
RF.MQ.3. El sistema permitirá modificar datos de las maquinas del sistema.
Descripción Caso de Uso
El programa permitirá editar una máquina existente en el sistema. De pedirá la información que no implique la unidad de la entidad. Se permitirá a un usuario que tenga el permiso de edición – <i>User</i> .
Precondiciones
Iniciar sesión, seleccionar una visión
Postcondiciones

La entidad es modificada en el sistema.
Actores
User
Descripción
<ol style="list-style-type: none"> 1. El usuario activará el modo edición pulsando en el botón editar. 2. El usuario introducirá los cambios que considere oportunos. 3. El usuario presiona el botón de guardar cambios.
Variaciones
<ul style="list-style-type: none"> ▪ <i>Escenario 1:</i> El usuario no introduce alguno de los campos. <ul style="list-style-type: none"> ○ El sistema mostrará un mensaje de error.

Tabla 16. Caso de uso: Modificar máquina

Caso de Uso
CU.13. Clonar máquina
Requisitos funcionales que satisface
RF.MQ.4. El sistema permitirá dar clonar maquinas ya existentes.
Descripción Caso de Uso
El programa permitirá clonar una máquina junto a todos los elementos que tengan relación con él. Se permitirá modificar la identidad de la entidad. Se permitirá a un usuario que tenga el permiso de edición – <i>User</i> .
Precondiciones
Iniciar sesión, existencia de una entidad original
Postcondiciones
La entidad clonada es guardada en el sistema.
Actores
User
Descripción
<ol style="list-style-type: none"> 1. El usuario presionará en el botón de clonación de una entidad original. 2. El usuario introducirá los nuevos valores de identificación 3. El usuario presiona el botón de clonar.
Variaciones
<ul style="list-style-type: none"> ▪ <i>Escenario 1:</i> El usuario no modifica alguno de los campos. <ul style="list-style-type: none"> ○ El sistema mostrará un mensaje de error. ▪ <i>Escenario 2:</i> El usuario desea cancelar la operación. <ul style="list-style-type: none"> ○ El usuario presiona el botón de cancelar.

Tabla 17. Caso de uso: Clonar máquina

Gestión de FactoryModel

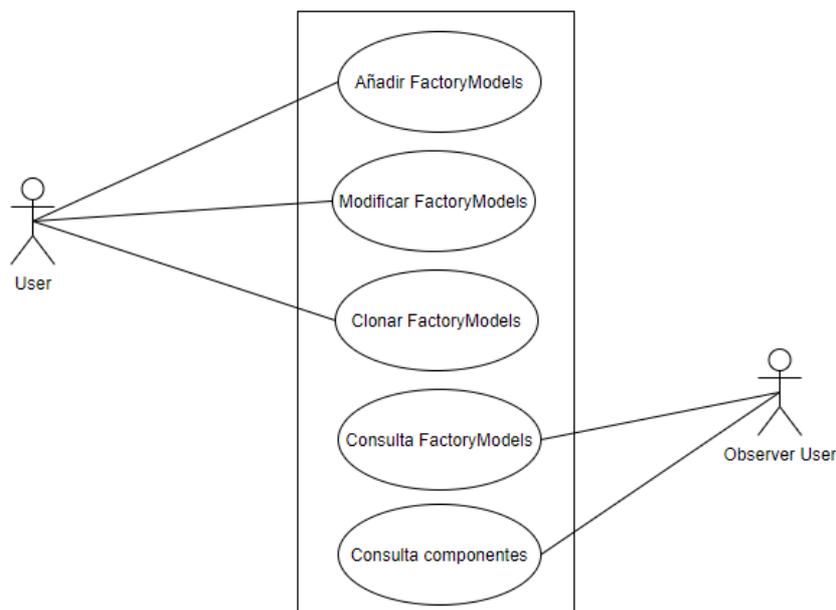


Fig 25. Diagrama de casos de uso: Gestión de FactoryModel

Caso de Uso
CU.14. Consulta FactoryModels
Requisitos funcionales que satisface
<p>RF.VS.5. El sistema permitirá consultar las diferentes visiones dados de alta en el sistema.</p> <p>RF.MQ.1. El sistema permitirá consultar las diferentes máquinas dados de alta en el sistema de una visión.</p> <p>RF.FM.1. El sistema permitirá consultar las diferentes <i>FactoryModels</i> dados de alta en el sistema de una MAQUINA.</p>
Descripción Caso de Uso
El programa mostrará, en formato similar a una tabla, la información pertinente de las visiones disponibles en el sistema. Se debe de permitir obtener una vista detalles.
Precondiciones
Iniciar sesión, seleccionar una máquina
Postcondiciones
Ninguna.
Actores
Observer User
Descripción



1. El sistema mostrará toda la información, de forma tabular, de las visiones disponibles en el sistema.

Variaciones

Ninguna.

Tabla 18. Caso de uso: Consulta FactoryModels

Caso de Uso
CU.15. Consulta componentes
Requisitos funcionales que satisface
RF.FMC.1. El sistema permitirá seleccionar los componentes activados en el sistema. RF.FMC.2. El sistema permitirá consultar los componentes activados en el sistema.
Descripción Caso de Uso
El programa mostrará, en formato similar a una tabla, los componentes que dispone un FactoryModel, mostrando la información relativa a estado del servicio y ahorros potenciales.
Precondiciones
Iniciar sesión, seleccionar un FactoryModel
Postcondiciones
Ninguna.
Actores
Observer User
Descripción
1. El sistema mostrará toda la información relativa a los componentes disponibles.
Variaciones
Ninguna.

Tabla 19. Caso de uso: Consulta componentes

Caso de Uso
CU.16. Añadir FactoryModels
Requisitos funcionales que satisface
RF.FM.2. El sistema permitirá dar de alta nuevos FactoryModels.
Descripción Caso de Uso
El programa permitirá añadir un nuevo FactoryModel al sistema. Se pedirá toda la información necesaria para generar una nueva entidad. Se permitirá a un usuario que tenga el permiso de edición – <i>User</i> .
Precondiciones
Iniciar sesión, seleccionar una máquina
Postcondiciones

La entidad es guardada en el sistema.
Actores
User
Descripción
<ol style="list-style-type: none"> 1. El usuario introducirá los campos necesarios. 2. El usuario presiona el botón de crear.
Variaciones
<ul style="list-style-type: none"> ▪ <i>Escenario 1:</i> El usuario no introduce alguno de los campos. <ul style="list-style-type: none"> ○ El sistema mostrará un mensaje de error. ▪ <i>Escenario 2:</i> El usuario desea cancelar la operación. <ul style="list-style-type: none"> ○ El usuario presiona el botón de cancelar.

Tabla 20. Caso de uso: Añadir FactoryModels

Caso de Uso
CU.17. Modificar FactoryModels
Requisitos funcionales que satisface
RF.FM.3. El sistema permitirá modificar datos de los <i>FactoryModel</i> del sistema.
Descripción Caso de Uso
El programa permitirá editar un <i>FactoryModel</i> existente en el sistema. De pedirá la información que no implique la unidad de la entidad. Se permitirá a un usuario que tenga el permiso de edición – <i>User</i> .
Precondiciones
Iniciar sesión, seleccionar una máquina
Postcondiciones
La entidad es modificada en el sistema.
Actores
User
Descripción
<ol style="list-style-type: none"> 1. El usuario activará el modo edición pulsando en el botón editar. 2. El usuario introducirá los cambios que considere oportunos. 3. El usuario presiona el botón de guardar cambios.
Variaciones
<ul style="list-style-type: none"> ▪ <i>Escenario 1:</i> El usuario no introduce alguno de los campos. <ul style="list-style-type: none"> ○ El sistema mostrará un mensaje de error.

Tabla 21. Caso de uso: Modificar FactoryModels

Caso de Uso
CU.18. Clonar FactoryModels
Requisitos funcionales que satisface

RF.FM.4. El sistema permitirá dar clonar <i>FactoryModel</i> ya existentes.
Descripción Caso de Uso
El programa permitirá clonar un <i>FactoryModel</i> junto a todos los elementos que tengan relación con él. Se permitirá modificar la identidad de la entidad. Se permitirá a un usuario que tenga el permiso de edición – <i>User</i> .
Precondiciones
Iniciar sesión, existencia de una entidad original
Postcondiciones
La entidad clonada es guardada en el sistema.
Actores
User
Descripción
<ol style="list-style-type: none"> 1. El usuario presionará en el botón de clonación de una entidad original. 2. El usuario introducirá los nuevos valores de identificación 3. El usuario presiona el botón de clonar.
Variaciones
<ul style="list-style-type: none"> ▪ <i>Escenario 1:</i> El usuario no modifica alguno de los campos. <ul style="list-style-type: none"> ○ El sistema mostrará un mensaje de error. ▪ <i>Escenario 2:</i> El usuario desea cancelar la operación. <ul style="list-style-type: none"> ○ El usuario presiona el botón de cancelar.

Tabla 22. Caso de uso: Clonar FactoryModels

Costes

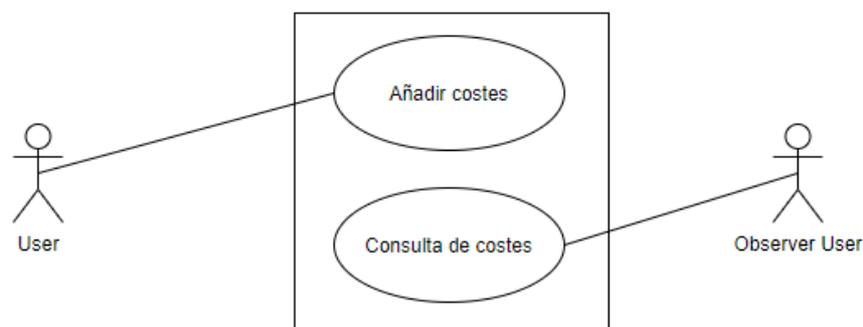


Fig 26. Diagrama de casos de uso: Gestión de Costes

Caso de Uso
CU.19. Añadir costes
Requisitos funcionales que satisface



RF.CS.1. El sistema permitirá consultar y modificar el componente de costes.

Descripción Caso de Uso
El programa permitirá añadir costes al sistema. Se permitirá añadir: <ul style="list-style-type: none"> – Costes fijos – Costes variables – Coste de mano de obra Es necesario disponer de permisos de edición – <i>User</i> .
Precondiciones
Iniciar sesión, selección de un FactoryModel
Postcondiciones
La entidad es guardada en el sistema.
Actores
User
Descripción
<ol style="list-style-type: none"> 1. El usuario introducirá los campos necesarios. 2. El usuario presiona el botón de guardar cambios.
Variaciones
Ninguna.

Tabla 23. Caso de uso: Añadir costes

Caso de Uso
CU.20. Consulta de costes
Requisitos funcionales que satisface
RF.CS.1. El sistema permitirá consultar y modificar el componente de costes.
Descripción Caso de Uso
El programa permitirá consultar los costes que tiene un FactoryModel, permitiendo obtener los cálculos completos que se encuentran en el sistema.
Precondiciones
Iniciar sesión, selección de FactoryModel
Postcondiciones
Ninguna.
Actores
Observer User
Descripción
<ol style="list-style-type: none"> 1. El sistema mostrará los costes de mano de obra. 2. El sistema mostrará los costes fijos. 3. El sistema mostrará los costes variables. 4. El sistema mostrará los ahorros potenciales arrojados.

5. El sistema mostrará el comentario (en caso de existir).

Variaciones

Ninguna.

Tabla 24. Caso de uso: Consulta de costes

Energía

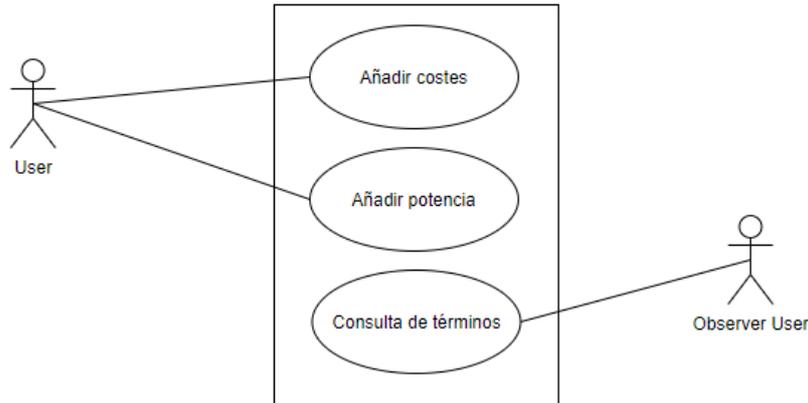


Fig 27. Diagrama de casos de uso: Gestión de Energía

Caso de Uso
CU.21. Añadir potencia
Requisitos funcionales que satisface
RF.ES.1. El sistema permitirá consultar y modificar el componente de energía.
Descripción Caso de Uso
El programa permitirá añadir el termino de potencia al sistema. Se permitirá añadir: <ul style="list-style-type: none"> – Tarifa – Potencias contratadas Es necesario disponer de permisos de edición – <i>User</i> .
Precondiciones
Iniciar sesión, selección de un FactoryModel
Postcondiciones
La entidad es guardada en el sistema.
Actores
User
Descripción
<ol style="list-style-type: none"> 1. El usuario introducirá los campos necesarios. 2. El usuario presiona el botón de guardar cambios.



Variaciones

Ninguna.

Tabla 25. Caso de uso: Añadir potencia

Caso de Uso
CU.22. Añadir consumo
Requisitos funcionales que satisface
RF.ES.1. El sistema permitirá consultar y modificar el componente de energía.
Descripción Caso de Uso
El programa permitirá añadir el termino de consumo al sistema. Se permitirá añadir: <ul style="list-style-type: none">– Tarifa– Discriminación horaria– Consumo anual Es necesario disponer de permisos de edición – <i>User</i> .
Precondiciones
Iniciar sesión, selección de un FactoryModel
Postcondiciones
La entidad es guardada en el sistema.
Actores
User
Descripción
<ol style="list-style-type: none">1. El usuario introducirá los campos necesarios.2. El usuario presiona el botón de guardar cambios.
Variaciones
Ninguna.

Tabla 26. Caso de uso: Añadir consumo

Caso de Uso
CU.23. Consulta de términos
Requisitos funcionales que satisface
RF.ES.1. El sistema permitirá consultar y modificar el componente de energía.
Descripción Caso de Uso
El programa permitirá consultar los datos almacenados en términos de potencia y consumo que tiene un FactoryModel, permitiendo obtener los cálculos completos que se encuentran en el sistema.
Precondiciones
Iniciar sesión, selección de FactoryModel
Postcondiciones

Ninguna.
Actores
Observer User
Descripción
<ol style="list-style-type: none"> 1. El sistema mostrará un resumen de los cálculos. 2. El sistema mostrará los términos de potencia. 3. El sistema mostrará los términos de consumo 4. El sistema mostrará los ahorros potenciales arrojados. 5. El sistema mostrará el comentario (en caso de existir).
Variaciones
Ninguna.

Tabla 27. Caso de uso: Consulta de términos

Producción

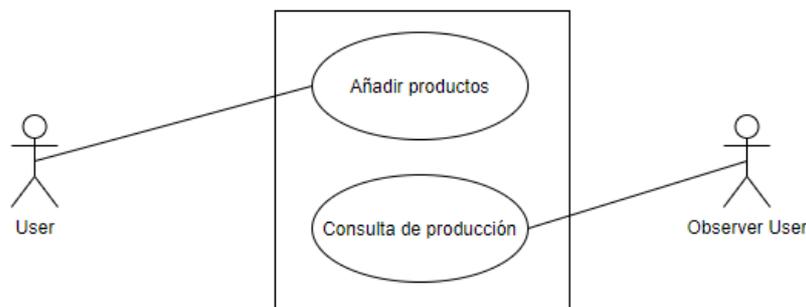


Fig 28. Diagrama de casos de uso: Gestión de Producción

Caso de Uso
CU.24. Añadir productos
Requisitos funcionales que satisface
RF.PS.1. El sistema permitirá consultar y modificar el componente de producción.
Descripción Caso de Uso
El programa permitirá añadir los costes al sistema. Se permitirá añadir: <ul style="list-style-type: none"> – Productos que tienen relevancia en la productividad. Es necesario disponer de permisos de edición – <i>User</i> .
Precondiciones
Iniciar sesión, selección de un FactoryModel
Postcondiciones
La entidad es guardada en el sistema.
Actores



User
Descripción
<ol style="list-style-type: none"> 1. El usuario introducirá los campos necesarios. 2. El usuario presiona el botón de guardar cambios.
Variaciones
Ninguna.

Tabla 28. Caso de uso: Añadir productos

Caso de Uso
CU.25. Consulta de producción
Requisitos funcionales que satisface
RF.PS.1. El sistema permitirá consultar y modificar el componente de producción.
Descripción Caso de Uso
El programa permitirá consultar los productos que tiene un FactoryModel, permitiendo obtener los cálculos completos que se encuentran en el sistema.
Precondiciones
Iniciar sesión, selección de FactoryModel
Postcondiciones
Ninguna.
Actores
Observer User
Descripción
<ol style="list-style-type: none"> 1. El sistema mostrará los productos disponibles 2. El sistema mostrará los ahorros potenciales arrojados. 3. El sistema mostrará el comentario (en caso de existir).
Variaciones
Ninguna.

Tabla 29. Caso de uso: Consulta de producción

Calidad

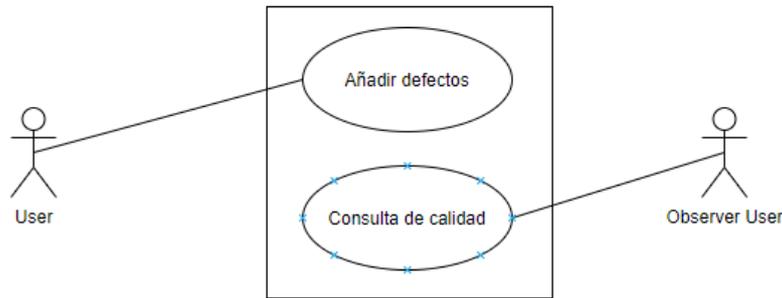


Fig 29. Diagrama de casos de uso: Gestión de Calidad

Caso de Uso
CU.26. Añadir defectos
Requisitos funcionales que satisface
RF.QS.1. El sistema permitirá consultar y modificar el componente de calidad.
Descripción Caso de Uso
El programa permitirá modificar los defectos de un producto que tiene almacenados en el sistema. Se permitirá añadir: <ul style="list-style-type: none"> Defectos vinculados a un producto Es necesario disponer de permisos de edición – <i>User</i> .
Precondiciones
Iniciar sesión, selección de un FactoryModel
Postcondiciones
La entidad es guardada en el sistema.
Actores
User
Descripción
<ol style="list-style-type: none"> El usuario introducirá los campos necesarios. El usuario presiona el botón de guardar cambios.
Variaciones
Ninguna.

Tabla 30. Caso de uso: Añadir defectos

Caso de Uso
CU.27. Consulta de calidad
Requisitos funcionales que satisface
RF.QS.1. El sistema permitirá consultar y modificar el componente de calidad.



Descripción Caso de Uso
El programa permitirá consultar los defectos vinculados a un producto que tiene un FactoryModel, permitiendo obtener los cálculos completos que se encuentran en el sistema.
Precondiciones
Iniciar sesión, selección de FactoryModel
Postcondiciones
Ninguna.
Actores
Observer User
Descripción
<ol style="list-style-type: none">1. El sistema mostrará la lista de productos resultado de Producción.2. El usuario escogerá un producto.3. El sistema mostrará los defectos de ese producto.4. El sistema mostrará el comentario (en caso de existir).
Variaciones
Ninguna.

Tabla 31. Caso de uso: Consulta de calidad

ASI 5: ANÁLISIS DE CLASES

El objetivo que se busca en este apartado es alcanzar una primera versión de las clases que van a tener los subsistemas anteriormente descritos. Se detallará el modelo de dominio con los atributos necesarios para el correcto funcionamiento del sistema.

Descripción del Modelo de Dominio

El modelo de dominio que se muestra en la [Fig. 30] nos representa todas las entidades que tendrá el sistema que se desea desarrollar. Se puede distinguir dos partes independientes que no tienen relación, la gestión de usuarios y el sistema en sí mismo.

Primeramente, vamos a describir la **gestión de usuarios**. El modelo dispone de dos entidades, usuario y roles. Un usuario es toda persona que tiene acceso a la aplicación con necesariamente un rol. Un rol es el permiso que va a disponer el usuario. Un usuario puede tener varios roles, que le permitirán realizar diferentes operaciones (lectura, escritura o administrador). En todo caso, un usuario debe de disponer de al menos de un rol de lectura. En un futuro, esta parte podría ser eliminada del modelo y ser sustituido por el sistema que disponga de autenticación del cliente.

Por otra parte, tenemos el **modelo del sistema**. Inicialmente, disponemos de una visión que almacenará una lista de máquinas en su interior que permitirán obtener la vista, en resumen. Una máquina, por su significado, solo puede pertenecer a una visión. La máquina, pertenecerá a una agrupación que almacenará las distintas máquinas que podremos tener un lugar físico. A su vez, una máquina podrá contener en su interior *FactoryModels*. Un *FactoryModel* será único por máquina no pudiendo pertenecer a varias máquinas, éste tendrá un estado que determinará la situación actual. Para alcanzar un sistema divisible, los cálculos vienen dados por componentes almacenados como identidad del *FactoryModel*. Es decir, es obligatorio que un *FactoryModel* genere X componentes como el sistema disponga en su fase de creación. Para alcanzar la abstracción necesaria entre el *FactoryModel* y sus componentes, se hará uso de una interfaz/clase abstracta que almacena los datos mínimos identificativos delegando a los diferentes componentes los cálculos pertinentes. Finalmente, se comentará el modelo de dominio a nivel de componentes:

- **Costes:** Disponemos del componente de costes que contiene cero o muchos costes variables y/o fijos. Estos costes tienen aspectos comunes, por lo que se usa una interfaz/clase.
- **Energía:** Disponemos del componente de energía que contiene al menos un periodo de discriminación horaria. A su vez, este contiene el coste y los valores necesarios que tiene cada periodo.
- **Producción/Calidad:** Disponemos del componente de producción que contiene cero o más productos que se producen o se manufacturan. A su vez, un producto puede contener cero o muchos defectos detectados.

Diagrama de Clases

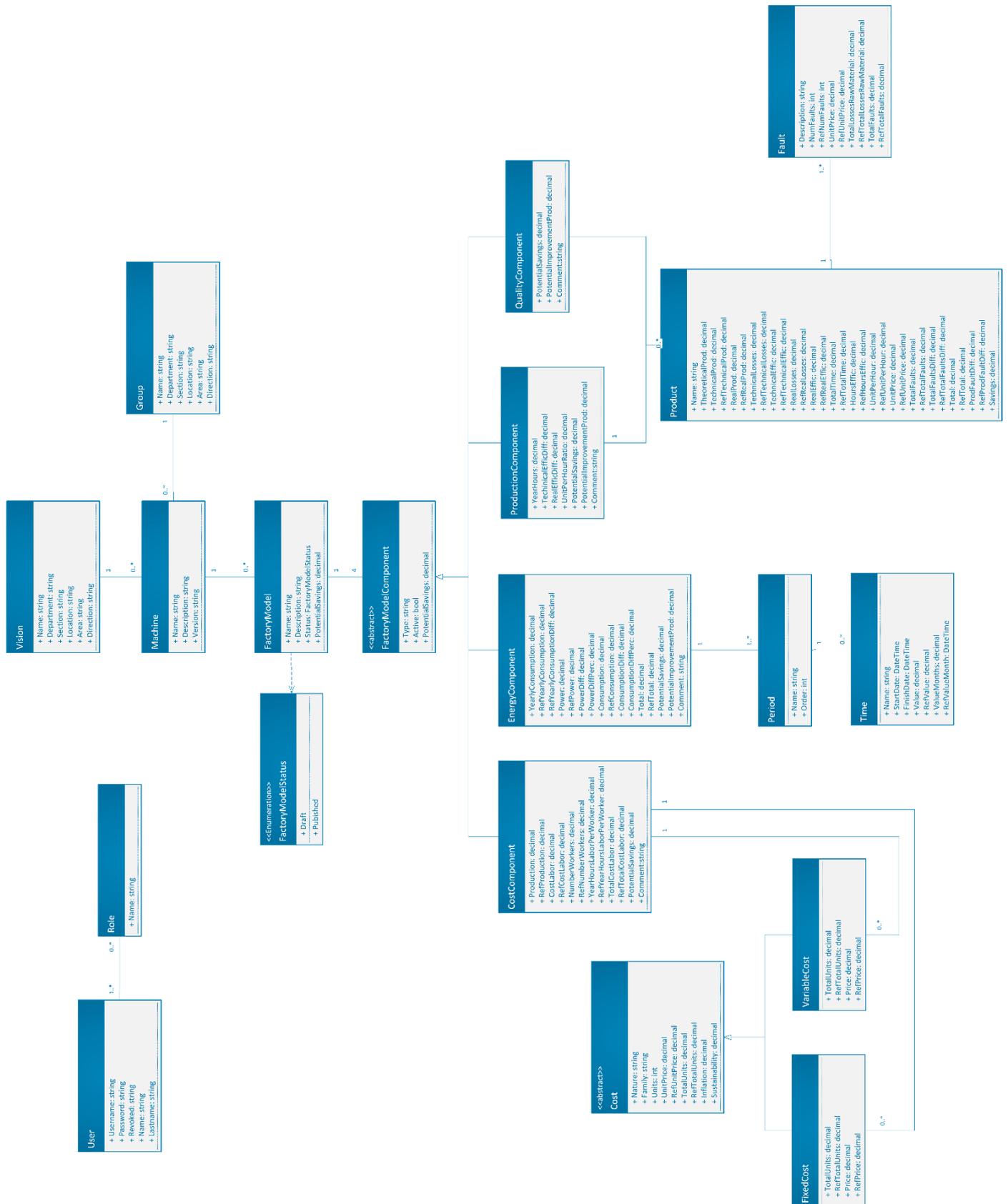


Fig 30. Análisis de clases: Diagrama de clases

Descripción de las Clases

A continuación, se realizará una breve descripción de las diferentes clases expuestas en el diagrama anterior.

Clase		
User		
Descripción		
Entidad que representa al usuario que puede usar la aplicación.		
Atributos		
Nombre	Tipo	Descripción
Username	string	Identificar del usuario. Se corresponde con el correo electrónico.
Password	string	Contraseña que tiene el usuario en el sistema.
Revoked	bool	Estado del usuario. Activo o no activo.
Name	string	Nombre del usuario.
Lastname	string	Apellido/s del usuario.

Tabla 32. Descripción de clases: User

Clase		
Role		
Descripción		
Entidad que representa los roles que puede tener un usuario.		
Atributos		
Nombre	Tipo	Descripción
Name	string	Nombre del role.

Tabla 33. Descripción de clases: Role

Clase		
Vision		
Descripción		
Entidad que guarda los diferentes supuestos de aplicar las técnicas de Cost-Deployment.		
Atributos		
Nombre	Tipo	Descripción
Name	string	Nombre que tendrá la visión.
Department	string	Departamento dentro de la empresa donde se va a efectuar la visión.
Section	string	Sección dentro del área del lugar real donde se va a efectuar la visión.

Location	string	Localidad del lugar real donde se va a efectuar la visión.
Area	string	Área del lugar real donde se va a efectuar la visión.
Direction	string	Dirección del lugar real donde se va a efectuar la visión.

Tabla 34. Descripción de clases: *Vision*

Clase		
Machine		
Descripción		
Entidad que guarda las diferentes formas productivas que dispone una empresa a las que se les quiere aplicar técnicas de Cost-Deployment		
Atributos		
Nombre	Tipo	Descripción
Name	string	Nombre de la máquina a tratar.
Description	string	Descripción que tendrá la máquina.
Version	string	Versionado de la máquina.

Tabla 35. Descripción de clases: *Machine*

Clase		
Group		
Descripción		
Entidad que guarda sitios reales a los que dispone una máquina. Permite disponer de varias máquinas en un mismo sitio real.		
Atributos		
Nombre	Tipo	Descripción
Name	string	Nombre de la agrupación
Department	string	Departamento a la que pertenece la agrupación de máquinas.
Section	string	Sección dentro del área del lugar real que pertenece la agrupación de máquinas.
Location	string	Localidad del lugar real que pertenece la agrupación de máquinas.
Area	string	Área del lugar real que pertenece la agrupación de máquinas.
Direction	string	Dirección del lugar real que pertenece la agrupación de máquinas.

Tabla 36. Descripción de clases: *Group*

Clase		
FactoryModel		
Descripción		
Entidad que permite realizar las diferentes simulaciones de una máquina. Los componentes de esta entidad están dentro de la clave natural de la entidad.		
Atributos		
Nombre	Tipo	Descripción
Name	string	Nombre del FactoryModel
Description	string	Descripción que se le asigna al FactoryModel
Status	FactoyModelStatus	Estado en el que se encuentra el FactoryModel.
PotentialSavings	decimal	Cálculo final resultado de los diferentes componentes que lo componen.

Tabla 37. Descripción de clases: *FactoryModel*

Enumeración	
FactoryModelStatus	
Descripción	
Define los diferentes estados en los que puede estar un FactoryModel	
Valores	
Draft	
Published	

Tabla 38. Descripción de clases: *FactoryModelStatus*

Clase		
CostComponent		
Descripción		
Entidad que representa a un componente de costes. Este pertenece a un FactoryModel.		
Atributos		
Nombre	Tipo	Descripción
Production	decimal	Producción.
RefProduction	decimal	Producción ajustada.
CostLabor	decimal	Coste de mano de obra.
RefCostLabor	decimal	Coste de mano de obra ajustada.
NumberWorkers	decimal	Número de trabajadores.
RefNumberWorkers	decimal	Número de trabajadores ajustados.
YearHoursLaborPerWorker	decimal	Horas anuales de trabajadores.
RefYearHoursLaborPerWorker	decimal	Horas anuales de trabajadores ajustadas.
TotalCostLabor	decimal	Coste total de mano de obra.

RefTotalCostLabor	decimal	Coste ajustado de mano de obra.
PotentialSavings	decimal	Ahorros que produce el componente.
Comment	String	Comentario del componente.

Tabla 39. Descripción de clases: *CostComponent*

Clase abstracta		
Cost		
Descripción		
Entidad que representa a un coste de forma abstracta. Aporta las cualidades básicas.		
Atributos		
Nombre	Tipo	Descripción
Nature	string	Naturaleza.
Family	string	Familia.
Units	int	Unidades.
UnitPrice	decimal	Precio/Unidad
RefUnitPrice	decimal	Referencia ajustada de Precio/Unidad
TotalUnits	decimal	Total coste unidades.
RefTotalUnits	decimal	Total ajustado coste unidades.
Inflation	decimal	Infracción.
Sustainability	decimal	Sostenibilidad económica.

Tabla 40. Descripción de clases: *Cost*

Clase		
FixedCost		
Descripción		
Entidad que representa a un coste fijo. Hereda de la clase abstracta Cost.		
Atributos		
Nombre	Tipo	Descripción
TotalUnits	decimal	Total de unidades.
RefTotalUnits	decimal	Total ajustado de unidades.
Price	decimal	Precio del coste fijo.
RefPrice	decimal	Precio ajustado del coste fijo.

Tabla 41. Descripción de clases: *FixedCost*

Clase		
VariableCost		
Descripción		
Entidad que representa a un coste variable. Hereda de la clase abstracta Cost.		
Atributos		
Nombre	Tipo	Descripción

TotalUnits	decimal	Total de unidades.
RefTotalUnits	decimal	Total ajustado de unidades.
Price	decimal	Precio del coste variable.
RefPrice	decimal	Precio ajustado del coste variable.

Tabla 42. Descripción de clases: *VariableCost*

Clase		
EnergyComponent		
Descripción		
Entidad que representa a un componente de energía. Este pertenece a un FactoryModel.		
Atributos		
Nombre	Tipo	Descripción
YearlyConsumption	decimal	Consumo anual.
RefYearlyConsumption	decimal	Consumo anual ajustada.
RefYearlyConsumptionDiff	decimal	Diferencia de consumo.
Power	decimal	Potencia contratada.
RefPower	decimal	Potencia contratada ajustada.
PowerDiff	decimal	Diferencia de potencia.
PowerDiffPerc	decimal	Diferencia en porcentaje de potencia.
Consumption	decimal	Consumo.
RefConsumption	decimal	Consumo ajustado.
ConsumptionDiff	decimal	Diferencia de consumo.
Total	decimal	Total de potencia y consumo.
RefTotal	decimal	Total ajustado de potencia y consumo.
PotentialSavings	decimal	Ahorros que produce el componente.
PotentialImprovementProd	decimal	Ahorros que produce el componente en porcentaje.
Comment	String	Comentario del componente.

Tabla 43. Descripción de clases: *EnergyComponent*

Clase		
Period		
Descripción		
Entidad que representa a un periodo del componente de energía.		
Atributos		
Nombre	Tipo	Descripción
Name	string	Nombre del periodo.
Order	int	Orden del periodo.

Tabla 44. Descripción de clases: *Period*

Clase		
Time		
Descripción		
Entidad que representa a un periodo de tiempo del componente de energía.		
Atributos		
Nombre	Tipo	Descripción
Name	string	Nombre del periodo de tiempo.
StartDate	DateTime	Fecha de inicio del periodo de tiempo.
FinishDate	DateTime	Fecha de fin del periodo de tiempo.
Value	decimal	Valor que tiene el periodo de tiempo.
RefValue	decimal	Valor ajustado que tiene el periodo de tiempo.
ValueMonths	decimal	Número de meses.
RefValueMonths	DateTime	Número de meses ajustado.

Tabla 45. Descripción de clases: *Time*

Clase		
ProductionComponent		
Descripción		
Entidad que representa a un componente de producción. Este pertenece a un FactoryModel.		
Atributos		
Nombre	Tipo	Descripción
YearHours	decimal	Horas de producción.
TechnicalEfficDiff	decimal	Porcentaje de eficiencia técnica.
RealEfficDiff	decimal	Porcentaje de eficiencia real.
UnitPerHourRatio	decimal	Unidades por hora producidas.
PotentialSavings	decimal	Ahorros que produce el componente.
PotentialImprovementProd	decimal	Ahorros que produce el componente en porcentaje.
Comment	String	Comentario del componente.

Tabla 46. Descripción de clases: *ProductionComponent*

Clase		
QualityComponent		
Descripción		
Entidad que representa a un componente de calidad. Este pertenece a un FactoryModel.		
Atributos		
Nombre	Tipo	Descripción
PotentialSavings	decimal	Ahorros que produce el componente.

PotentialImprovementProd	decimal	Ahorros que produce el componente en porcentaje.
Comment	String	Comentario del componente.

Tabla 47. Descripción de clases: *QualityComponent*

Clase		
Product		
Descripción		
Entidad que representa a un producto de manufacturación. Es usado tanto por el componente Producción como por Calidad.		
Atributos		
Nombre	Tipo	Descripción
Name	string	Nombre del producto.
TheoreticalProd	decimal	Producción teórica.
TechnicalProd	decimal	Producción técnica.
RefTechnicalProd	decimal	Producción ajustada técnica.
RealProd	decimal	Producción real.
RefRealProd	decimal	Producción ajustada real.
TechnicalLosses	decimal	Pérdida técnica.
RefTechnicalLosses	decimal	Pérdida ajusta técnica.
TechnicalEffic	decimal	Eficiencia técnica.
RefTechnicalEffic	decimal	Eficiencia ajustada técnica.
RealLosses	decimal	Pérdidas reales.
RefRealLosses	decimal	Pérdidas ajustadas reales
RealEffic	decimal	Eficiencia real
RefRealEffic	decimal	Eficiencia ajustada real.
TotalTime	decimal	Horas totales
RefTotalTime	decimal	Horas totales ajustadas.
HoursEffic	decimal	Eficiencias
RefHoursEffic	decimal	
UnitPerHour	decimal	Unidades por horas.
RefUnitPerHour	decimal	Unidades por horas ajustadas.
UnitPrice	decimal	Precio/Unidad
RefUnitPrice	decimal	Precio/Unidad ajustado.
TotalFaults	decimal	Total de pérdidas por defectos
RefTotalFaults	decimal	Total ajustado de pérdidas por defectos
RefTotalFaultsDiff	decimal	Diferencia de totales de pérdidas por defectos.
Total	decimal	Total de producción.
RefTotal	decimal	Total ajustado de producción

ProdFaultDiff	decimal	Diferencia de calidad de producto.
RefProdFaultDiff	decimal	Diferencia ajustada de calidad del producto.
Savings	decimal	Ahorros por la calidad del producto.

Tabla 48. Descripción de clases: *Product*

Clase		
Fault		
Descripción		
Entidad que representa un defecto de manufacturación de un producto. Es usado por el componente por Calidad.		
Atributos		
Nombre	Tipo	Descripción
Description	string	Descripción del defecto.
NumFaults	decimal	Número de defectos.
RefNumFaults	decimal	Referencia ajusta de número de defectos.
UnitPrice	decimal	Precio/Unidad
RefUnitPrice	decimal	Ref Precio/Unidad
TotalLossesRawMaterial	decimal	Pérdidas en materia prima
RefTotalLossesRawMaterial	decimal	Pérdidas ajustadas en materia prima
TotalFaults	decimal	Coste de los defectos.
RefTotalFaults	decimal	Coste ajustado de los defectos.

Tabla 49. Descripción de clases: *Fault*

ASI 8: DEFINICIÓN DE INTERFACES DE USUARIO

En este apartado, se explicarán los conceptos necesarios en materia de interfaz de usuario. Dado la temática de la aplicación, así como la negativa del cliente para adaptar la aplicación a un móvil, se tratará solo de una interfaz compatible con pantallas de escritorio.

Será necesario hacer uso de colores sobrios que cumplan con el estándar de accesibilidad WCAG AAA. Se contemplarán colores en escala de grises y colores suaves que no superen 14 puntos de ratio de contraste.

Descripción de la Interfaz

En este apartado, se definirán todas las pantallas que formarán parte del producto. Son conceptos en los que se basarán las diferentes pantallas finales.

Inicio de sesión

Esta pantalla permitirá acceder al usuario a la zona privada de la aplicación.

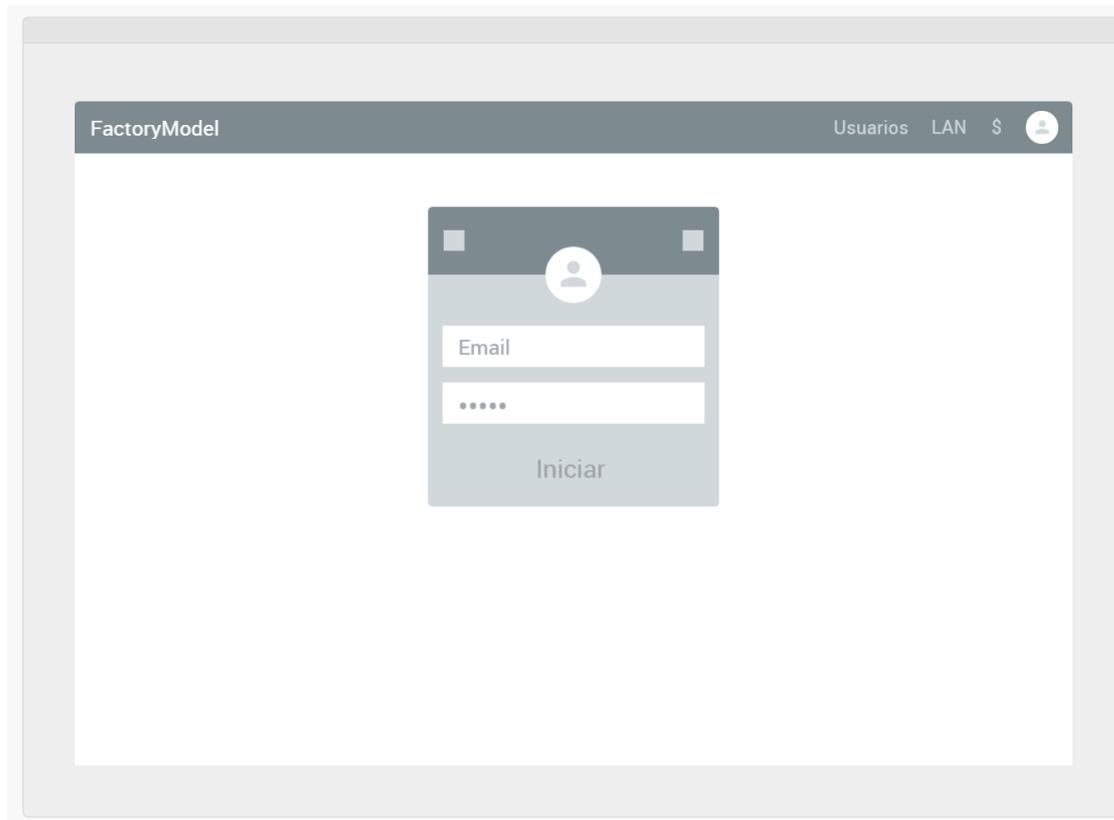


Fig 31. Prototipo: Inicio de sesión

Listado de usuarios

Esta pantalla permitirá a un usuario con permisos de administrador crear o modificar los usuarios que se encuentran disponibles en el sistema.

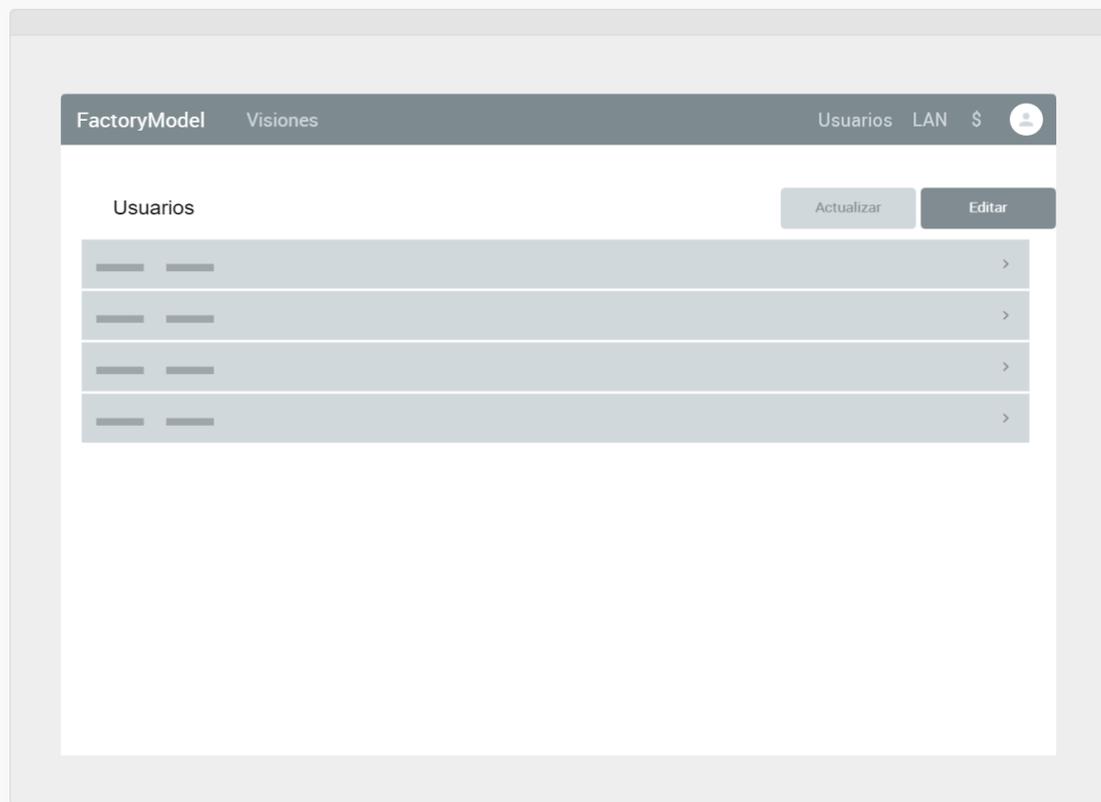


Fig 32. Prototipo: Listado de usuarios

Dialogo de creación de nueva entidad

Dialogo común para todas las entidades usado para añadir una nueva entidad al sistema. Mostrará los errores necesarios para su correcto funcionamiento. Una vez completado el formulario, podrá crear la entidad. En todo momento se podrá cancelar el cuadro de dialogo pulsando en el botón de Cerrar.

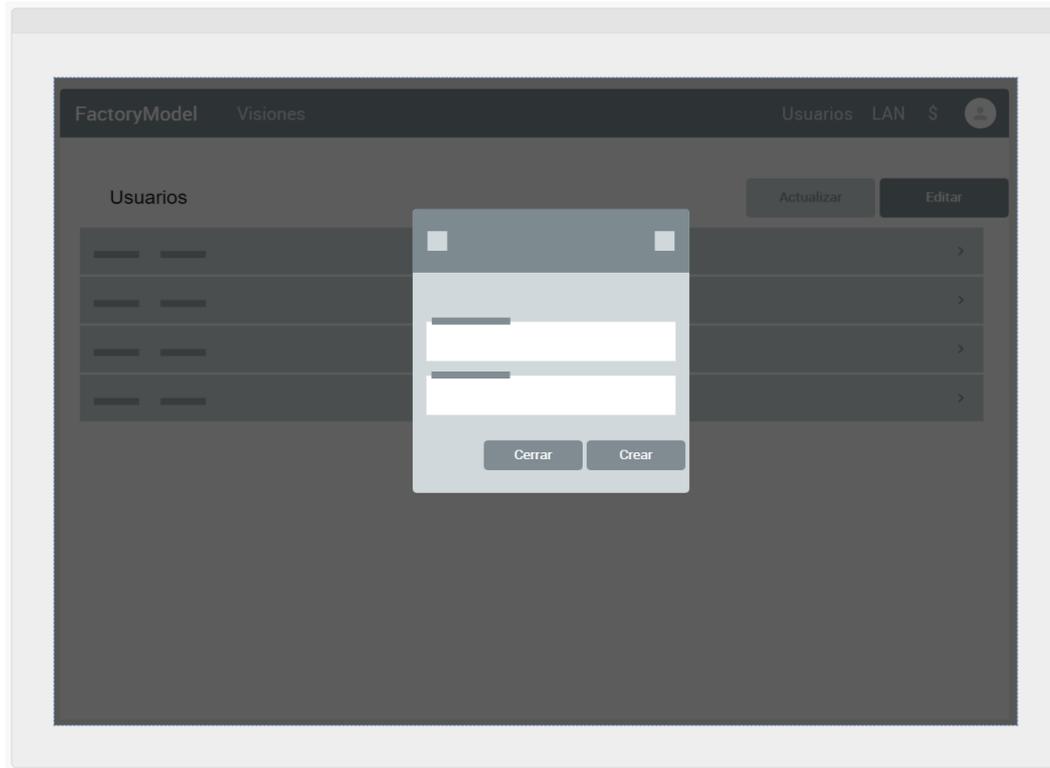


Fig 33. Prototipo: Dialogo de creación

Dialogo de clonación

Dialogo común para todas las entidades usado para clonar una entidad del sistema compatible. Mostrará los errores pertinentes. Una vez completado el formulado, podrá clonar la entidad. En todo momento se podrá cancelar el cuadro de dialogo pulsando en el botón de Cerrar.

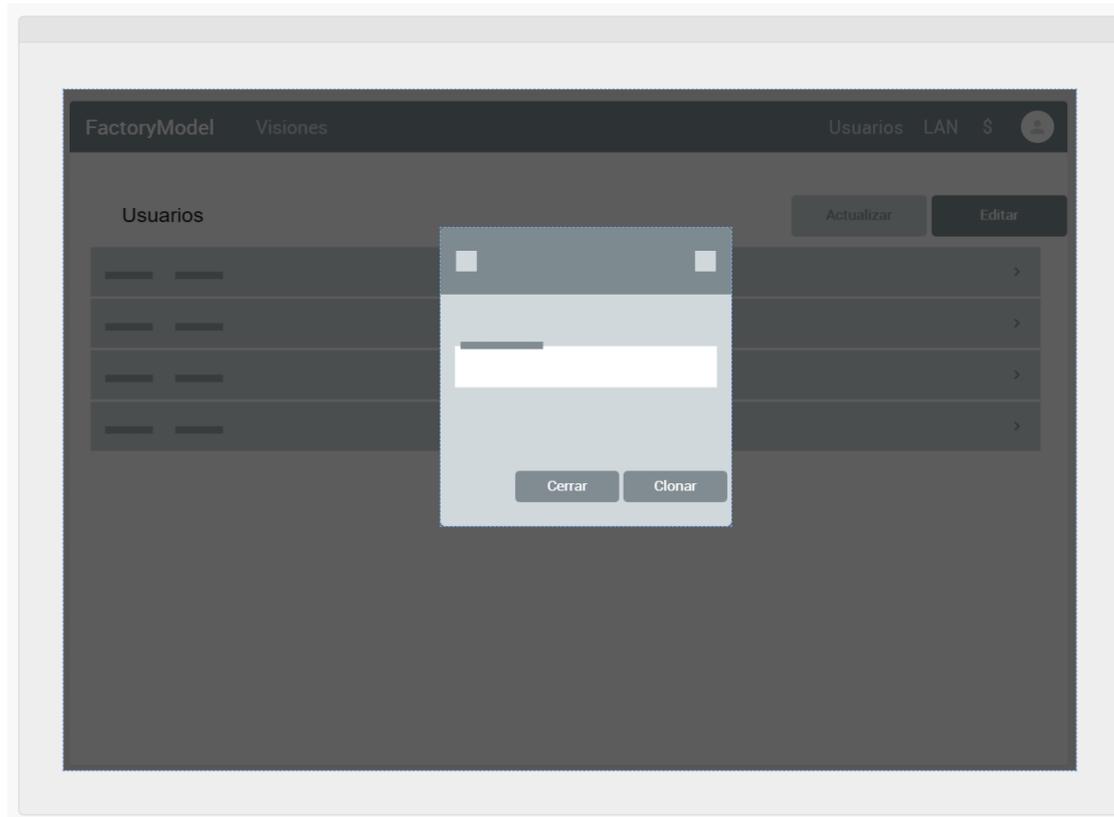


Fig 34. Prototipo: Dialogo de clonación

Listado de visiones

Listado que permite visualizar las visiones que se encuentran en el sistema. Permitirá añadir nuevas visiones pulsando el botón de Añadir. Se podrá editar la información pulsando en Editar y posteriormente a Guardar Cambios; y se podrá clonar una entidad ya creada. Si se pulsa encima de la entidad, el sistema permitirá ir a la vista de detalles de la visión.

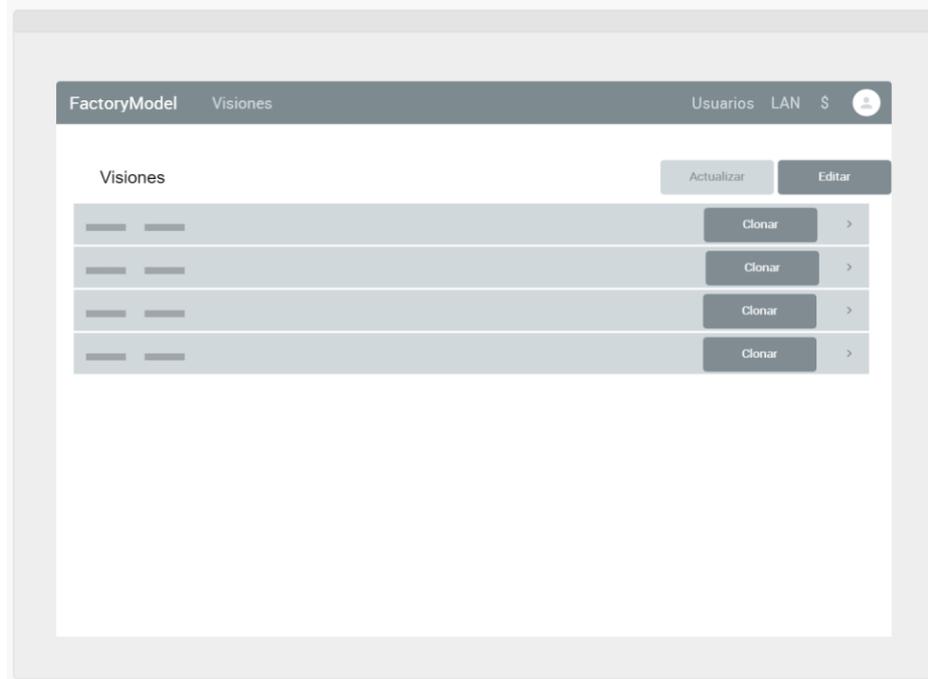


Fig 35. Prototipo: Listado de visiones

Vista de detalle de una visión

Vista que permite conocer las máquinas que tiene una visión. Se cargará un listado de máquinas que permitirá realizar las tareas de añadir, editar y clonar una máquina. Una vez seleccionada una máquina, el sistema cargará la información de los *FactoryModels*. A su vez, permitirá añadir, editar y clonar un *FactoryModel*. Cuando se seleccione un *FactoryModel*, se pasará a la vista en detalles de un *FactoryModel*.

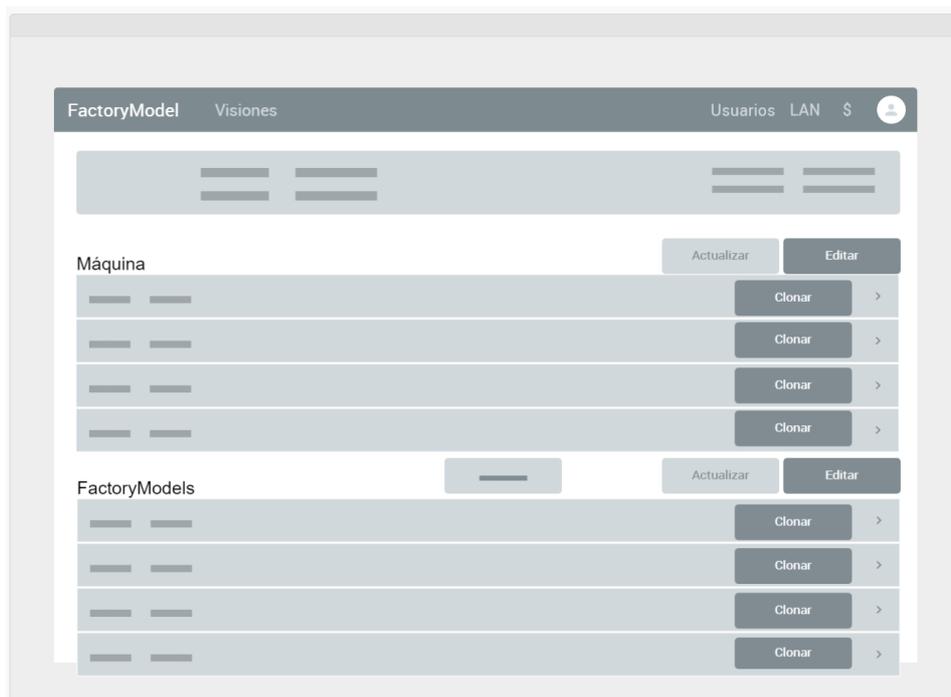


Fig 36. Prototipo: Vista detalles visión

Listado de componentes

Vista en detalles de un FactoryModel que permite conocer los componentes de los que dispone. Permite conocer los detalles del FactoryModel, activar o desactivar un componente y visualizar la información completa de un componente. A su vez, permite Publicar un FM.

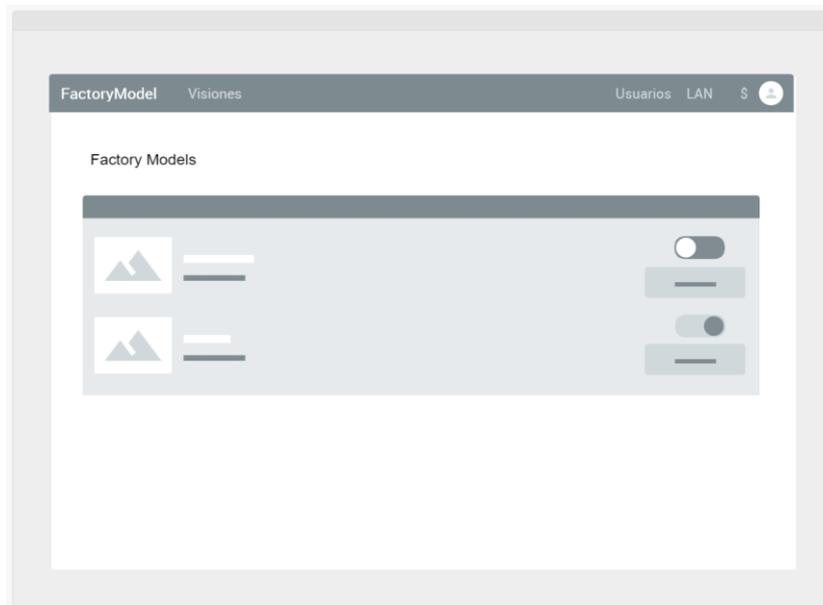


Fig 37. Prototipo: Listado componentes

Componente Costes

Vista en detalles del componente Costes que nos permitirá conocer los detalles que tiene. Dispondrá de dos cajas superiores con la información relativa a mano de obra y resumen. Dispondrá, también, de dos tablas para almacenar los Costes Fijos y Variables. Finalmente, dispondrá de un cuadro de texto para posibles comentarios. Se podrá editar mientras el FactoryModel este activo.

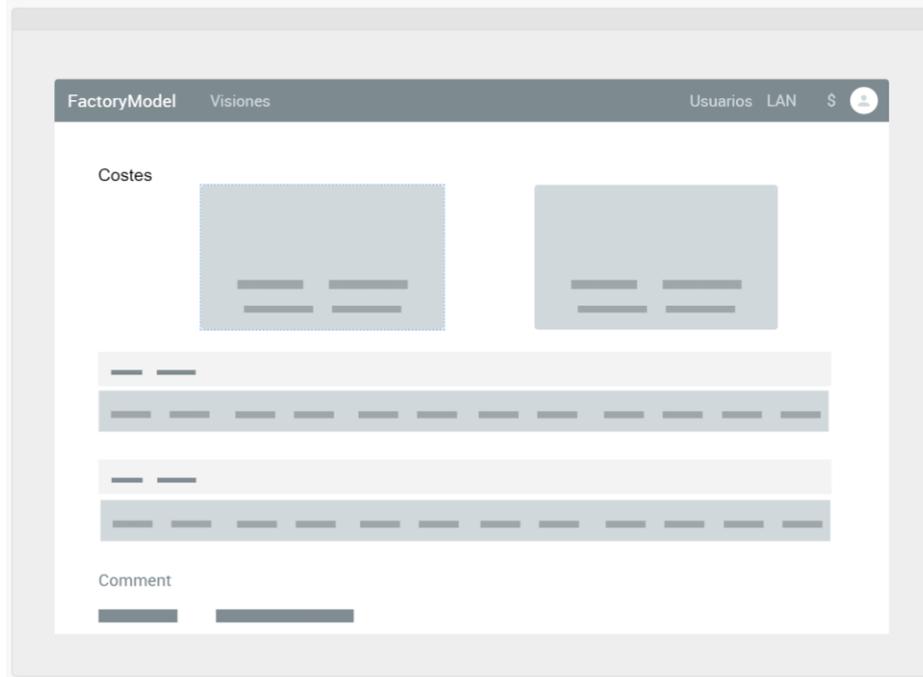


Fig 38. Prototipo: Componente Costes

Componente Energía

Vista en detalles del componente Energía que nos permitirá conocer los detalles que tiene. Dispondrá de dos cajas superiores con un resumen y los ahorros potenciales. A mayores, dispondrá de dos apartados: Término de potencia y Término de energía. Finalmente, dispondrá de un cuadro de texto para posibles comentarios. Se podrá editar mientras el *FactoryModel* este activo.

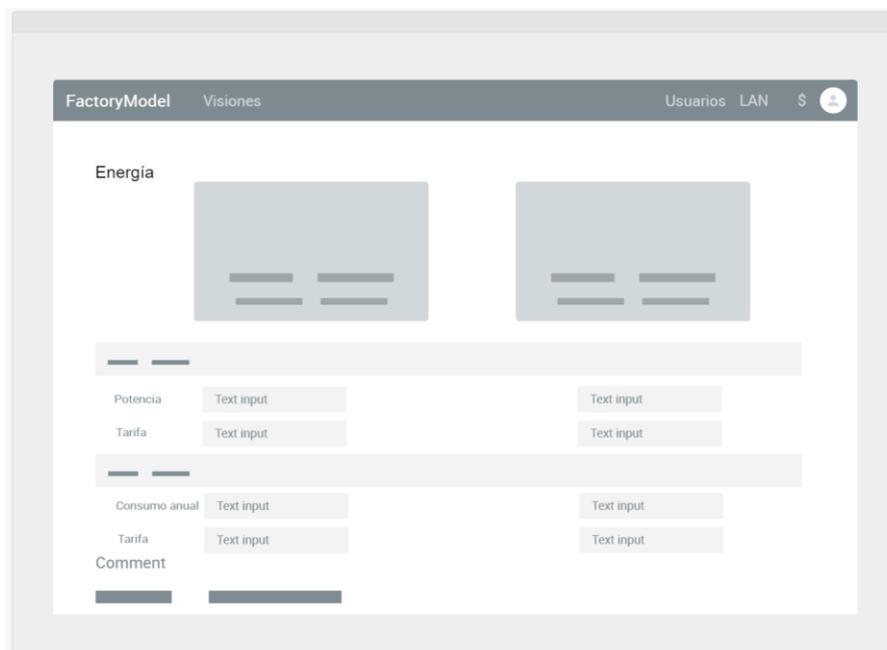


Fig 39. Prototipo: Energía

Componente Producción

Vista en detalles del componente Producción que nos permitirá conocer los detalles que tiene. Dispondrá de dos cajas superiores con las variaciones en producción pertinentes y los ahorros potenciales. A mayores, dispondrá de dos tablas: Referencia y Referencia Ajustada almacenando los productos dados. Finalmente, dispondrá de un cuadro de texto para posibles comentarios. Se podrá editar mientras el FactoryModel este activo.

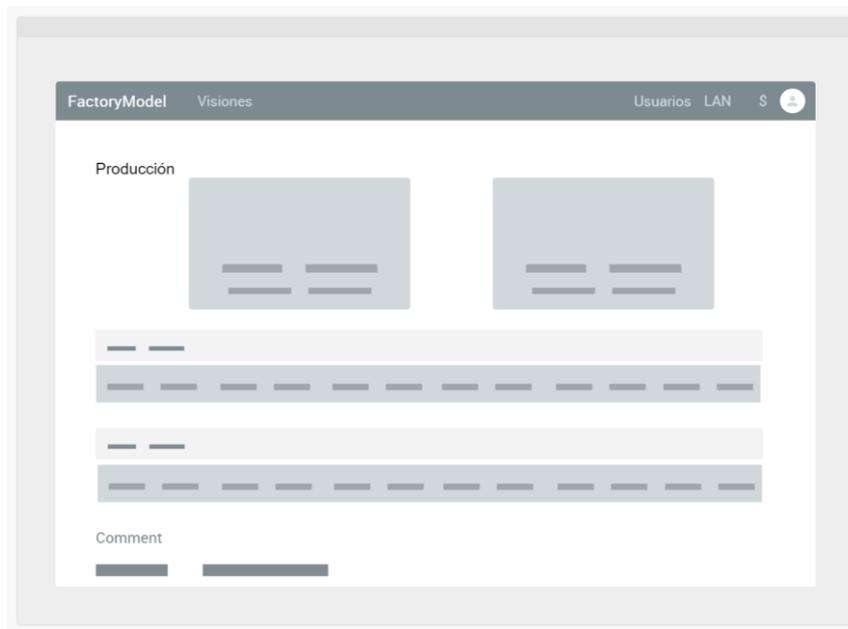


Fig 40. Prototipo: Producción

Componente Calidad

Vista en detalles del componente Calidad que nos permitirá conocer los detalles que tiene. Dispondrá de dos cajas superiores con un buscador de productos resultado de Producción y los ahorros potenciales. A mayores, dispondrá de dos tablas: Listado de productos en modo lectura y un listado defectos del producto seleccionado en primera instancia. Finalmente, dispondrá de un cuadro de texto para posibles comentarios. Se podrá editar mientras el FactoryModel este activo.

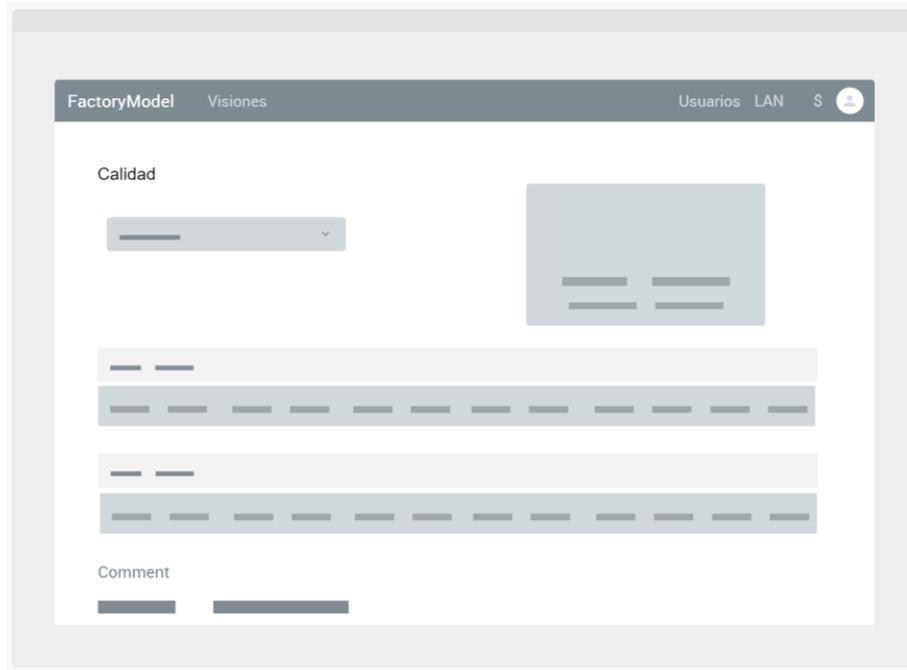


Fig 41. Prototipo: Calidad

Definición de los componentes del sistema

La interfaz gráfica sobre la que sitúa el sistema debe estar sustentada por pantallas diseñadas y descritas anteriormente. Para su correcta puesta en funcionamiento, se debe de permitir la reutilización de componentes. Gracias al uso de los *framework* actuales de JavaScript, esta tarea se vuelve muy sencilla.

Se pasa a comentar los componentes por los que estará formado la aplicación:

- **Barra de navegación:** Barra superior con la información necesario en todo momento para moverse por la aplicación. Los enlaces que dispondrá serán los siguientes:
 - **Inicio:** Acceso a la página principal de la aplicación.
 - [*Usuario que ha iniciado sesión*] **Visiones:** Acceso al listado de visiones disponibles en el sistema.
 - [*Usuario administrador*] **Usuarios:** Administración de los usuarios disponibles en el sistema.
 - **Lenguaje:** Permitir adaptar la aplicación a un idioma determinado.
 - **Moneda:** Permite seleccionar la moneda con la que queremos realizar los cálculos.
 - [*Usuario que ha iniciado sesión*] **Perfil:** Se corresponde con un submenú que dispone:
 - Perfil: Acceso al perfil personal del usuario.
 - Cerrar sesión: Permite cerrar la sesión.
- **Breadcrumb:** Barra localizada en la parte inferior de la barra de navegación que permite en conocer el recorrido seguido en la aplicación pudiendo regresar a una página anteriormente visitada.



- **Tablas:** Permiten mostrar de forma tabular la información del sistema. Debe de permitir recargar los datos de forma automática.
- **Cuadros de texto:** Permiten la interacción del usuario con la aplicación. Deben de ser configurables y con posibilidad de formateo automático de un campo numérico.
- **Gráficos de barras:** Permiten mostrar de una forma visual la información dada por la aplicación. Útil para visualizar comparaciones entre cálculos.
- **Tablas Excel:** Permiten a un usuario introducir los datos a usar por la aplicación. El sistema debe de ser similar al de una hoja de datos. Deben de permitir:
 - Bloquear celdas.
 - Insertar nuevas filas.
 - Eliminar filas.
 - Deshacer los cambios.
 - Rehacer los cambios.
 - Guardar los datos de la tabla.
- **Spinner:** Permite dejar la aplicación en un estado de *Cargando*. Útil para llamadas asíncronas por las que hay que esperar.
- **Notificaciones:** Permiten dar *feedback* al usuario sobre las acciones a realizar en la aplicación. Debe de soportar mensajes de error, advertencia, información y completado correctamente.

Diagrama de Navegabilidad

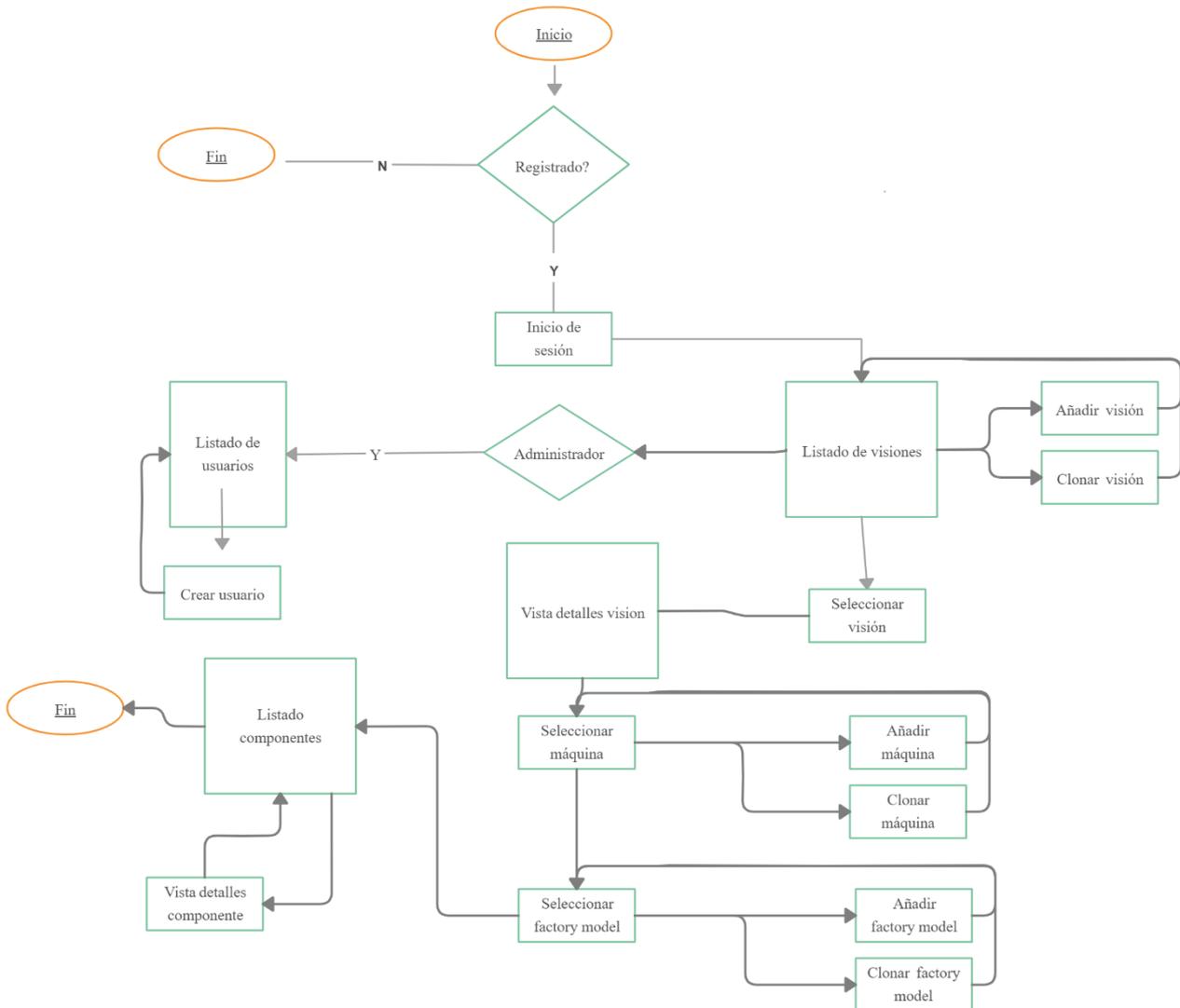


Fig 42. Diagrama de navegabilidad

ASI 10: ESPECIFICACIÓN DEL PLAN DE PRUEBAS

Pruebas Unitarias

Inicio de sesión

Iniciar sesión	
Prueba	Resultado Esperado
<i>Inicio de sesión correcto</i>	<i>El sistema permitirá entrar a la zona privada devolviendo un token JWT con la duración definida.</i>
<i>Inicio de sesión correcto escogiendo recordar los datos</i>	<i>El sistema permitirá entrar a la zona privada devolviendo un token JWT con una duración definida superior al anterior caso.</i>
<i>Inicio de sesión con datos erróneos</i>	<i>El sistema devolverá un mensaje de error.</i>
<i>Inicio de sesión de usuario desactivado</i>	<i>El sistema devolverá un mensaje de error.</i>

Tabla 50. Pruebas Unitarias: Iniciar sesión

Gestión de usuarios

Consulta de usuarios	
Prueba	Resultado Esperado
<i>Consulta con datos</i>	<i>El sistema devolverá la lista de usuarios disponibles.</i>

Tabla 51. Pruebas Unitarias: Consulta de usuarios

Dar de alta a usuarios	
Prueba	Resultado Esperado
<i>Añadir entidad correctamente</i>	<i>El sistema permitirá añadir al sistema la entidad.</i>
<i>Añadir entidad con datos erróneos</i>	<i>El sistema responderá con un mensaje de error.</i>

Tabla 52. Pruebas Unitarias: Dar de alta a usuarios

Modificar usuario	
Prueba	Resultado Esperado
<i>Modificar entidad que existe</i>	<i>El sistema permitirá modificar los datos de la entidad.</i>

Modificar entidad que no existe El sistema responderá con un mensaje de error.

Tabla 53. Pruebas Unitarias: Modificar usuario

Consultar detalles de un usuario	
Prueba	Resultado Esperado
<i>El usuario existe</i>	<i>El sistema devolverá la entidad guardada en el sistema de persistencia.</i>
<i>El usuario no existe</i>	<i>El sistema responderá con un mensaje de error.</i>

Tabla 54. Pruebas Unitarias: Consultar detalles de un usuario

Gestión de visiones

Consulta de visiones	
Prueba	Resultado Esperado
<i>Consulta con datos</i>	<i>Se mostrará el número de visiones disponibles en el sistema.</i>

Tabla 55. Pruebas Unitarias: Consulta de visiones

Añadir visión	
Prueba	Resultado Esperado
<i>Añadir entidad correctamente</i>	<i>El sistema permitirá añadir al sistema la entidad.</i>
<i>Añadir entidad sin datos</i>	<i>El sistema responderá con un mensaje de error.</i>

Tabla 56. Pruebas Unitarias: Añadir visión

Modificar visión	
Prueba	Resultado Esperado
<i>Modificar entidad que existe</i>	<i>El sistema permitirá modificar los datos de la entidad.</i>
<i>Modificar entidad que no existe</i>	<i>El sistema responderá con un mensaje de error.</i>

Tabla 57. Pruebas Unitarias: Modificar visión

Clonar visión	
Prueba	Resultado Esperado
<i>Clonar entidad que existe</i>	<i>El sistema permitirá clonar una entidad.</i>
<i>Clonar entidad que no existe</i>	<i>El sistema responderá con un mensaje de error.</i>

Tabla 58. Pruebas Unitarias: Clonar visión

Gestión de máquinas

Consulta de máquinas	
Prueba	Resultado Esperado
<i>Consulta con datos</i>	<i>El sistema devolverá la lista de máquinas disponibles para una Visión.</i>
<i>Visión no válido</i>	<i>El sistema responderá con un mensaje de error.</i>

Tabla 59. Pruebas Unitarias: Consulta de máquinas

Añadir máquina	
Prueba	Resultado Esperado
<i>Añadir entidad correctamente</i>	<i>El sistema permitirá añadir al sistema la entidad.</i>
<i>Añadir entidad sin datos</i>	<i>El sistema responderá con un mensaje de error.</i>

Tabla 60. Pruebas Unitarias: Añadir máquina

Modificar máquinas	
Prueba	Resultado Esperado
<i>Modificar entidad que existe</i>	<i>El sistema permitirá modificar los datos de la entidad.</i>
<i>Modificar entidad que no existe</i>	<i>El sistema responderá con un mensaje de error.</i>

Tabla 61. Pruebas Unitarias: Modificar máquinas

Clonar máquinas	
Prueba	Resultado Esperado
<i>Clonar entidad que existe</i>	<i>El sistema permitirá clonar una entidad.</i>
<i>Clonar entidad que no existe</i>	<i>El sistema responderá con un mensaje de error.</i>

Tabla 62. Pruebas Unitarias: Clonar máquinas

Gestión de FactoryModel

Consulta de FactoryModels	
Prueba	Resultado Esperado
<i>Consulta con datos</i>	<i>El sistema devolverá la lista de FactoryModels disponibles.</i>

Tabla 63. Pruebas Unitarias: Consulta de FactoryModels

Consulta de Componentes	
Prueba	Resultado Esperado

Consulta con datos	<i>El sistema devolverá la lista de componentes disponibles para un FactoryModels.</i>
FactoryModel no válido	<i>El sistema responderá con un mensaje de error.</i>

Tabla 64. Pruebas Unitarias: Consulta de Componentes

Añadir FactoryModels	
Prueba	Resultado Esperado
Añadir correctamente	<i>El sistema permitirá añadirá al sistema la entidad.</i>
Añadir entidad sin datos	<i>El sistema responderá con un mensaje de error.</i>

Tabla 65. Pruebas Unitarias: Añadir FactoryModels

Modificar FactoryModels	
Prueba	Resultado Esperado
Modificar entidad que existe	<i>El sistema permitirá modificar los datos de la entidad.</i>
Modificar entidad que no existe	<i>El sistema responderá con un mensaje de error.</i>

Tabla 66. Pruebas Unitarias: Modificar FactoryModels

Clonar FactoryModels	
Prueba	Resultado Esperado
Clonar entidad que existe	<i>El sistema permitirá clonar una entidad.</i>
Clonar entidad que no existe	<i>El sistema responderá con un mensaje de error.</i>

Tabla 67. Pruebas Unitarias: Clonar FactoryModels

Costes

Vista de detalles y edición de FM Costes	
Prueba	Resultado Esperado
Consultar FM Costes de un FactoryModel	<i>El sistema devolverá correctamente el componente Costes pedido.</i>
Consulta FM Costes de un FactoryModel que no existe	<i>El sistema responderá con un mensaje de error.</i>
Actualizar información mostrada	<i>El sistema devolverá la entidad actualizada sin realizar cambios en la persistencia.</i>
Actualizar información	<i>El sistema modificará la entidad de la persistencia correctamente.</i>

Actualizar información de entidad no válida El sistema responderá con un mensaje de error.

Tabla 68. Pruebas Unitarias: Vista de detalles y edición de FM Costes

Creación de FM Costes	
Prueba	Resultado Esperado
Crear FM Costes	El sistema creará una nueva entidad con los datos por defecto del sistema.
Crear FM Costes existiendo uno asignado al mismo FactoryModel	El sistema borrará todos los datos anteriores del sistema y creará una nueva entidad con los datos por defecto.
Crear otro FM	El sistema responderá con un mensaje de error.

Tabla 69. Pruebas Unitarias: Creación de FM Costes

Clonación de FM Costes	
Prueba	Resultado Esperado
Clonar la entidad correctamente	El sistema creará una nueva entidad con los datos originales.
Clonar entidad con datos incorrectos	El sistema responderá con un mensaje de error.

Tabla 70. Pruebas Unitarias: Clonación de FM Costes

Energía

Vista de detalles y edición de FM Energía	
Prueba	Resultado Esperado
Consultar FM Energías de un FactoryModel	El sistema devolverá correctamente el componente Energías pedido.
Consulta FM Energía de un FactoryModel que no existe	El sistema responderá con un mensaje de error.
Actualizar información mostrada	El sistema devolverá la entidad actualizada sin realizar cambios en la persistencia.
Actualizar información	El sistema modificará la entidad de la persistencia correctamente.
Actualizar información de entidad no válida	El sistema responderá con un mensaje de error.

Tabla 71. Pruebas Unitarias: Vista detalles y edición de FM Energía

Creación de FM Energías	
Prueba	Resultado Esperado

Crear FM Energías	<i>El sistema creará una nueva entidad con los datos por defecto del sistema. Se debe de comprobar que se inicializa el sistema para un modelo 3.X.</i>
Crear FM Energía existiendo uno asignado al mismo FactoryModel	<i>El sistema borrará todos los datos anteriores del sistema y creará una nueva entidad con los datos por defecto.</i>
Crear otro FM	<i>El sistema responderá con un mensaje de error.</i>

Tabla 72. Pruebas Unitarias: Creación de FM Calidad

Clonación de FM Energías	
Prueba	Resultado Esperado
Clonar la entidad correctamente	<i>El sistema creará una nueva entidad con los datos originales.</i>
Clonar entidad con datos incorrectos	<i>El sistema responderá con un mensaje de error.</i>

Tabla 73. Pruebas Unitarias: Clonación de FM Energías

Producción

Vista de detalles y edición de FM Producción	
Prueba	Resultado Esperado
Consultar FM Producción de un FactoryModel	<i>El sistema devolverá correctamente el componente Producción pedido.</i>
Consulta FM Producción de un FactoryModel que no existe	<i>El sistema responderá con un mensaje de error.</i>
Actualizar información mostrada	<i>El sistema devolverá la entidad actualizada sin realizar cambios en la persistencia.</i>
Actualizar información	<i>El sistema modificará la entidad de la persistencia correctamente.</i>
Actualizar información de entidad no válida	<i>El sistema responderá con un mensaje de error.</i>

Tabla 74. Pruebas Unitarias: Vista de detalles y edición de FM Producción

Creación de FM Producción	
Prueba	Resultado Esperado
Crear FM Producción	<i>El sistema creará una nueva entidad con los datos por defecto del sistema.</i>
Crear FM Producción existiendo uno asignado al mismo FactoryModel	<i>El sistema borrará todos los datos anteriores del sistema y creará una nueva entidad con los datos por defecto.</i>

Crear otro FM

El sistema responderá con un mensaje de error.

Tabla 75. Pruebas Unitarias: Creación de FM Producción

Clonación de FM Producción	
Prueba	Resultado Esperado
Clonar la entidad correctamente	<i>El sistema creará una nueva entidad con los datos originales.</i>
Clonar entidad con datos incorrectos	<i>El sistema responderá con un mensaje de error.</i>

Tabla 76. Pruebas Unitarias: Clonación de FM Producción

Calidad

Vista de detalles y edición de FM Calidad	
Prueba	Resultado Esperado
Consultar FM Calidad de un FactoryModel	<i>El sistema devolverá correctamente el componente Calidad pedido.</i>
Consulta FM Calidad de un FactoryModel que no existe	<i>El sistema responderá con un mensaje de error.</i>
Actualizar información mostrada	<i>El sistema devolverá la entidad actualizada sin realizar cambios en la persistencia.</i>
Actualizar información	<i>El sistema modificará la entidad de la persistencia correctamente.</i>
Actualizar información de entidad no válida	<i>El sistema responderá con un mensaje de error.</i>

Tabla 77. Pruebas Unitarias: Vista de detalles y edición de FM Calidad

Creación de FM Calidad	
Prueba	Resultado Esperado
Crear FM Calidad	<i>El sistema creará una nueva entidad con los datos por defecto del sistema.</i>
Crear FM Calidad existiendo uno asignado al mismo FactoryModel	<i>El sistema borrará todos los datos anteriores del sistema y creará una nueva entidad con los datos por defecto.</i>
Crear otro FM	<i>El sistema responderá con un mensaje de error.</i>

Tabla 78. Pruebas Unitarias: Creación de FM Calidad

Clonación de FM Calidad	
Prueba	Resultado Esperado



Clonar la entidad correctamente	<i>El sistema creará una nueva entidad con los datos originales.</i>
--	--

Clonar entidad con datos incorrectos	<i>El sistema responderá con un mensaje de error.</i>
---	---

Tabla 79. Pruebas Unitarias: Clonación de FM Calidad

Pruebas de Comportamiento

A continuación, se definirán las pruebas de comportamiento que se han de hacer a la aplicación. Para especificar las diferentes pruebas a realizar, se hará haciendo uso del lenguaje DSL Gherkin [12].

Inicio de sesión

Caso de uso: Inicio de sesión	
Escenario	Tarea
Iniciar sesión con datos correctos.	<i>Given:</i> Un usuario. <i>When:</i> Introduce sus credenciales. <i>Then:</i> Inicia sesión correctamente. Redirección a <i>layout</i> privado.
Iniciar sesión con datos incorrectos.	<i>Given:</i> Un usuario. <i>When:</i> Introduce sus credenciales. <i>Then:</i> Mensaje de error de usuario o contraseña incorrecto.
Iniciar sesión un usuario desactivado.	<i>Given:</i> Un usuario. <i>When:</i> Introduce sus credenciales. <i>Then:</i> Mensaje de error de usuario desactivado.

Tabla 80. Pruebas de comportamiento: Inicio de sesión

Gestión de usuarios

Caso de uso: Dar de alta a usuarios	
Escenario	Tarea
Dar de alta a un usuario correctamente.	<i>Given:</i> Un usuario administrador. <i>When:</i> Introduce todos los campos. <i>Then:</i> Mensaje de que se creado correctamente el usuario. <i>And:</i> Nuevo usuario en el sistema.
No introducir todos los campos necesarios.	<i>Given:</i> Un usuario administrador. <i>When:</i> Introduce 0..n-1 campos obligatorios. <i>Then:</i> Mensaje de error de campo requerido.
El usuario cancela la operación.	<i>Given:</i> Un usuario administrador. <i>When:</i> Pulsar botón de cancelar operación. <i>Then:</i> Regreso a listado de usuarios.
El usuario ya existe en el sistema.	<i>Given:</i> Un usuario administrador. <i>When:</i> Introduce todos los campos completando el usuario con un valor ya existente en el SGBD. <i>Then:</i> Mensaje de error de usuario ya existente.

Tabla 81. Pruebas de comportamiento: Dar de alta a usuarios

Caso de uso: Consulta todos los usuarios	
Escenario	Tarea
Consultar los usuarios del sistema	<i>Given:</i> Un usuario administrador. <i>When:</i> Se dirige a la zona privada de gestión de usuarios. <i>Then:</i> Listado de los usuarios del sistema.

Tabla 82. Pruebas de comportamiento: Consulta todos los usuarios

Caso de uso: Modificar usuario	
Escenario	Tarea
Modificar un usuario.	<i>Given:</i> Un usuario administrador en listado de usuarios. <i>When:</i> Introduce todos los campos. <i>Then:</i> Mensaje de que se modificado correctamente el usuario. <i>And:</i> Modificado usuario en el sistema.
No introducir todos los campos necesarios.	<i>Given:</i> Un usuario administrador en listado de usuarios. <i>When:</i> Introduce 0..n-1 campos obligatorios. <i>Then:</i> Mensaje de error de campo requerido.

Tabla 83. Pruebas de comportamiento: Modificar usuario

Caso de uso: Consulta detalles de un usuario	
Escenario	Tarea
Consultar detalles de un usuario	<i>Given:</i> Un usuario que ha iniciado sesión. <i>When:</i> Se dirige a página personal del usuario que ha iniciado sesión. <i>Then:</i> Página de perfil del usuario que ha iniciado sesión.

Tabla 84. Pruebas de comportamiento: Consulta todos los usuarios

Gestión de visiones

Caso de uso: Consulta visiones	
Escenario	Tarea
Consultar listado de visiones	<i>Given:</i> Un usuario que ha iniciado sesión con permisos de lectura. <i>When:</i> Se dirige al listado de visiones. <i>Then:</i> Listado de las visiones del sistema.

Tabla 85. Pruebas de comportamiento: Consulta de visiones

Caso de uso: Añadir visión	
Escenario	Tarea
Dar de alta una visión.	<i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura. <i>When:</i> Selecciona añadir visión. <i>And:</i> Introduce todos los campos. <i>Then:</i> Mensaje de que se creado correctamente la visión.

	<i>And:</i> Nueva visión en el sistema.
No introducir todos los campos necesarios.	<i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura. <i>When:</i> Pulsar botón añadir visión e introduce 0..n-1 campos obligatorios. <i>Then:</i> Mensaje de error de campo requerido.
El usuario cancela la operación.	<i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura. <i>When:</i> Pulsar botón añadir visión. <i>And:</i> Pulsar botón de cancelar operación. <i>Then:</i> Vuelta al listado de visiones.

Tabla 86. Pruebas de comportamiento: Añadir visión

Caso de uso: Modificar visión	
Escenario	Tarea
Modificar una visión.	<i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura. <i>When:</i> Selecciona editar una visión ya creada. <i>And:</i> Introduce todos los campos. <i>Then:</i> Mensaje de que se modificado correctamente la visión. <i>And:</i> Modificada visión en el sistema.
No introducir todos los campos necesarios.	<i>Given:</i> Un usuario administrador en listado de usuarios. <i>When:</i> Introduce 0..n-1 campos obligatorios. <i>Then:</i> Mensaje de error de campo requerido.

Tabla 87. Pruebas de comportamiento: Modificar visión

Caso de uso: Clonar visión	
Escenario	Tarea
Clonar visión	<i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura. <i>When:</i> Selecciona clonar una visión ya creada <i>And:</i> Modifica los campos. <i>Then:</i> Mensaje de confirmación que se ha realizado correctamente. <i>And:</i> Nueva visión clonada del sistema.
El usuario cancela la operación.	<i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura. <i>When:</i> Pulsar botón de cancelar operación. <i>Then:</i> Vuelta al listado de visiones.

Tabla 88. Pruebas de comportamiento: Clonar visión

Gestión de máquinas

Caso de uso: Consulta máquinas	
Escenario	Tarea

Consultar listado de máquinas de una visión *Given:* Un usuario que ha iniciado sesión con permisos de lectura.
When: Seleccionar una visión en listado de visiones.
Then: Listado de las máquinas del sistema.

Tabla 89. Pruebas de comportamiento: Consulta de máquinas

Caso de uso: Añadir máquinas	
Escenario	Tarea
Dar de alta una máquina.	<i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura. <i>When:</i> Seleccionar una visión en listado de visiones. <i>And:</i> Pulsar en el botón de añadir. <i>And:</i> Introducir todos los campos. <i>Then:</i> Mensaje de que se creado correctamente la máquina. <i>And:</i> Nueva máquina en el sistema.
No introducir todos los campos necesarios.	<i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura. <i>When:</i> Seleccionar una visión en listado de visiones. <i>And:</i> Pulsar en el botón de añadir. <i>And:</i> Introduce 0..n-1 campos obligatorios. <i>Then:</i> Mensaje de error de campos requeridos.
El usuario cancela la operación.	<i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura. <i>When:</i> Seleccionar una visión en listado de visiones. <i>And:</i> Pulsar en el botón de añadir. <i>And:</i> Pulsar en el botón de cancelar operación. <i>Then:</i> Mensaje de error de campos requeridos.

Tabla 90. Pruebas de comportamiento: Añadir máquinas

Caso de uso: Modificar máquina	
Escenario	Tarea
Modificar una máquina.	<i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura. <i>When:</i> Selecciona editar una visión ya creada. <i>And:</i> Pulsar en editar una máquina. <i>And:</i> Introduce todos los campos. <i>Then:</i> Mensaje de que se modificado correctamente la máquina. <i>And:</i> Modificada máquina del sistema.
No introducir todos los campos necesarios.	<i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura. <i>When:</i> Selecciona una visión ya creada. <i>And:</i> Pulsar en editar una máquina. <i>And:</i> Introduce 0..n-1 campos obligatorios. <i>Then:</i> Mensaje de error de campo requerido.

Tabla 91. Pruebas de comportamiento: Modificar máquina.

Caso de uso: Clonar máquina	
Escenario	Tarea
Clonar máquina	<p><i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura.</p> <p><i>When:</i> Selecciona una visión ya creada.</p> <p><i>And:</i> Pulsar en clonar una máquina.</p> <p><i>And:</i> Modificar todos los campos.</p> <p><i>Then:</i> Mensaje de que se clonado correctamente la máquina.</p> <p><i>And:</i> Nueva máquina clonada del sistema.</p>
El usuario cancela la operación.	<p><i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura.</p> <p><i>When:</i> Selecciona una visión ya creada.</p> <p><i>And:</i> Pulsar en clonar una máquina.</p> <p><i>And:</i> Pulsar en cancelar operación.</p> <p><i>Then:</i> Vuelta a listado de máquinas.</p>

Tabla 92. Pruebas de comportamiento: Clonar visión

Gestión de FactoryModel

Caso de uso: Consulta de FactoryModel	
Escenario	Tarea
Consultar listado de FactoryModels de una máquina	<p><i>Given:</i> Un usuario que ha iniciado sesión con permisos de lectura.</p> <p><i>When:</i> Seleccionar una visión en listado de visiones.</p> <p><i>And:</i> Selecciona una máquina.</p> <p><i>Then:</i> Listado de los FactoryModels de esa máquina.</p>

Tabla 93. Pruebas de comportamiento: Consulta de FactoryModel

Caso de uso: Consulta de componentes	
Escenario	Tarea
Consultar listado de componentes dado un FactoryModels	<p><i>Given:</i> Un usuario que ha iniciado sesión con permisos de lectura.</p> <p><i>When:</i> Seleccionar una visión en listado de visiones.</p> <p><i>And:</i> Selecciona una máquina.</p> <p><i>And:</i> Selecciona un FactoryModel.</p> <p><i>Then:</i> Listado de los componentes del FactoryModels.</p>

Tabla 94. Pruebas de comportamiento: Consulta de componentes

Caso de uso: Añadir FactoryModels	
Escenario	Tarea
Dar de alta un FactoryModel.	<p><i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura.</p> <p><i>When:</i> Seleccionar una visión en listado de visiones.</p> <p><i>And:</i> Selecciona una máquina en listado de máquinas.</p> <p><i>And:</i> Pulsar en el botón de añadir.</p>

	<p><i>And:</i> Introducir todos los campos.</p> <p><i>Then:</i> Mensaje de que se creado correctamente el FactoryModel.</p> <p><i>And:</i> Nuevo FactoryModel en el sistema.</p>
No introducir todos los campos necesarios.	<p><i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura.</p> <p><i>When:</i> Seleccionar una visión en listado de visiones.</p> <p><i>And:</i> Selecciona una máquina en listado de máquinas.</p> <p><i>And:</i> Pulsar en el botón de añadir.</p> <p><i>And:</i> Introduce 0..n-1 campos obligatorios.</p> <p><i>Then:</i> Mensaje de error de campos requeridos.</p>
El usuario cancela la operación.	<p><i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura.</p> <p><i>When:</i> Seleccionar una visión en listado de visiones.</p> <p><i>And:</i> Selecciona una máquina en listado de máquinas.</p> <p><i>And:</i> Pulsar en el botón de añadir.</p> <p><i>And:</i> Pulsar en el botón de cancelar operación.</p> <p><i>Then:</i> Mensaje de error de campos requeridos.</p>

Tabla 95. Pruebas de comportamiento: Añadir FactoryModel

Caso de uso: Modificar FactoryModels	
Escenario	Tarea
Modificar una máquina.	<p><i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura.</p> <p><i>When:</i> Seleccionar una visión en listado de visiones.</p> <p><i>And:</i> Selecciona una máquina en listado de máquinas.</p> <p><i>And:</i> Pulsar en el botón de editar.</p> <p><i>And:</i> Introduce todos los campos.</p> <p><i>Then:</i> Mensaje de que se modificado correctamente la máquina.</p> <p><i>And:</i> Modificado FactoryModel del sistema.</p>
No introducir todos los campos necesarios.	<p><i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura.</p> <p><i>When:</i> Seleccionar una visión en listado de visiones.</p> <p><i>And:</i> Selecciona una máquina en listado de máquinas.</p> <p><i>And:</i> Pulsar en el botón de editar.</p> <p><i>And:</i> Introduce 0..n-1 campos obligatorios.</p> <p><i>Then:</i> Mensaje de error de campo requerido.</p>

Tabla 96. Pruebas de comportamiento: Modificar FactoryModel.

Caso de uso: Clonar FactoryModels	
Escenario	Tarea
Clonar máquina	<p><i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura.</p> <p><i>When:</i> Seleccionar una visión en listado de visiones.</p> <p><i>And:</i> Selecciona una máquina en listado de máquinas.</p> <p><i>And:</i> Seleccionar un FactoryModel.</p>

	<p><i>And:</i> Pulsar en el botón de clonar.</p> <p><i>And:</i> Modificar todos los campos.</p> <p><i>Then:</i> Mensaje de que se clonado correctamente la máquina.</p> <p><i>And:</i> Nuevo FactoryModel clonado del sistema.</p>
El usuario cancela la operación.	<p><i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura.</p> <p><i>When:</i> Seleccionar una visión en listado de visiones.</p> <p><i>And:</i> Selecciona una máquina en listado de máquinas.</p> <p><i>And:</i> Pulsar en el botón de clonar.</p> <p><i>And:</i> Pulsar en cancelar operación.</p> <p><i>Then:</i> Vuelta a listado de FactoryModel.</p>

Tabla 97. Pruebas de comportamiento: Clonar FactoryModel

Costes

Caso de uso: Añadir costes	
Escenario	Tarea
Dar de alta un nuevo coste	<p><i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura.</p> <p><i>When:</i> Seleccionar una visión en listado de visiones.</p> <p><i>And:</i> Selecciona una máquina en listado de máquinas.</p> <p><i>And:</i> Selecciona un FactoryModel en listado de FactoryModel.</p> <p><i>And:</i> Selecciona componente de costes.</p> <p><i>And:</i> Introducir nuevo coste fijo y/o variable.</p> <p><i>And:</i> Guardar cambios.</p> <p><i>Then:</i> Actualización de los cálculos del sistema.</p>

Tabla 98. Pruebas de comportamiento: Añadir costes

Caso de uso: Consulta de costes	
Escenario	Tarea
Consultar listado de componentes Costes dado un FactoryModel	<p><i>Given:</i> Un usuario que ha iniciado sesión con permisos de lectura.</p> <p><i>When:</i> Seleccionar una visión en listado de visiones.</p> <p><i>And:</i> Selecciona una máquina en listado de máquinas.</p> <p><i>And:</i> Selecciona un FactoryModel en listado de FactoryModel.</p> <p><i>And:</i> Selecciona componente de costes.</p> <p><i>Then:</i> Cálculos relativos a ese componente de costes.</p>

Tabla 99. Pruebas de comportamiento: Consulta de costes

Energía

Caso de uso: Añadir potencia	
Escenario	Tarea

Modificar valores de potencia añadiendo nuevos datos	<p><i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura.</p> <p><i>When:</i> Seleccionar una visión en listado de visiones.</p> <p><i>And:</i> Selecciona una máquina en listado de máquinas.</p> <p><i>And:</i> Selecciona un FactoryModel en listado de FactoryModel.</p> <p><i>And:</i> Selecciona componente de energía.</p> <p><i>And:</i> Modificar valores de potencia.</p> <p><i>And:</i> Guardar cambios.</p> <p><i>Then:</i> Actualización de los cálculos del sistema.</p>
---	--

Tabla 100. Pruebas de comportamiento: Añadir potencia

Caso de uso: Añadir consumo	
Escenario	Tarea
Modificar valores de consumo añadiendo nuevos datos	<p><i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura.</p> <p><i>When:</i> Seleccionar una visión en listado de visiones.</p> <p><i>And:</i> Selecciona una máquina en listado de máquinas.</p> <p><i>And:</i> Selecciona un FactoryModel en listado de FactoryModel.</p> <p><i>And:</i> Selecciona componente de energía.</p> <p><i>And:</i> Modificar valores de consumo.</p> <p><i>And:</i> Guardar cambios.</p> <p><i>Then:</i> Actualización de los cálculos del sistema.</p>

Tabla 101. Pruebas de comportamiento: Añadir consumo

Caso de uso: Consulta de términos	
Escenario	Tarea
Consultar listado de componentes Energía dado un FactoryModel	<p><i>Given:</i> Un usuario que ha iniciado sesión con permisos de lectura.</p> <p><i>When:</i> Seleccionar una visión en listado de visiones.</p> <p><i>And:</i> Selecciona una máquina en listado de máquinas.</p> <p><i>And:</i> Selecciona un FactoryModel en listado de FactoryModel.</p> <p><i>And:</i> Selecciona componente de energía.</p> <p><i>Then:</i> Cálculos relativos a ese componente de energía.</p>

Tabla 102. Pruebas de comportamiento: Consulta de energía

Producción

Caso de uso: Añadir productos	
Escenario	Tarea
Dar de alta un nuevo producto	<p><i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura.</p> <p><i>When:</i> Seleccionar una visión en listado de visiones.</p> <p><i>And:</i> Selecciona una máquina en listado de máquinas.</p> <p><i>And:</i> Selecciona un FactoryModel en listado de FactoryModel.</p>

And: Selecciona componente de producción.
And: Introducir nuevo producto.
And: Guardar cambios.
Then: Actualización de los cálculos del sistema.

Tabla 103. Pruebas de comportamiento: Añadir productos

Caso de uso: Consulta de producción	
Escenario	Tarea
Consultar listado de componentes	<i>Given:</i> Un usuario que ha iniciado sesión con permisos de lectura. <i>When:</i> Seleccionar una visión en listado de visiones.
Producción dado un FactoryModel	<i>And:</i> Selecciona una máquina en listado de máquinas. <i>And:</i> Selecciona un FactoryModel en listado de FactoryModel. <i>And:</i> Selecciona componente de producción. <i>Then:</i> Cálculos relativos a ese componente de producción.

Tabla 104. Pruebas de comportamiento: Consulta de producción

Calidad

Caso de uso: Añadir defectos	
Escenario	Tarea
Dar de alta un nuevo defecto	<i>Given:</i> Un usuario que ha iniciado sesión con permisos de escritura. <i>When:</i> Seleccionar una visión en listado de visiones. <i>And:</i> Selecciona una máquina en listado de máquinas. <i>And:</i> Selecciona un FactoryModel en listado de FactoryModel. <i>And:</i> Selecciona componente de calidad. <i>And:</i> Seleccionar un producto existente. <i>And:</i> Introducir nuevo defecto. <i>And:</i> Guardar cambios. <i>Then:</i> Actualización de los cálculos del sistema.

Tabla 105. Pruebas de comportamiento: Añadir defectos

Caso de uso: Consulta de producción	
Escenario	Tarea
Consultar listado de componentes Calidad dado un FactoryModel	<i>Given:</i> Un usuario que ha iniciado sesión con permisos de lectura. <i>When:</i> Seleccionar una visión en listado de visiones. <i>And:</i> Selecciona una máquina en listado de máquinas. <i>And:</i> Selecciona un FactoryModel en listado de FactoryModel. <i>And:</i> Selecciona componente de calidad. <i>Then:</i> Cálculos relativos a ese componente de producción.

Tabla 106. Pruebas de comportamiento: Consulta de calidad



Pruebas de Usabilidad y Accesibilidad

Se realizará el cuestionario definido por *Yusef Hassan Montero*. Se puede consultar información relativo a continuación [13]. La prueba será válida si en, su mayor medida, las afirmaciones tienen una respuesta afirmativa.

Como medio de prueba adicional, se harán pruebas con usuarios reales del funcionamiento de la aplicación para comprobar si la aplicación es usable por cualquier tipo de usuario.



Pruebas de Carga y Rendimiento

Se realizarán este tipo de pruebas para comprobar que el sistema funciona correctamente a una carga de usuarios dada. Se deberá hacer pruebas contra el sistema desplegado en un entorno de pruebas.

Pruebas de Código

Este tipo de pruebas se van a llevar a cabo para determinar la calidad, la reusabilidad, el acoplamiento de las clases y el código muerto existente en la aplicación.

Complejidad ciclomática	
Prueba	Resultado Esperado
Número de caminos	<i>El sistema, en su mayor medida, no debe de superior 50.</i>

Tabla 107. Pruebas de Código: Complejidad ciclomática

Profundidad de herencia	
Prueba	Resultado Esperado
Número de clases de herencia	<i>El sistema, en su mayor medida, no debe de exceder 4 niveles de herencia.</i>

Tabla 108. Pruebas de Código: Profundidad de herencia

Líneas de código	
Prueba	Resultado Esperado
Líneas de código por método	<i>Un método, en su mayor medida, no debe de superar las 20 líneas de código.</i>

Tabla 109. Pruebas de Código: Líneas de código

Mantenibilidad	
Prueba	Resultado Esperado
Probar si el código es mantenible	<i>Debe de alcanzarse un número superior a 20 y, preferiblemente, mayor a 50.</i>

Tabla 110. Pruebas de Código: Mantenibilidad



Capítulo 4 DISEÑO DEL SISTEMA DE INFORMACIÓN

FASE DE DESARROLLO

DSI

DSI 4: DISEÑO DE CLASES

Para la representación del sistema, se ha decidido separar los diagramas de clases en diferentes apartados. Se detallarán las principales clases de los diferentes módulos del sistema.

Clases comunes del sistema

Debido a la necesidad que tiene el sistema de tener duplicadas entidades, es necesario disponer de unas clases comunes en un módulo independiente que pueda ser usado por todos los módulos de la lógica.

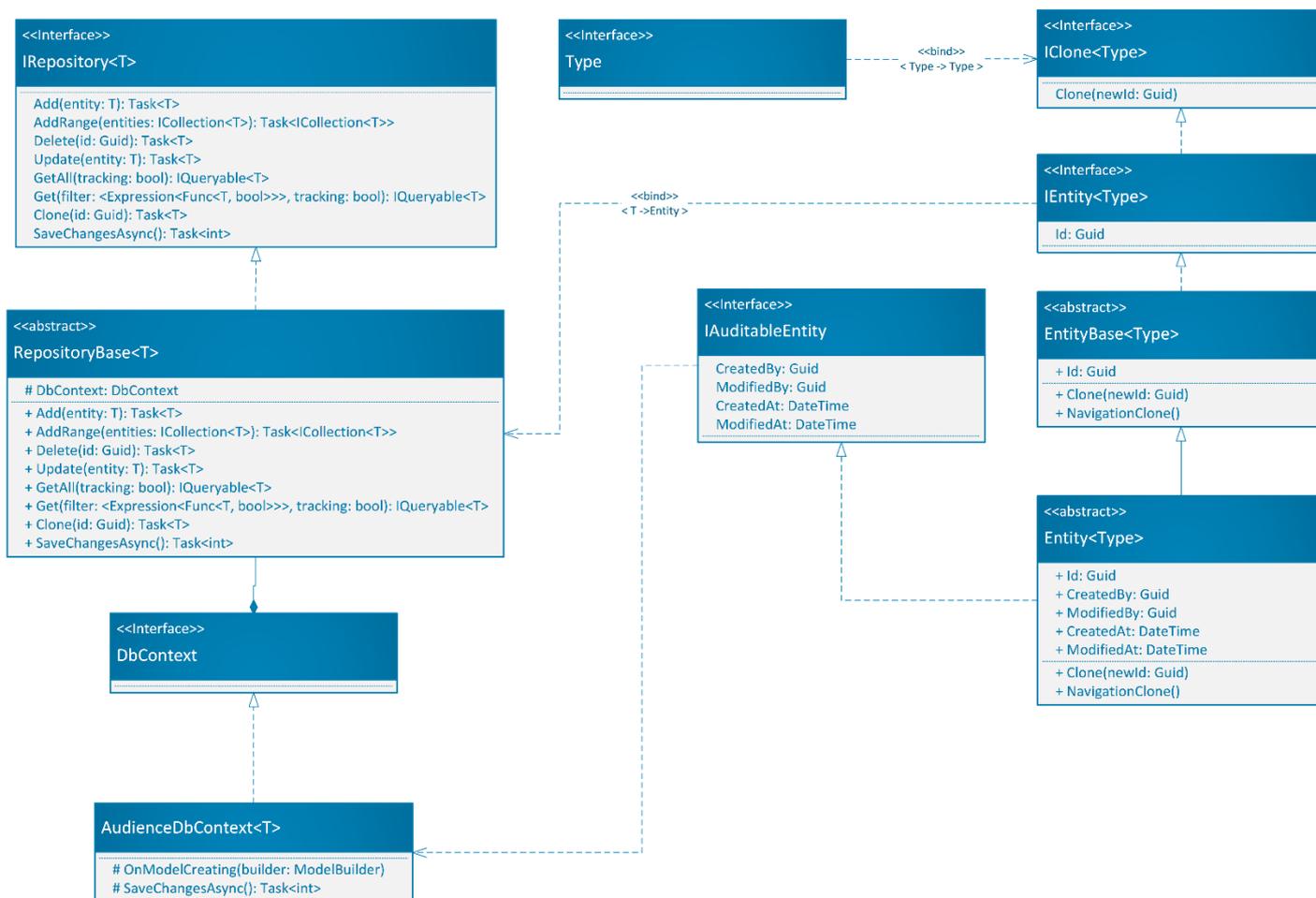


Fig 43. Diagrama de clases comunes (Entidades y Repositorios)

Interfaz	
IClone	
Descripción	
Contrato para permitir a una clase clonarse.	
Métodos	

Nombre	Return	Descripción
Clone	Type	Contrato para clonar una entidad devolviendo el nuevo objeto.

Tabla 111. Diseño de clases: Interfaz IClone

Interfaz		
IAuditableEntity		
Descripción		
Contrato para permitir a un objeto que sea auditable.		
Atributos		
Nombre	Tipo	Descripción
CreatedBy	Guid	Contrato para especificar la entidad que ha generado la entidad.
ModifiedBy	Guid	Contrato para especificar la entidad que ha modificado la entidad.
CreatedAt	DateTime	Contrato para especificar la fecha de creación.
CreatedBy	DateTime	Contrato para especificar la fecha de modificación.

Tabla 112. Diseño de clases: Interfaz IAuditableEntity

Interfaz		
IEntity		
Descripción		
Contrato con la información mínima de una entidad.		
Atributos		
Nombre	Tipo	Descripción
Id	Guid	Identificador de la entidad.

Tabla 113. Diseño de clases: Interfaz IEntity

Clase		
EntityBase		
Descripción		
Implementación de una entidad con las características mínimas para ser una entidad.		
Atributos		
Nombre	Tipo	Descripción
Id	Guid	Identificador de la entidad.
Métodos		
Nombre	Return	Descripción
Clone	void	Método para clonar una entidad dada.

NavigationClone	void	Método que se redefine para permitir clonar entidades relacionadas.
-----------------	------	---

Tabla 114. Diseño de clases: Clase abstracta EntityBase

Clase		
Entity		
Descripción		
Implementación de una entidad con las características para auditarse.		
Atributos		
Nombre	Tipo	Descripción
Id	Guid	Identificador de la entidad
CreatedBy	Guid	Atributo para especificar la entidad que ha generado la entidad.
ModifiedBy	Guid	Atributo para especificar la entidad que ha modificado la entidad.
CreatedAt	DateTime	Atributo para especificar la fecha de creación.
CreatedBy	DateTime	Atributo para especificar la fecha de modificación.
Métodos		
Nombre	Return	Descripción
Clone	void	Método para clonar una entidad dada.
NavigationClone	void	Método que se redefine para permitir clonar entidades relacionadas.

Tabla 115. Diseño de clases: Clase abstracta Entity

Interfaz		
IRepository		
Descripción		
Contrato de un repositorio que tiene las acciones CRUD para tratar con una base de datos.		
Métodos		
Nombre	Return	Descripción
Add	Task<T>	Contrato para añadir entidades. Se devuelve la entidad añadida o null.
AddRange	Task<ICollection<T>>	Contrato para añadir entidades al sistema. Se devuelve la lista de entidades añadidas o null en caso de error.
Delete	Task<T>	Contrato para borrar entidades del sistema. Si la entidad existe, se devuelve la entidad. Si no existe, devuelve null.

Update	Task<T>	Contrato para actualizar entidades del sistema. Si la entidad existe, se actualiza con la nueva información. En caso de algún error, se lanzará una excepción.
GetAll	IQueryable<T>	Contrato para obtener todas las entidades de un contexto (por tipo).
Get	IQueryable<T>	Contrato para obtener una lista lazy de los elementos filtrados.
Clone	Task<T>	Contrato para poder clonar la entidad. Si la entidad existe, se devolverá una copia con los nuevos valores calculados. En caso de no existe, devuelve null.
SaveChangesAsync	Task<int>	Contrato que permite guardar los cambios realizados en el Contexto.

Tabla 116. Diseño de clases: Interfaz IRepository

Clase		
RepositoryBase		
Descripción		
Implementación de un repositorio con las acciones básicas de CRUD. Será heredado por las clases hijas.		
Métodos		
Nombre	Return	Descripción
Add	Task<T>	Método para añadir entidades. Se devuelve la entidad añadida o null.
AddRange	Task<ICollection<T>>	Método para añadir entidades al sistema. Se devuelve la lista de entidades añadidas o null en caso de error.
Delete	Task<T>	Método para borrar entidades del sistema. Si la entidad existe, se devuelve la entidad. Si no existe, devuelve null.
Update	Task<T>	Método para actualizar entidades del sistema. Si la entidad existe, se actualiza con la nueva información. En caso de algún error, se lanzará una excepción.
GetAll	IQueryable<T>	Método para obtener todas las entidades de un contexto (por tipo).
Get	IQueryable<T>	Método para obtener una lista lazy de los elementos filtrados.

Clone	Task<T>	Método para poder clonar la entidad. Si la entidad existe, se devolverá una copia con los nuevos valores calculados. En caso de no existe, devuelve null.
SaveChangesAsync	Task<int>	Método que permite guardar los cambios realizados en el Contexto.

Tabla 117. Diseño de clases: Clase abstracta RepositoryBase

Clase		
AuditableContext		
Descripción		
Clase redefinida que permite actualizar los campos de auditoría de forma automática.		
Métodos		
Nombre	Return	Descripción
OnModelCreating	void	Redefinición del método para añadir configuraciones comunes a los repositorios.
SaveChangesAsync	Task<int>	Redefinición del método para dar soporte a entidades que sean de tipo IAuditableEntity.

Tabla 118. Diseño de clases: Clase abstracta AuditableContext

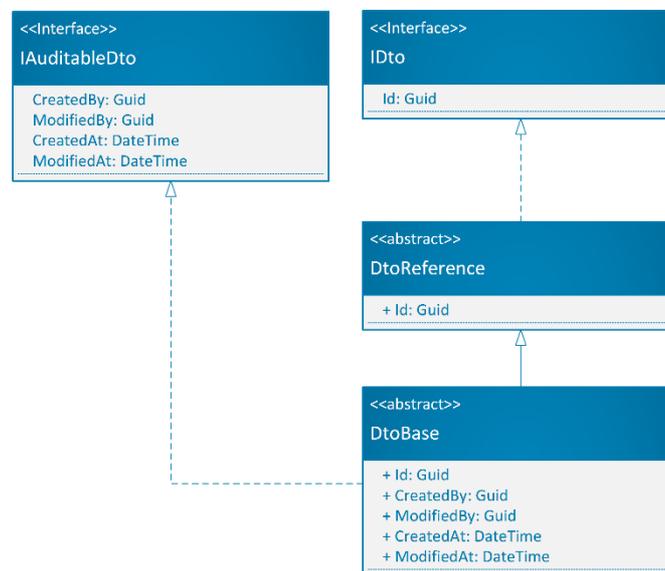


Fig 44. Diagrama de clases comunes (Dto)

Interfaz
IAuditableDto
Descripción

Contrato para permitir a un DTO que sea auditable.		
Atributos		
Nombre	Tipo	Descripción
CreatedBy	Guid	Contrato para especificar la entidad que ha generado la entidad.
ModifiedBy	Guid	Contrato para especificar la entidad que ha modificado la entidad.
CreatedAt	DateTime	Contrato para especificar la fecha de creación.
CreatedBy	DateTime	Contrato para especificar la fecha de modificación.

Tabla 119. Diseño de clases: Interfaz IAuditableDto

Interfaz		
IDto		
Descripción		
Contrato con la información mínima de un DTO para ser compatible con una entidad original.		
Atributos		
Nombre	Tipo	Descripción
Id	Guid	Identificador de la entidad original.

Tabla 120. Diseño de clases: Interfaz IDto

Clase		
DtoBase		
Descripción		
Implementación de un DTO con las características mínimas para ser compatible con una entidad.		
Atributos		
Nombre	Tipo	Descripción
Id	Guid	Identificador de la entidad.

Tabla 121. Diseño de clases: Clase abstracta DtoBase

Subsistema de identificación

Subsistema que contiene toda la lógica necesaria para la gestión de la autenticación y la gestión de usuarios. Se corresponde con el subsistema descrito en el apartado de Análisis de [Subsistema de identificación](#).

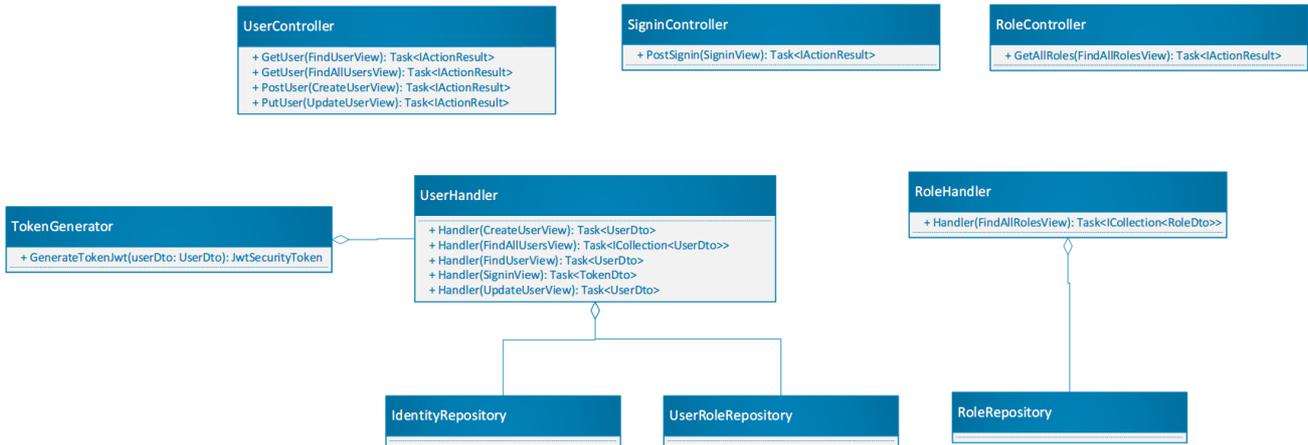


Fig 45. Diseño de clases: Subsistema de identificación

Clase		
UserController		
Descripción		
Clase que actúa como <i>controller</i> para gestión de un usuario.		
Métodos		
Nombre	Parámetro	Descripción
GetUser	FindUserView	Permite consultar un usuario.
GetUser	FindAllUsersView	Permite obtener la lista de usuarios.
PostUser	CreateUserView	Permite crear un nuevo usuario.
PutUser	UpdateUserView	Permite actualizar un nuevo usuario.

Tabla 122. Diseño de clases: Clase UserController

Clase		
SigninController		
Descripción		
Clase que actúa como <i>controller</i> para permitir iniciar sesión en el sistema.		
Métodos		
Nombre	Parámetro	Descripción
PostSignin	SigninView	Permite realizar una petición de inicio de sesión.

Tabla 123. Diseño de clases: Clase SigninController

Clase		
RoleHandler		
Descripción		
Clase que actúa como <i>controller</i> para gestionar los roles.		
Métodos		
Nombre	Parámetro	Descripción

GetAllRoles	FindAllRolesView	Permite obtener la lista de roles del sistema.
-------------	------------------	--

Tabla 124. Diseño de clases: Clase RoleController

Clase		
TokenGenerator		
Descripción		
Clase que permite generar un token JWT.		
Métodos		
Nombre	Return	Descripción
GenerateTokenJwt	JwtSecurityToken	Permite generar un token JWT dado unos parámetros dados.

Tabla 125. Diseño de clases: Clase TokenGenerator

Clase	
UserHandler	
Descripción	
Clase que contiene la lógica necesaria para realizar la gestión de usuarios.	
Métodos	
Parámetro	Descripción
CreateUserView	Lógica de negocio de crear un usuario.
FindAllUsersView	Lógica de negocio de listar todos los usuarios del sistema.
FindUserView	Lógica de negocio de consultar un usuario.
SignInView	Lógica de negocio de inicio de sesión.
UpdateUserView	Lógica de negocio de actualizar un usuario.

Tabla 126. Diseño de clases: Clase UserHandler

Clase	
RoleHandler	
Descripción	
Clase que contiene la lógica necesaria para realizar la gestión de roles de un usuario.	
Métodos	
Parámetro	Descripción
FindAllRolesView	Lógica de negocio de listar todos los roles disponibles en el sistema.

Tabla 127. Diseño de clases: Clase RoleHandler

Subsistema de visiones

Subsistema que contiene toda la lógica necesaria para la gestión de las visiones. Se corresponde con el subsistema descrito en el apartado de Análisis de [Subsistema de visiones](#).

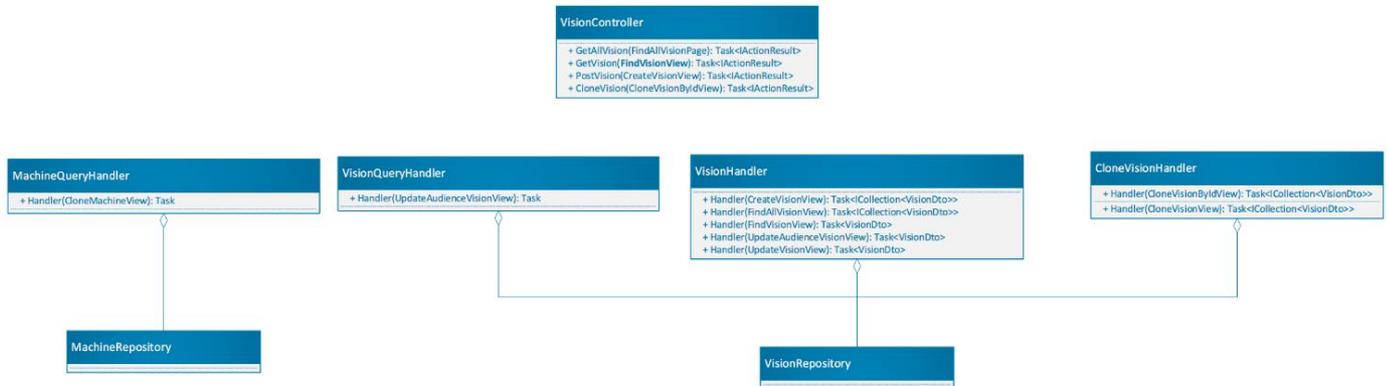


Fig 46. Diseño de clases: Subsistema de visiones

Clase		
VisionController		
Descripción		
Clase que actúa como <i>controller</i> para gestión de una visión.		
Métodos		
Nombre	Parámetro	Descripción
GetAllVision	FindAllVisionView	Permite consultar una visión.
GetVision	FindVisionView	Permite obtener la lista de visiones.
PostVision	CreateVisionView	Permite crear una nueva visión.
CloneVision	CloneVisionByIdView	Permite clonar una visión
PutVision	UpdateVisionView	Permite actualizar una visión.

Tabla 128. Diseño de clases: Clase VisionController

Clase	
MachineQueryHandler	
Descripción	
Clase que contiene la lógica necesaria para controlar eventos relativos a máquinas.	
Métodos	
Parámetro	Descripción
CloneMachineView	Lógica de negocio para mandar un evento de clonación de máquinas

Tabla 129. Diseño de clases: Clase MachineQueryHandler

Clase	
VisionQueryHandler	
Descripción	

Clase que contiene la lógica necesaria para controlar eventos relativos a visiones.	
Métodos	
Parámetro	Descripción
UpdateAudienceVisionView	Lógica de negocio para actualizar los datos de auditoria de una visión.

Tabla 130. Diseño de clases: Clase VisionQueryHandler

Clase	
VisionHandler	
Descripción	
Clase que contiene la lógica necesaria para realizar la gestión de visiones.	
Métodos	
Parámetro	Descripción
CreateVisionView	Lógica de negocio de crear una visión.
FindAllVisionsView	Lógica de negocio de listar todas las visiones del sistema.
FindVisionView	Lógica de negocio de consultar una visión.
UpdateAudienceVisionView	Lógica de negocio de actualizar los datos de auditoria.
UpdateVisionView	Lógica de negocio de actualizar una visión

Tabla 131. Diseño de clases: Clase VisionHandler

Clase	
CloneVisionHandler	
Descripción	
Clase que contiene la lógica necesaria para realizar la clonación de visiones.	
Métodos	
Parámetro	Descripción
CloneVisionByIdView	Lógica de negocio de clonar una visión dada una ID.
CloneVisionView	Lógica de negocio de clonar una visión dado la entidad original y una nueva.

Tabla 132. Diseño de clases: Clase CloneVisionHandler

Subsistema de máquinas

Subsistema que contiene toda la lógica necesaria para la gestión de máquinas. Se corresponde con el subsistema descrito en el apartado de Análisis de [Subsistema de máquinas](#).

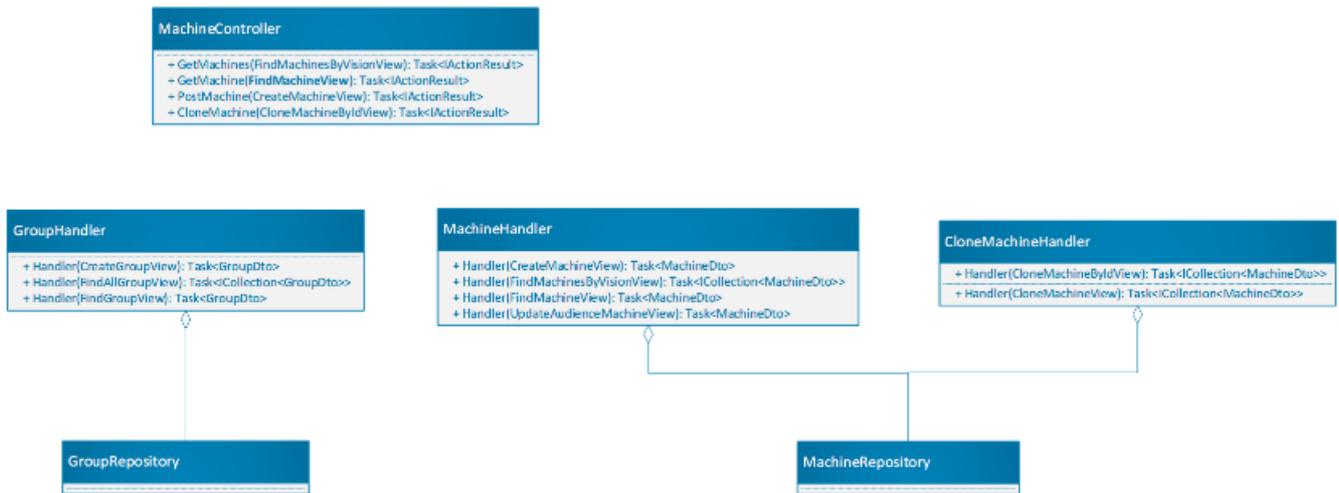


Fig 47. Diseño de clases: Subsistema de máquinas (I)

Clase		
MachineController		
Descripción		
Clase que actúa como <i>controller</i> para gestión de una máquina.		
Métodos		
Nombre	Parámetro	Descripción
GetMachines	FindMachinesByVisionView	Permite consultar la lista de máquinas de una visión.
GetMachine	FindMachineView	Permite obtener una máquina.
PostMachine	CreateMachineView	Permite crear una nueva máquina.
CloneMachine	CloneMachineByIdView	Permite clonar una máquina.

Tabla 133. Diseño de clases: Clase MachineController

Clase	
GroupHandler	
Descripción	
Clase que contiene la lógica necesaria para realizar la gestión de agrupaciones de una máquina.	
Métodos	
Parámetro	Descripción
CreateGroupView	Lógica de negocio de crear una agrupación.
FindGroupView	Lógica de negocio de listar todas las agrupaciones.
FindGroupView	Lógica de negocio de consultar una agrupación.
UpdateAudienceVisionView	Lógica de negocio de actualizar los datos de auditoria.

Tabla 134. Diseño de clases: Clase GroupHandler

Clase	
MachineHandler	

Descripción	
Clase que contiene la lógica necesaria para realizar la gestión una máquina.	
Métodos	
Parámetro	Descripción
CreateMachineView	Lógica de negocio de crear una máquina.
FindMachinesByVisionView	Lógica de negocio de listar todas las máquinas dada una visión.
FindMachineView	Lógica de negocio de consultar una máquina.
UpdateAudienceVisionView	Lógica de negocio de actualizar los datos de auditoria.

Tabla 135. Diseño de clases: Clase MachineHandler

Clase	
CloneMachineHandler	
Descripción	
Clase que contiene la lógica necesaria para realizar la clonación de máquinas.	
Métodos	
Parámetro	Descripción
CloneMachineByIdView	Lógica de negocio de clonar una máquina dada una ID.
CloneMachineView	Lógica de negocio de clonar una máquina dado la entidad original y una nueva.

Tabla 136. Diseño de clases: Clase CloneMachineHandler

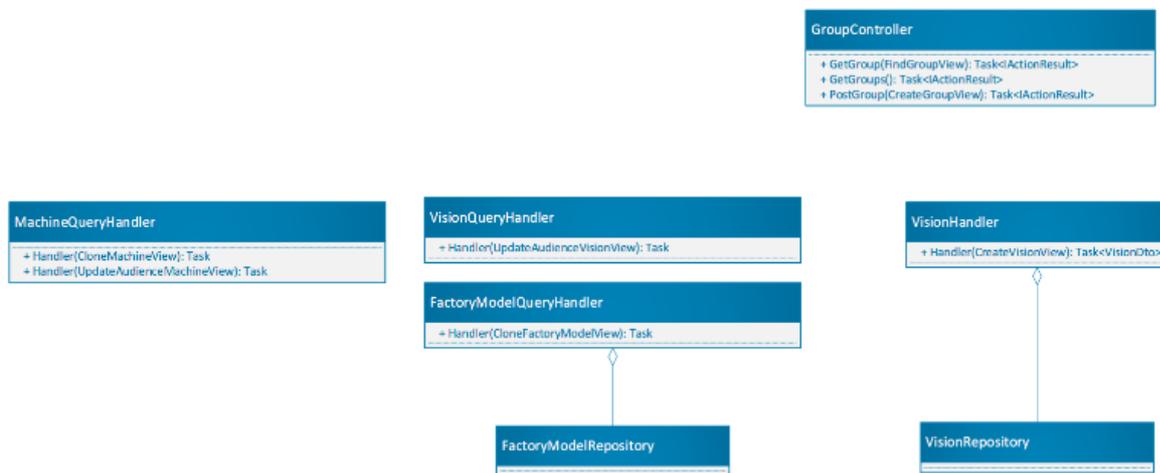


Fig 48. Diseño de clases: Subsistema de máquinas (II)

Clase		
GroupController		
Descripción		
Clase que actúa como <i>controller</i> para gestión de una agrupación de una máquina.		
Métodos		
Nombre	Parámetro	Descripción

GetGroups		Permite obtener una lista de las agrupaciones del sistema.
GetGroup	FindGroupView	Permite consultar una agrupación.
PostVision	CreateGroupView	Permite crear una nueva agrupación.

Tabla 137. Diseño de clases: Clase GroupController

Clase	
MachineQueryHandler	
Descripción	
Clase que contiene la lógica necesaria para controlar eventos relativos a máquinas.	
Métodos	
Parámetro	Descripción
CloneMachineView	Lógica de negocio para mandar un evento de clonación de máquinas.
UpdateAudienceMachineView	Lógica de negocio para actualizar los datos de auditoria de una máquina.

Tabla 138. Diseño de clases: Clase MachineQueryHandler

Clase	
VisionQueryHandler	
Descripción	
Clase que contiene la lógica necesaria para controlar eventos relativos a visiones.	
Métodos	
Parámetro	Descripción
UpdateAudienceVisionView	Lógica de negocio para actualizar los datos de auditoria de una visión.

Tabla 139. Diseño de clases: Clase VisionQueryHandler

Clase	
FactoryModelQueryHandler	
Descripción	
Clase que contiene la lógica necesaria para controlar eventos relativos a FactoryModels.	
Métodos	
Parámetro	Descripción
CloneFactoryModelView	Lógica de negocio para mandar un evento de clonación de FactoryModels.

Tabla 140. Diseño de clases: Clase FactoryModelQueryHandler

Clase
VisionHandler

Descripción	
Clase que contiene la lógica necesaria para realizar la gestión de visiones.	
Métodos	
Parámetro	Descripción
CreateVisionView	Lógica de negocio de crear una visión.

Tabla 141. Diseño de clases: Clase VisionHandler

Subsistema de FactoryModels

Subsistema que contiene toda la lógica necesaria para la gestión de FactoryModels. Se corresponde con el subsistema descrito en el apartado de Análisis de [Subsistema de FactoryModels](#).

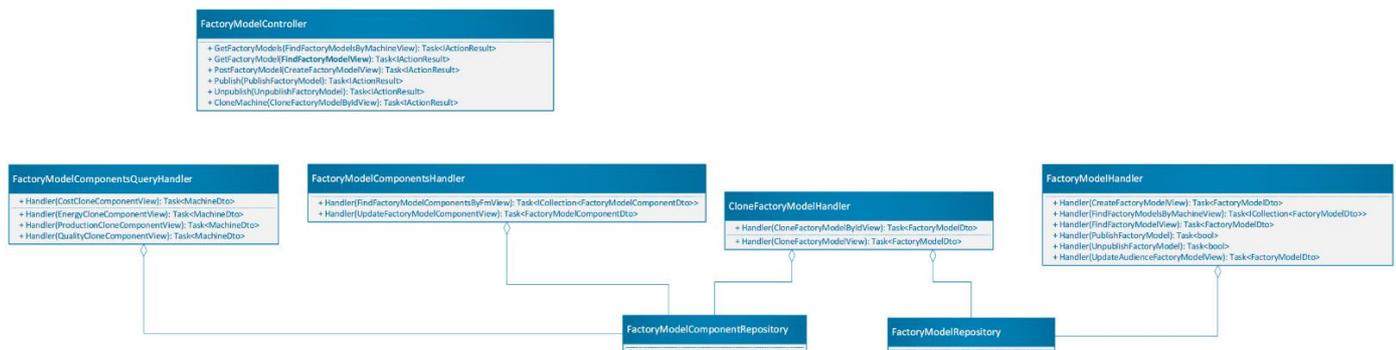


Fig 49. Diseño de clases: Subsistema de FactoryModels (I)

Clase		
FactoryModelController		
Descripción		
Clase que actúa como <i>controller</i> para gestión de un FactoryModel.		
Métodos		
Nombre	Parámetro	Descripción
GetFactoryModels	FindFactoryModelsByMachineView	Permite consultar la lista de FactoryModel de una máquina.
GetFactoryModel	FindFactoryModelView	Permite obtener un FactoryModel.
PostFactoryModel	CreateFactoryModelView	Permite crear un nuevo FactoryModel.
PublishFactoryModelView	PublishFactoryModelView	Permite cambiar el estado a Publicado un FactoryModel.
UnpublishFactoryModelView	UnpublishFactoryModelView	Permite cambiar el estado a Borrador un FactoryModel.
CloneMachine	CloneFactoryModelByIdView	Permite clonar una máquina.

Tabla 142. Diseño de clases: Clase FactoryModelController

Clase	
FactoryModelComponentsQueryHandler	
Descripción	
Clase que contiene la lógica necesaria para controlar eventos relativos a componentes de FactoryModel.	
Métodos	
Parámetro	Descripción
CostCloneComponentView	Lógica de negocio para mandar un evento de clonación de un componente de costes.
EnergyCloneComponentView	Lógica de negocio para mandar un evento de clonación de un componente de energía.
ProductionCloneComponentView	Lógica de negocio para mandar un evento de clonación de un componente de producción.
QualityCloneComponentView	Lógica de negocio para mandar un evento de clonación de un componente de calidad.

Tabla 143. Diseño de clases: Clase FactoryModelComponentsQueryHandler

Clase	
FactoryModelComponentsHandler	
Descripción	
Clase que contiene la lógica necesaria para realizar la gestión una máquina.	
Métodos	
Parámetro	Descripción
FindFactoryModelComponentsByFmView	Lógica de negocio de listar todos los componentes de FactoryModels dada un FactoryModel.
UpdateFactoryModelComponentView	Lógica de negocio de actualizar un componente de un FactoryModel.

Tabla 144. Diseño de clases: Clase FactoryModelComponentsHandler

Clase	
CloneFactoryModelHandler	
Descripción	
Clase que contiene la lógica necesaria para realizar la clonación de FactoryModel.	
Métodos	
Parámetro	Descripción
CloneFactoryModelByIdView	Lógica de negocio de clonar un FactoryModel dada una ID.
CloneFactoryModelView	Lógica de negocio de clonar un FactoryModel dado la entidad original y una nueva.

Tabla 145. Diseño de clases: Clase CloneFactoryModelHandler

Clase	
FactoryModelHandler	
Descripción	
Clase que contiene la lógica necesaria para realizar la gestión un FactoryModel.	
Métodos	
Parámetro	Descripción
CreateFactoryModelView	Lógica de negocio de crear un FactoryModel.
FindFactoryModelsByMachineView	Lógica de negocio de listar todos los FactoryModel dada una máquina.
FindFactoryModelView	Lógica de negocio de consultar un FactoryModel.
PublishFactoryModel	Lógica de negocio para cambiar el estado a Publicado un FactoryModel.
UnpublishFactoryModel	Lógica de negocio para cambiar el estado a Borrador un FactoryModel.
UpdateAudienceFactoryModelView	Lógica de negocio de actualizar los datos de auditoria.
UpdateFactoryModelView	Lógica de negocio de actualizar un FactoryModel.

Tabla 146. Diseño de clases: Clase MachineHandler

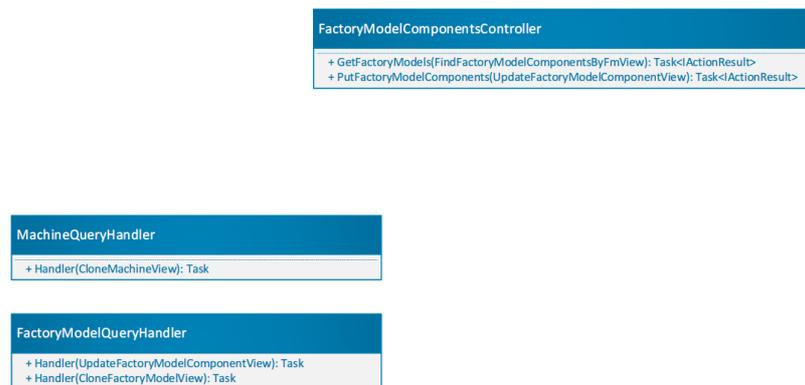


Fig 50. Diseño de clases: Subsistema de FactoryModels (II)

Clase		
FactoryModelComponentsController		
Descripción		
Clase que actúa como <i>controller</i> para gestión de un componente de un FactoryModel.		
Métodos		
Nombre	Parámetro	Descripción
GetFactoryModels	FindFactoryModelComponentsByFmView	Permite consultar la lista de FactoryModels dada una máquina.

PutFactoryModelComponents	UpdateFactoryModelComponentView	Permite actualizar un FactoryModel dado.
---------------------------	---------------------------------	--

Tabla 147. Diseño de clases: Clase FactoryModelComponentsController

Clase	
MachineQueryHandler	
Descripción	
Clase que contiene la lógica necesaria para controlar eventos relativos a máquinas.	
Métodos	
Parámetro	Descripción
CloneMachineView	Lógica de negocio para mandar un evento de clonación de máquinas.

Tabla 148. Diseño de clases: Clase MachineQueryHandler

Clase	
FactoryModelQueryHandler	
Descripción	
Clase que contiene la lógica necesaria para controlar eventos relativos a FactoryModels.	
Métodos	
Parámetro	Descripción
CloneFactoryModelView	Lógica de negocio para mandar un evento de clonación de FactoryModels.
UpdateAudienceFactoryModelView	Lógica de negocio para actualizar los datos de auditoria de un FactoryModel.

Tabla 149. Diseño de clases: Clase FactoryModelQueryHandler

Subsistemas de componentes de FactoryModel

Los siguientes diagramas de clases son válidos para los subsistemas cuyo objetivo es mantener un componente de un FactoryModel. Se tomará como referencia el componente de costes para la explicación de las clases principales. Los siguientes subsistemas analizados en el Capítulo 2 harán uso de este diseño:

- [Subsistema de costes](#)
- [Subsistema de energía](#)
- [Subsistema de producción](#)
- [Subsistema de calidad](#)



Fig 51. Diseño de clases: Subsistema de componentes de FactoryModel (I)

Clase		
EchoController		
Descripción		
Clase que actúa como <i>controller</i> para comprobar que el servicio está activo.		
Métodos		
Nombre	Parámetro	Descripción
GetEcho		Permite comprobar el estado del servicio.

Tabla 150. Diseño de clases: Clase EchoController

Clase		
CostController		
Descripción		
Clase que actúa como <i>controller</i> para gestión de los costes.		
Métodos		
Nombre	Parámetro	Descripción
Refresh	RefreshCostComponentView	Permite actualizar los cálculos de una entidad dada.

Tabla 151. Diseño de clases: Clase CostController

Clase		
CostComponentController		
Descripción		
Clase que actúa como <i>controller</i> para gestión de un componente de costes.		
Métodos		
Nombre	Parámetro	Descripción
GetCostComponents	FindCostComponentByFactoryModelView	Permite consultar la lista de componentes costes.
GetCost	FindCostComponentView	Permite obtener un componente de costes.
PostCost		Permite crear un nuevo componente de costes.
PutCost	CostComponentView	Permite actualizar un componente de costes.

Tabla 152. Diseño de clases: Clase CostComponentController



Fig 52. Diseño de clases: Subsistema de componentes de FactoryModel (II)

Clase	
FactoryModelComponentQueryHandler	
Descripción	
Clase que contiene la lógica necesaria para controlar eventos relativos a componentes de FactoryModel.	
Métodos	
Parámetro	Descripción
CostCloneComponentView	Lógica de negocio para mandar un evento de clonación de componente de costes.

Tabla 153. Diseño de clases: Clase FactoryModelComponentQueryHandler

Clase	
FactoryModelQueryHandler	
Descripción	
Clase que contiene la lógica necesaria para controlar eventos relativos a FactoryModels.	
Métodos	
Parámetro	Descripción
UpdateAudienceFactoryModelView	Lógica de negocio para actualizar los datos de auditoria de un FactoryModel.

Tabla 154. Diseño de clases: Clase FactoryModelQueryHandler

Clase	
RefreshCostHandler	
Descripción	
Clase que contiene la lógica necesaria para realizar la gestión una máquina.	
Métodos	
Parámetro	Descripción
RefreshCostComponentView	Lógica de negocio para actualizar los cálculos de una entidad dada.

Tabla 155. Diseño de clases: Clase RefreshCostHandler

Clase	
FactoryModelHandler	
Descripción	
Clase que contiene la lógica necesaria para realizar la gestión un FactoryModel.	
Métodos	
Parámetro	Descripción
CreateFactoryModelView	Lógica de negocio de crear un FactoryModel.
FindFactoryModelsView	Lógica de negocio para consultar un FactoryModel.

Tabla 156. Diseño de clases: Clase FactoryModelHandler

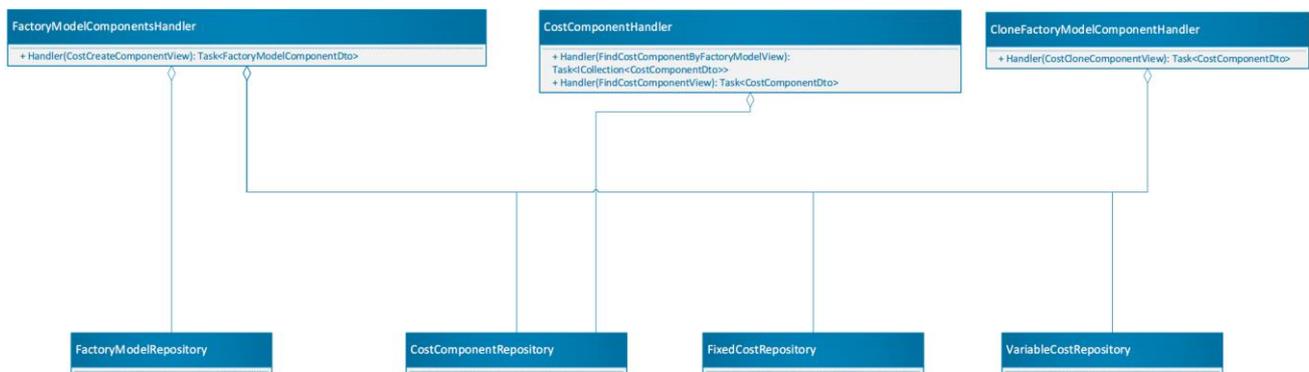


Fig 53. Diseño de clases: Subsistema de componentes de FactoryModel (III)

Clase	
FactoryModelComponentsHandler	
Descripción	
Clase que contiene la lógica necesaria para realizar la gestión de componentes de FactoryModel.	
Métodos	
Parámetro	Descripción
CostCreateComponentView	Lógica de negocio para crear un componente de costes.

Tabla 157. Diseño de clases: Clase FactoryModelComponentsHandler

Clase	
CostComponentHandler	
Descripción	
Clase que contiene la lógica necesaria para realizar la gestión un componente de costes.	
Métodos	
Parámetro	Descripción



FindCostComponentByFactoryModelView	Lógica de negocio de consultar todos los componentes de costes dado un FactoryModel
FindCostComponentView	Lógica de negocio para consultar un componente de costes.

Tabla 158. Diseño de clases: Clase FactoryModelHandler

Clase	
CloneFactoryModelComponentsHandler	
Descripción	
Clase que contiene la lógica necesaria para realizar la clonación de componentes de FactoryModel.	
Métodos	
Parámetro	Descripción
CostCloneComponentView	Lógica de negocio de clonar un componente de costes dado la entidad original y una nueva.

Tabla 159. Diseño de clases: Clase CloneFactoryModelComponentsHandler

DSI 5: DISEÑO DE LA ARQUITECTURA DE MÓDULOS DEL SISTEMA

Diagramas de Paquetes

Dada la naturaleza que tiene el proyecto donde cada módulo funciona independiente al resto, no existe un orden establecido de los paquetes que forman el proyecto. Aun así, muchos de ellos, realizan tareas similares, por lo que tienen una estructura jerárquica similar entre ellos. Se hace uso de una arquitectura basada en el patrón CQRS donde se delega la capa de servicios a un *Handler*.

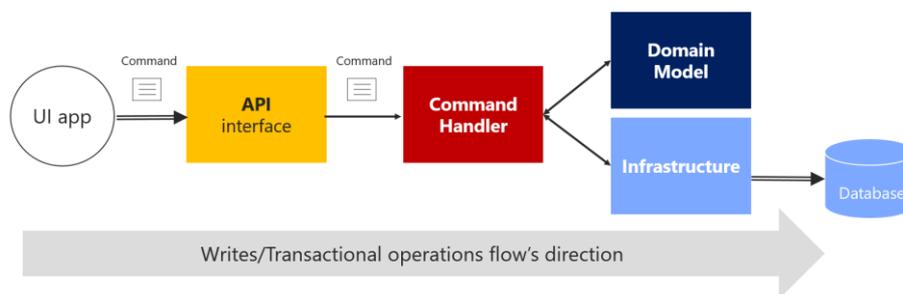


Fig 54. Arquitectura basada en el patrón CQRS [14]

Se explicarán los paquetes de los que dispone un módulo general, y en su defecto, si existiera alguna excepción, ese paquete.

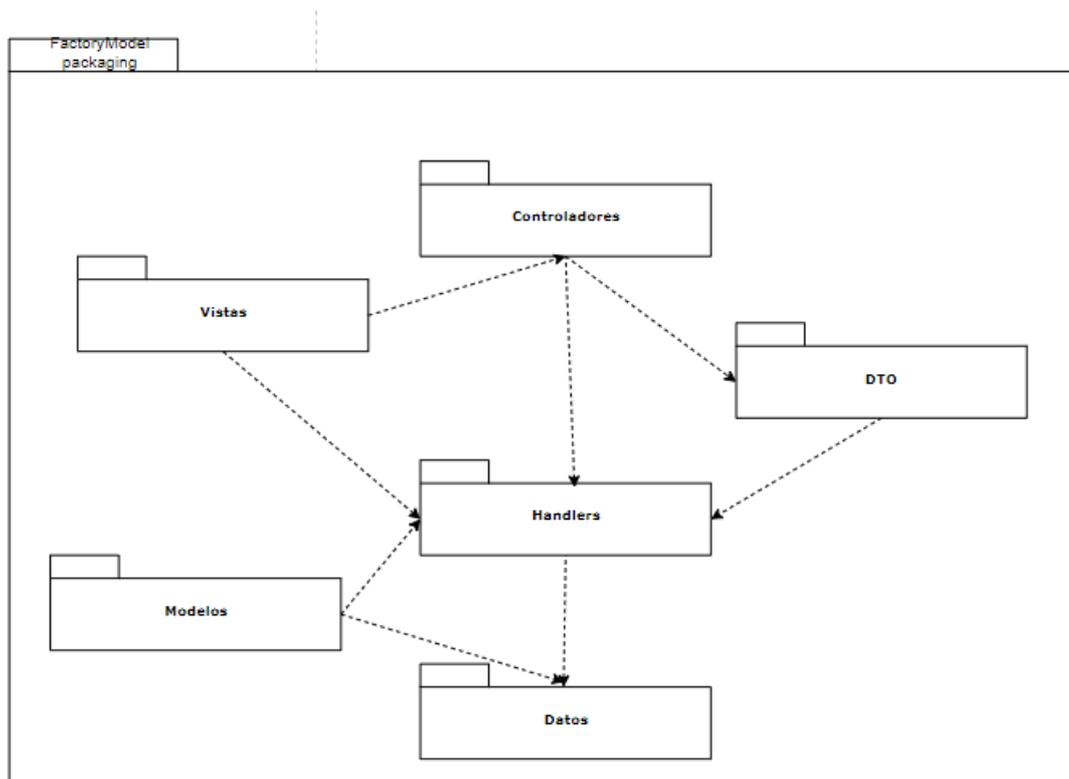


Fig 55. Diagrama de paquetes de un módulo



Controladores

Este paquete almacenará todos los controladores HTTP que se usarán para dar el servicio. Deben de seguir la siguiente regla: Nombre acabado en *Controller*.

Su aplicación será el punto de acceso por parte del cliente al servicio. Deberá de restringir los accesos a los permisos necesarios, todos los puntos de acceso serán asíncronos y podrán tratar solo con ficheros JSON.

Se encargarán de controlar que una petición que ha sido efectuada de forma correcta y reenviarán la petición a la capa de servicios.

Datos

Este paquete almacenará todo lo relativo con los contextos de la base de datos. Incluirá todos los componentes necesarios para el correcto funcionamiento (repositorios) y posibles inicializadores de base de datos.

Este paquete será el único que puede interactuar con el contexto generado por la base de datos y no deberá efectuar ninguna operación de lógica de negocio.

Data-Transfer-Object

Paquete que almacena las entidades que serán tratadas en la transición entre capas. Deben de ser entidades públicas que no dispongan de ningún tipo de lógica en su interior. Normalmente, existirá una entidad por entidad del dominio.

Handlers

Paquete que almacena la lógica de negocio de forma asíncrona. Será necesario que haga uso de paso por mensajes de forma asíncrona. Dispondrá de una operación por cada tarea necesaria por los controladores o la respuesta a eventos de mensajes. Hará uso de un patrón CQRS para hacer una separación entre acciones y consultas.

En caso de intentar controlar un evento del que no se dispone, se controlara una excepción.

Modelos

Paquete que almacenará los modelos con los que interactúa la base de datos. Una entidad completa vendrá definida como un objeto que tiene las necesidades de auditarse. Podrá considerarse como entidad basada en referencias las derivadas de otro módulo que no almacenan datos.

Vistas

Paquete que almacenará las entidades con las que tratará el paquete de lógica de negocio. A través de controladores o eventos de mensajes, se tratarán y se distribuirán de forma asíncrona al servicio que sepa controlarlo.

Diagrama de Arquitectura Modular

Una vez definidos los módulos por los que está compuesto el sistema, se pasará a mostrar la **arquitectura cloud** necesaria para ejecutar la aplicación:

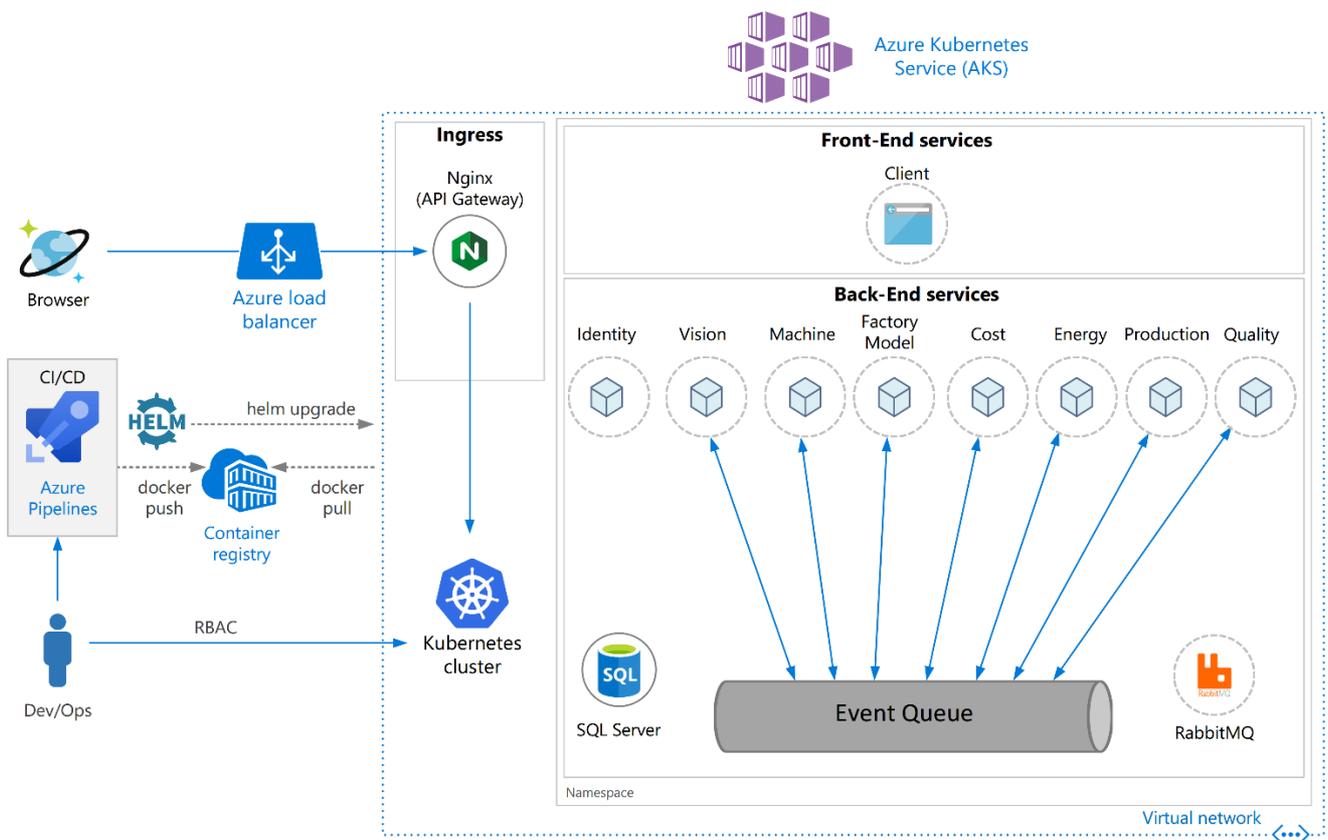


Fig 56. Diagrama de Arquitectura Cloud

Diagramas de Despliegue y Componentes

El sistema será desplegado sobre un servidor **Kubernetes** que contendrá todos los servicios necesarios para el correcto funcionamiento. El cliente web hará llamadas a un servidor proxy inverso **Nginx** que permitirá realizar peticiones a los microservicios sin conocer información del despliegue de cada microservicio.

Aunque no se considera el módulo del cliente como un microservicio, se desplegará sobre el mismo servidor.

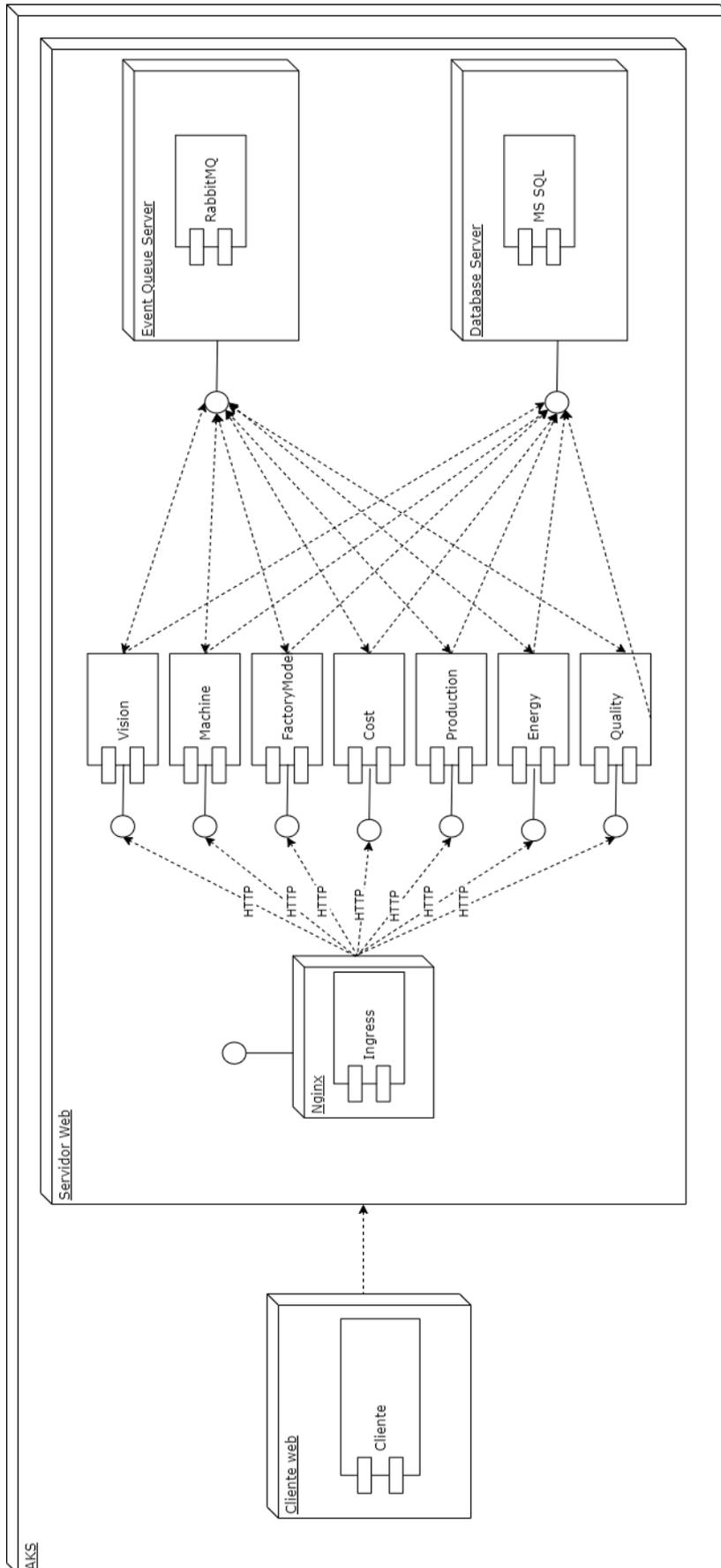


Fig 57. Diagrama de despliegue



DSI 6: DISEÑO FÍSICO DE DATOS

Descripción del SGBD usado

El motor de base de datos usado en la aplicación será **Microsoft SQL Server** en su versión 2017. Se ha decidido usar esta versión por considerar como una versión lo suficientemente estable para un uso comercial. Aunque en la actualidad, el sistema no hará uso de diversos servidores de base de datos, deberá estar perfectamente adaptado para usar servidores independientes entre los microservicios.

Dada la naturaleza que tiene el proyecto, cada microservicio interactuará únicamente con su base de datos, pudiendo existir **N bases de datos** tantas como microservicios disponga el sistema.

Integración del SGBD en nuestro sistema

Para la integración del sistema de base de datos, se va a hacer uso de **Entity Framework** en su versión Core.

Entity Framework es un conjunto de tecnologías que nos permiten hacer un desarrollo de aplicaciones de software orientadas a objetos. Para su correcto funcionamiento, se diseñarán unos modelos – entidades que se mapean con las tablas SQL – y contextos que almacenarán los diferentes ámbitos de uso del contexto. Actualmente, es el principal motor de ORM existente en la tecnología .NET.

Para modelar un entorno basado en capas, se ha decidido hacer uso del patrón Repositorio, que nos permite gestionar la base de datos de forma separada a la lógica. Este patrón nos permite controlar las posibles interacciones con la base de datos, permitiendo realizar operaciones comunes sin duplicidad de código. La opción por la que se ha decidido es definir unas clases abstractas en un módulo independiente que tienen todas las operaciones principales de BBDD: consultar, añadir, actualizar y borrar.

Dado el funcionamiento de .NET y Entity Framework, se pueden generar tantas consultas como sean necesarias con el sistema *IQueryable* y *LINQ*. Esta tecnología nos permite realizar consultar a la base de datos usando el lenguaje de programación principal, independientemente de la relación y las capacidades de la base de datos. La información que recuperará entrará en un ORM que convertirá el modelo en un DTO haciendo una proyección, es decir, pidiendo los campos únicamente necesarios para recuperar los datos del DTO. Esto nos aporta un sistema rápido, reutilizable y mantenible; evitando posibles errores que puedan aparecer en la fase de desarrollo, derivados por cambios continuos entre entidades y DTOs.

La capa de repositorios solo conocerá la información relativa a la base de datos y en ningún caso un posible DTO.

Diagrama E-R

Tal como se ha comentado anteriormente, cada módulo dispondrá de un sistema de base de datos, es decir, se disponen de tantos esquemas como N microservicios necesitan tener persistencia. Por lo tanto, se definirá los diagramas de E-R de los microservicios que lo necesiten.

Identity

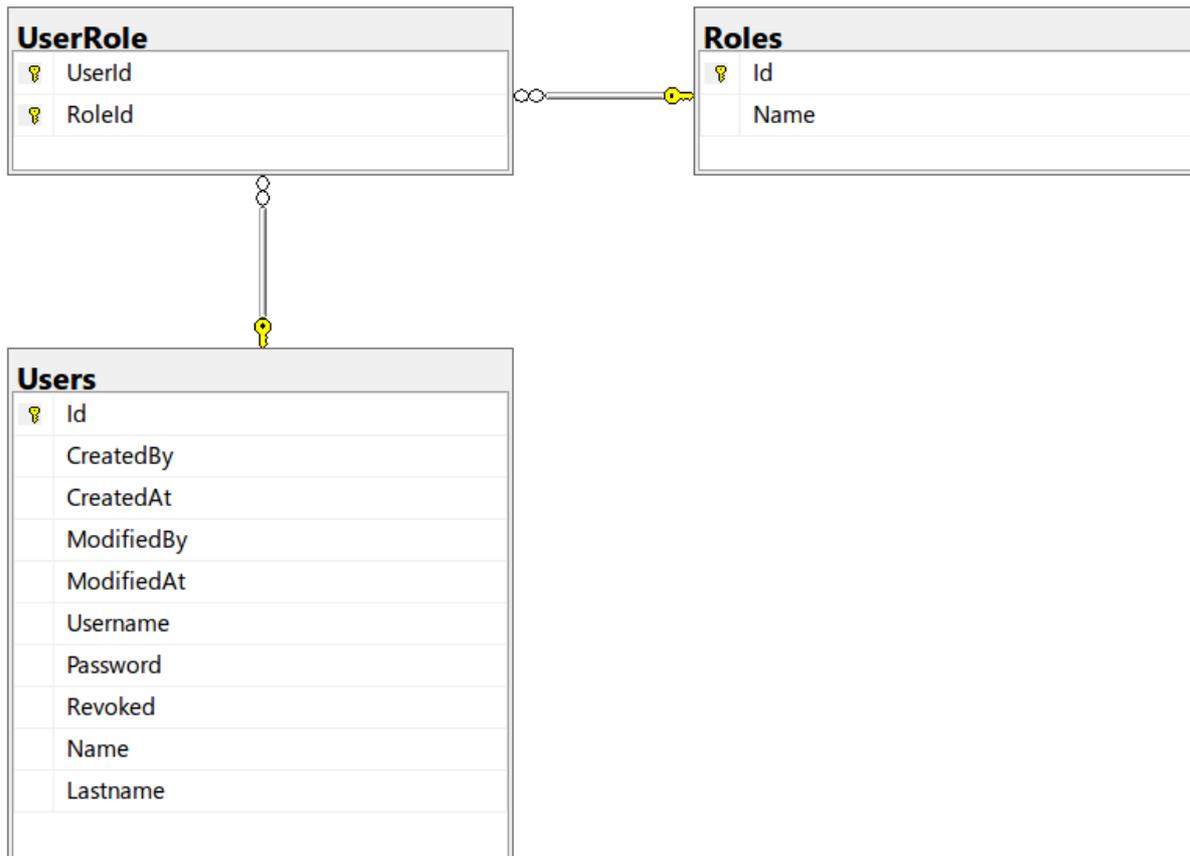


Fig 58. Diagrama E-R: Módulo de Identidad

Vision

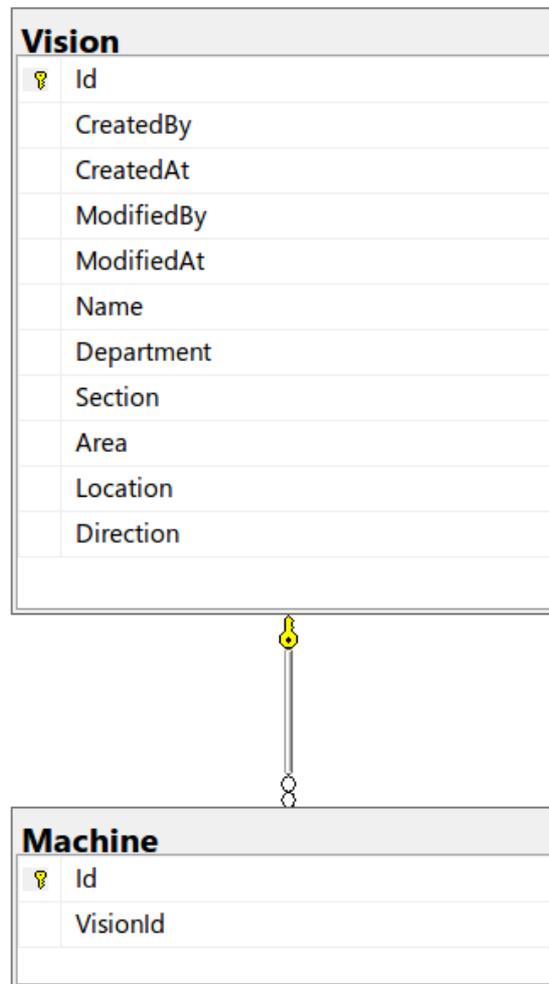


Fig 59. Diagrama E-R: Módulo de Vision

Machine

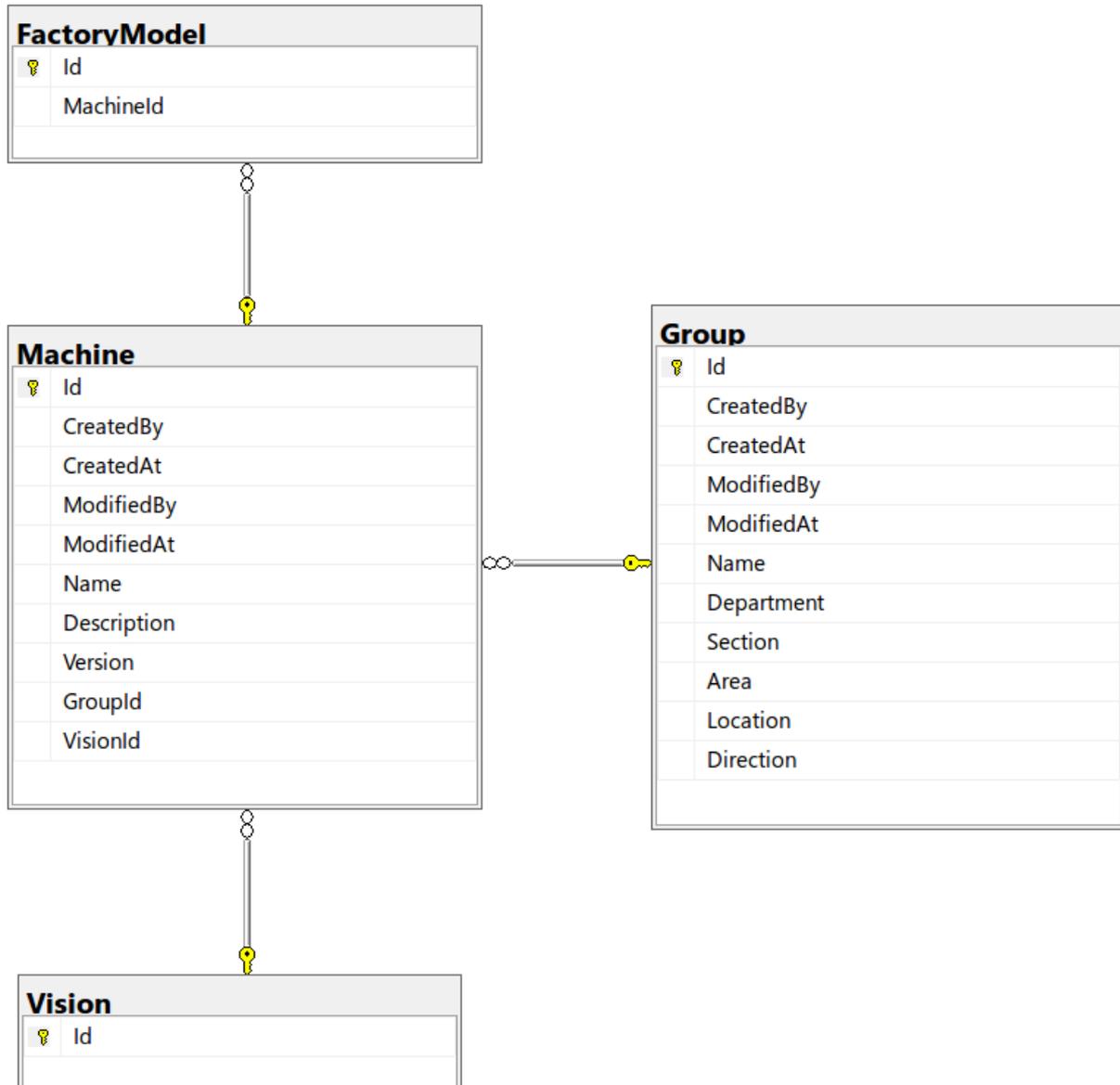


Fig 60. Diagrama E-R: Módulo de máquinas

FactoryModel

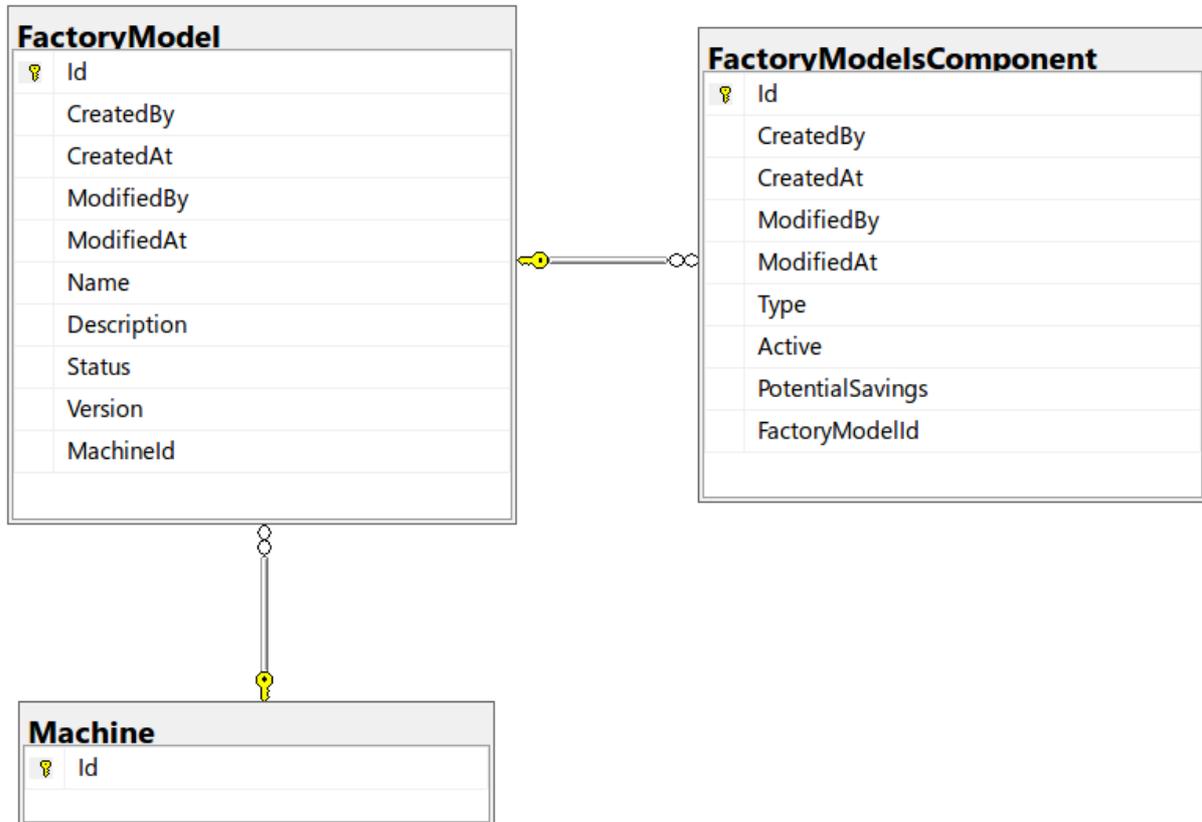


Fig 61. Diagrama E-R: Módulo de FactoryModels

Cost

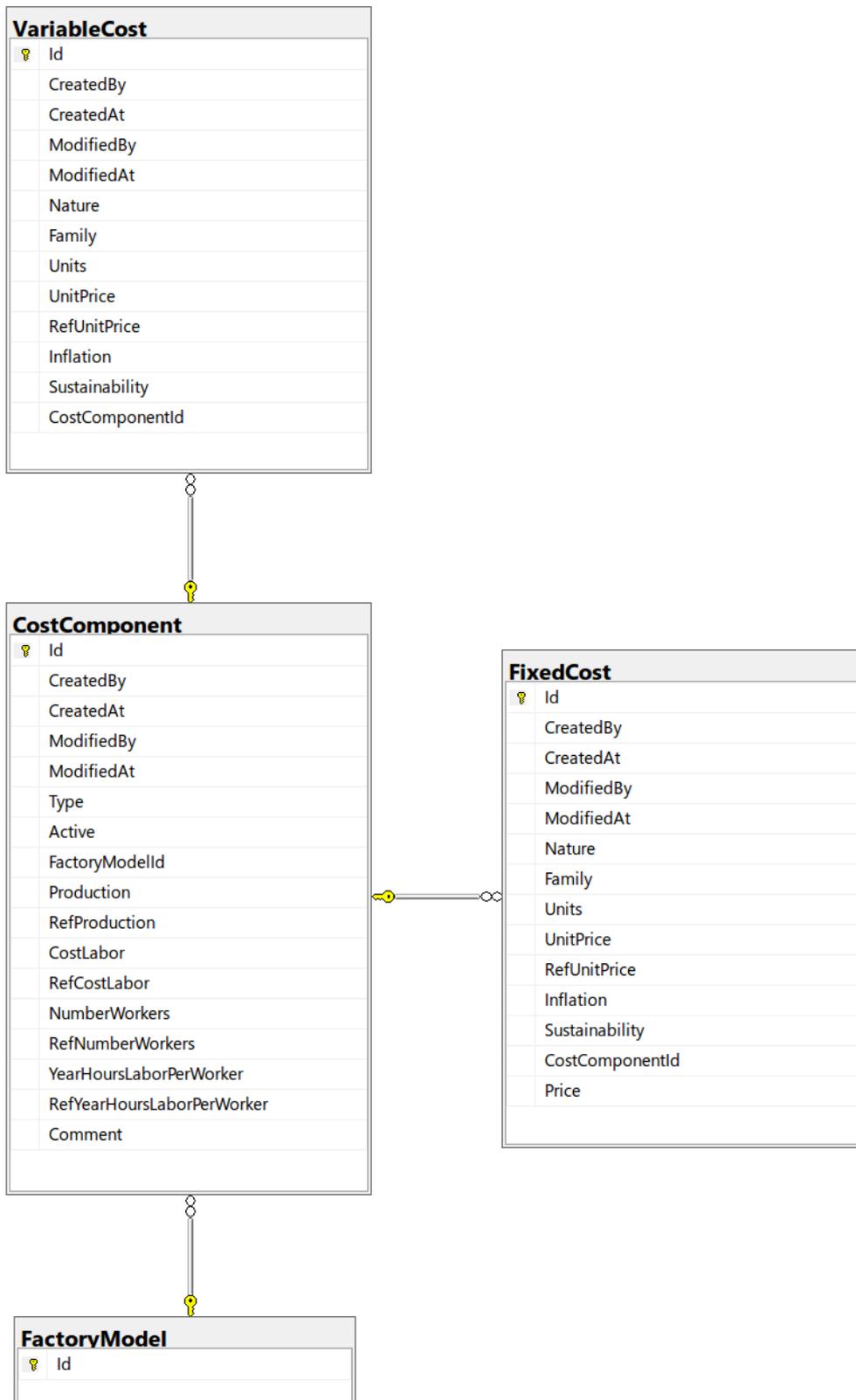


Fig 62. Diagrama E-R: Módulo de Costes

Energy

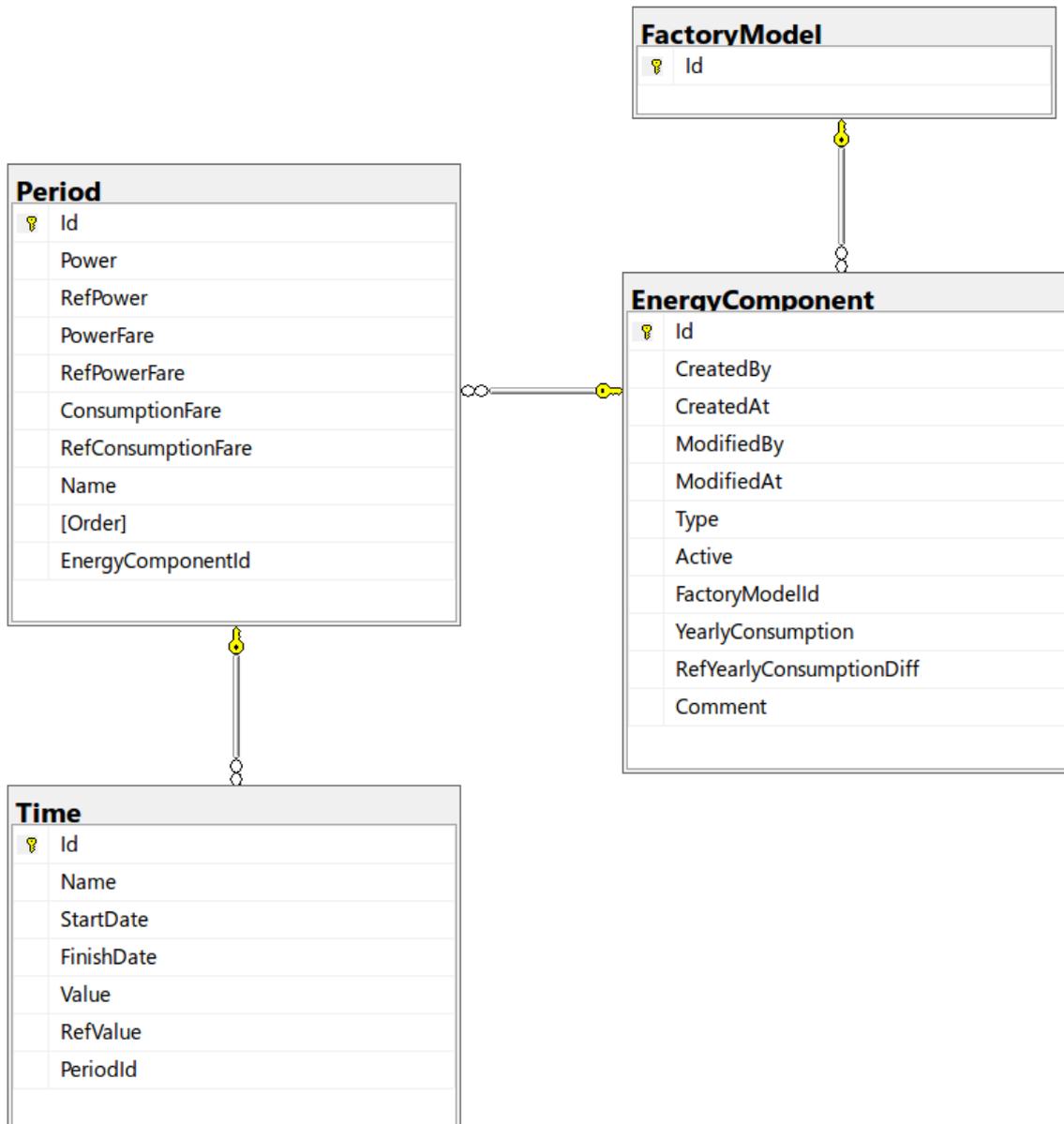


Fig 63. Diagrama E-R: Módulo de Energía

Production

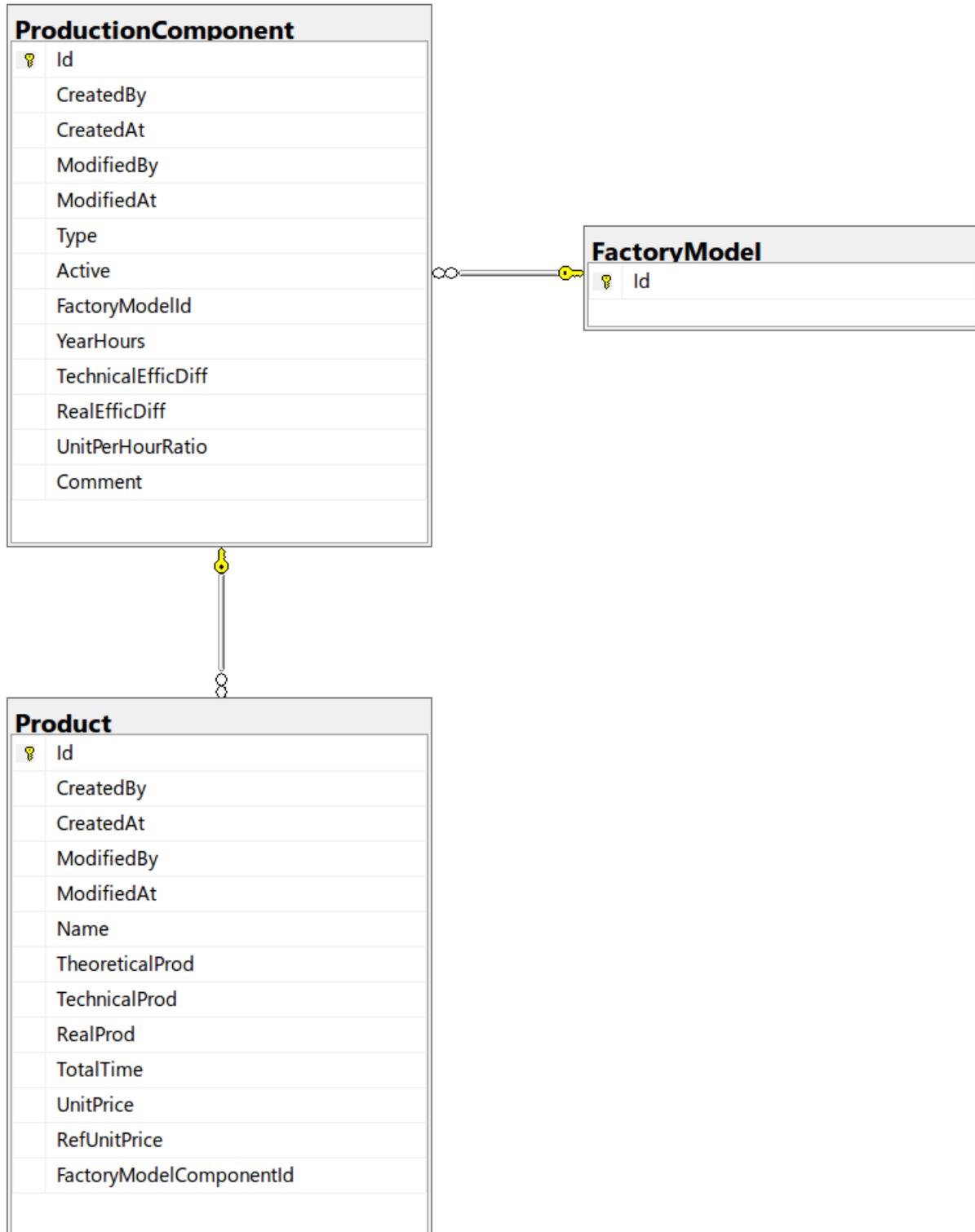


Fig 64. Diagrama E-R: Módulo de Producción

Quality

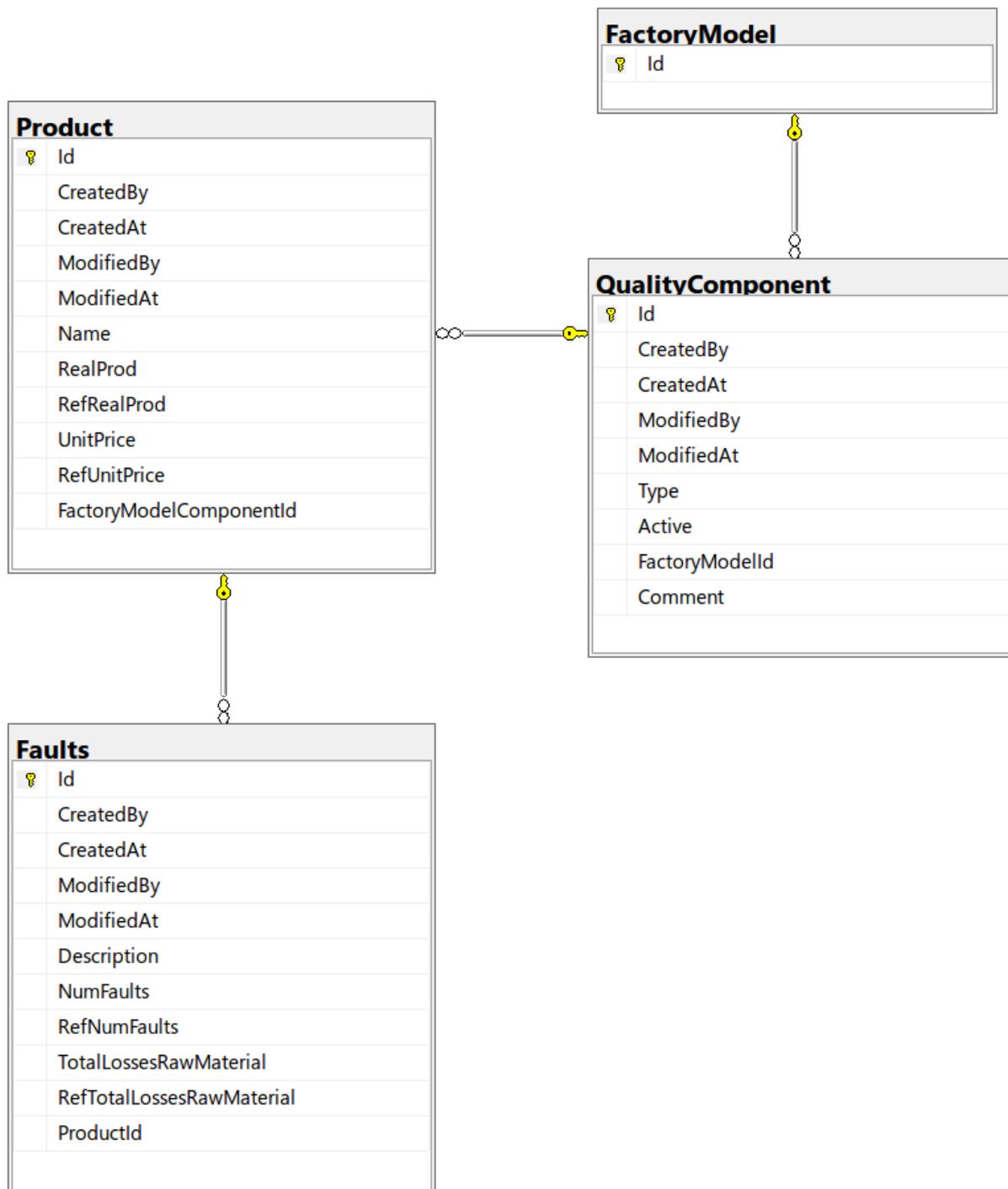


Fig 65. Diagrama E-R: Módulo de Calidad

DSI 10: ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS

En este apartado se describirán las pruebas a ejecutar sobre el sistema. Se detallarán las entradas y los resultados esperados que deben devolver las diferentes pruebas. Todas las pruebas han sido ejecutadas sobre un medio local con un sistema operativo Windows 10 2004 y sobre el sistema desplegado en **AKS** en la región de *westeurope*.

Pruebas Unitarias

Para la realización de las pruebas unitarias se creará un entorno de **integración continua** que nos permitirá ejecutar de forma automática las pruebas para comprobar que se ejecutan de forma correcta. Se alojará en una Azure Pipeline con la subida de cambios a las ramas principales del proyecto, *develop* o *master*.

Cogiendo de partida los datos definidos en el [Plan de pruebas especificado](#) anteriormente, definiremos de una forma más exacta, las entradas y salidas de los casos de prueba descritos.

Inicio de sesión

Iniciar sesión		
Prueba	Entrada	Resultado Esperado
Inicio de sesión correcto	<code>SignInView { Username = javier@seresco.es Password = "12345678" RemindMe=false }</code>	Token: válido, CurrentUser: válido, Username: válido, ValidTo: suma de fecha de arranque más el tiempo dado
Inicio de sesión correcto escogiendo recordar los datos	<code>SignInView { Username = javier@seresco.es Password = "12345678" RemindMe=true }</code>	Token: válido, CurrentUser: válido, Username: válido, ValidTo: suma de fecha de arranque más el tiempo dado
Inicio de sesión con datos erróneos	<code>SignInView { Username = javier@seresco.es Password = "12" RemindMe=false }</code>	Excepción: <i>UnauthorizedException</i>
Inicio de sesión de usuario desactivado	<code>SignInView { Username = maria@seresco.es Password = "12" RemindMe=true }</code>	Excepción: <i>UnauthorizedException</i>

Tabla 160. Diseño Pruebas Unitarias: Iniciar sesión

Gestión de visiones

Consulta de visiones		
Prueba	Entrada	Resultado Esperado
Consulta con datos	FindAllVisionPage { Page = 0 PageSize = 0 }	Resultado.Length = Contexto.Vision.Length

Tabla 161. Diseño Pruebas Unitarias: Consulta de visiones

Añadir visión		
Prueba	Entrada	Resultado Esperado
Añadir entidad correctamente	CreateVisionView { Name = "Vision" Department = "Department" Section = "Section" Area = "Area" Location = "Location" Direction = "Direction" }	Resultado = válido Id = válido Name = "Vision" Department = "Department" Section = "Section" Area = "Area" Location = "Location"
Añadir entidad sin datos	CreateVisionView	Exepción: BadRequestException

Tabla 162. Diseño Pruebas Unitarias: Añadir visión

Modificar visión		
Prueba	Entrada	Resultado Esperado
Modificar entidad que existe	UpdateVisionView { Id = Vision1 Name = "Vision" Department = "Department" Section = "Section" Area = "Area" Location = "Location" Direction = "Direction" }	Resultado = válido Id = válido Id = Vision1 Name = "Vision" Department = "Department" Section = "Section" Area = "Area" Location = "Location"
Modificar entidad que no existe	UpdateVisionView { Id = Empty } UpdateVisionView { Id = null } UpdateVisionView	Excepción: BadRequestExcepcion

Tabla 163. Diseño Pruebas Unitarias: Modificar visión

Clonar visión

Prueba	Entrada	Resultado Esperado
Clonar entidad que existe	CloneVisionByIdView { OldId = Vision1.Id }	Id = válido Id != Vision1.Id Name = Vision1.Name Department = Vision1.Department Section = Vision1.Section Area = Vision1.Area Location = Vision1.Location Direction = Vision1.Direction
Clonar entidad que existe con nuevo nombre	CloneVisionByIdView { OldId = Vision1.Id NewName = "new-name" }	Id = válido Id != Vision1.Id Name = "new-name" Department = Vision1.Department Section = Vision1.Section Area = Vision1.Area Location = Vision1.Location Direction = Vision1.Direction
Clonar entidad que no existe	CloneVisionByIdView { OldId = Empty } CloneVisionByIdView	Excepción: BadRequestException

Tabla 164. Diseño Pruebas Unitarias: Clonar visión

Gestión de máquinas

Consulta de máquinas		
Prueba	Entrada	Resultado Esperado
Consulta con datos	FindMachinesByVisionView { VisionId = Vision1.id }	Resultado.Length = Contexto.Machine.Length
Visión no válido	FindMachinesByVisionView { VisionId = Empty } FindMachinesByVisionView	Resultado.Length = 0

Tabla 165. Diseño Pruebas Unitarias: Consulta de máquinas

Añadir máquina		
Prueba	Entrada	Resultado Esperado
Añadir entidad correctamente	CreateMachineView { Name = "Máquina" Description = "Description" Version = "Version" Group = Group1 Vision = Vision1 }	Resultado = válido Id = válido Name = "Máquina" Description = "Description" Version = "Version" Group = Group1 Vision = Vision1

Añadir entidad de visión que no existe	CreateMachineView { Name = "Máquina" Description = "Description" Version = "Version" Group = Group1 }	Resultado = válido Id = válido Name = "Máquina" Description = "Description" Version = "Version" Group = Group1 Vision = válida
Añadir entidad sin datos	CreateMachine	Excepción: BadRequestException

Tabla 166. Diseño Pruebas Unitarias: Añadir máquina

Modificar máquinas		
Prueba	Entrada	Resultado Esperado
Modificar entidad que existe	UpdateMachineView { Id = Machine1 Name = "Machine 1" Description = "Description" Version = "Version" }	Resultado = válido Id = válido Id = Machine 1 Name = "Machine 1" Department = "Description" Section = "Version"
Modificar entidad que no existe	UpdateMachineView { Name = "Machine 1" Description = "Description" Version = "Version" }	Excepción: BadRequestException

Tabla 167. Diseño Pruebas Unitarias: Modificar máquinas

Clonar máquinas		
Prueba	Entrada	Resultado Esperado
Clonar entidad que existe	CloneMachineByIdView { OldId = Machine1.Id }	Id = válido Id != Machine1.Id Name = Machine1.Name Description = Machine1.Description Version = Machine1.Version Group = Machine1.Group VisionId = Machine1.VisionId FactoryModelId.Length = Machine.FactoryModelId.Length FactoryModelId.Id != FactoryModelId.Id
Clonar entidad que existe con nuevo nombre	CloneMachineByIdView { OldId = Machine1.Id NewName = "new-name" }	Id = válido Id != Machine1.Id Name = "new-name" Description = Machine1.Description Version = Machine1.Version Group = Machine1.Group VisionId = Machine1.VisionId

		FactoryModelId.Lenth = Machine.FactoryModelId.Length FactoryModelId.Id != FactoryModelId.Id
Clonar entidad que no existe	CloneMachineByIdView { OldId = Empty } CloneMachineByIdView	Excepción: BadRequestException

Tabla 168. Diseño Pruebas Unitarias: Clonar máquinas

Gestión de FactoryModel

Consulta de FactoryModels		
Prueba	Entrada	Resultado Esperado
Consulta con datos	FindFactoryModelByMachineView { MachineId = Machine1.id }	Resultado.Length = Contexto.FactoryModel.Length
Visión no válido	FindFactoryModelByMachineView { MachineId = Empty } FindFactoryModelByMachineView	Resultado.Length = 0

Tabla 169. Diseño Pruebas Unitarias: Consulta de FactoryModels

Consulta de Componentes		
Prueba	Entrada	Resultado Esperado
Consulta con datos	FindFactoryModelComponentsByFmView { FactoryModelId = FactoryModel1 }	Resultado.Length = Contexto.FactoryModelComponents(FactoryModelId = FactoryModel1.Id).Length
FactoryModel no válido	FindFactoryModelComponentsByFmView	Expepción: BadRequestException

Tabla 170. Diseño Pruebas Unitarias: Consulta de Componentes

Añadir FactoryModels		
Prueba	Entrada	Resultado Esperado
Añadir entidad correctamente	CreateFactoryModelView { Name = "FactoryModel" Description = "Description" Version = "Version" Machine = Machine1 }	Resultado = válido Id = válido Name = "FactoryModel" Description = "Description" Version = "Version" Machine = Machine
Añadir entidad de FactoryModel que no existe	CreateFactoryModelView { Name = "FactoryModel" Description = "Description"	Resultado = válido Id = válido Name = "FactoryModel" Description = "Description"

	<code>Version = "Version"</code> <code>}</code>	<code>Version = "Version"</code>
Añadir entidad sin datos	<code>CreateFactoryModelView</code>	Excepción: <code>BadRequestException</code>

Tabla 171. Diseño Pruebas Unitarias: Añadir FactoryModels

Modificar FactoryModels		
Prueba	Entrada	Resultado Esperado
Modificar entidad que existe	<code>UpdateFactoryModelView {</code> <code>Id = FactoryModel1.Id</code> <code>Name = "FactoryModel"</code> <code>Description = "Description"</code> <code>Version = "Version"</code> <code>}</code>	Resultado = válido <code>Id = válido</code> <code>Id = FactoryModel1</code> <code>Name = "FactoryModel"</code> <code>Department = "Description"</code> <code>Section = "Version"</code>
Modificar entidad que no existe	<code>UpdateFactoryModelView {</code> <code>Name = "FactoryModel"</code> <code>Description = "Description"</code> <code>Version = "Version"</code> <code>}</code>	Excepción: <code>BadRequestException</code>

Tabla 172. Diseño Pruebas Unitarias: Modificar FactoryModels

Clonar FactoryModels		
Prueba	Entrada	Resultado Esperado
Clonar entidad que existe	<code>CloneMachineByIdView {</code> <code>OldId = Machine1.Id</code> <code>}</code>	<code>Id = válido</code> <code>Id != Machine1.Id</code> <code>Name = Machine1.Name</code> <code>Description = Machine1.Description</code> <code>Version = Machine1.Version</code> <code>Group = Machine1.Group</code> <code>VisionId = Machine1.VisionId</code> <code>FactoryModelId.Length =</code> <code>Machine.FactoryModelId.Length</code> <code>FactoryModelId.Id != FactoryModelId.Id</code>
Clonar entidad que existe con nuevo nombre	<code>CloneMachineByIdView {</code> <code>OldId = Machine1.Id</code> <code>NewName = "new-name"</code> <code>}</code>	<code>Id = válido</code> <code>Id != Machine1.Id</code> <code>Name = "new-name"</code> <code>Description = Machine1.Description</code> <code>Version = Machine1.Version</code> <code>Group = Machine1.Group</code> <code>VisionId = Machine1.VisionId</code> <code>FactoryModelId.Length =</code> <code>Machine.FactoryModelId.Length</code> <code>FactoryModelId.Id != FactoryModelId.Id</code>
Clonar entidad que no existe	<code>CloneMachineByIdView {</code> <code>OldId = Empty</code> <code>}</code>	Excepción: <code>BadRequestException</code>

CloneMachineByIdView

Tabla 173. Diseño Pruebas Unitarias: Clonar FactoryModels

Gestión de usuarios

Consulta de usuarios		
Prueba	Entrada	Resultado Esperado
Consulta con datos	FindAllUsersView { Skip = 0 Take = 0 }	Resultado.Length = Contexto.Users.Length

Tabla 174. Diseño Pruebas Unitarias: Consulta de usuarios

Dar de alta a usuarios		
Prueba	Entrada	Resultado Esperado
Añadir entidad correctamente	CreateUserView { Name = Lucía, Lastname = Martínez, Password = aaaaaa, Username = lucia@casa.com, Roles = { new RoleDto { Id = Role1.Id } } }	Name = Lucia Lastname = Martínez Username = lucia@casa.com Roles.Length = 1
Añadir entidad con datos erróneos	CreateUserView { }	Exception : ConflictRequestException

Tabla 175. Diseño Pruebas Unitarias: Dar de alta a usuarios

Modificar usuario		
Prueba	Entrada	Resultado Esperado
Modificar entidad que existe	UpdateUserView { Id = User1, Name = Javier Revoked = true }	Resultado : válido Resultado.Id = válido Resultado.Name = Javier Resultado.Revoked = true
Modificar entidad que no existe	UpdateUserView { Id = Empty }	El sistema responderá con un mensaje de error.

Tabla 176. Diseño Pruebas Unitarias: Modificar usuario

Consultar detalles de un usuario		
Prueba	Entrada	Resultado Esperado

Consulta con datos	FindUserView { Id = User1.Id }	Resultado: válido
Consulta sin datos	FindUserView { Id = User2.Id }	Resultado: no válido

Tabla 177. Diseño Pruebas Unitarias: Consultar detalles de un usuario

Costes

Vista de detalles y edición de FM Costes		
Prueba	Entrada	Resultado Esperado
Consultar FM Costes de un FactoryModel	FindCostComponentByFactoryModelView { FactoryModelId: FactoryModel1.Id }	Id != no válido Id = Cost1.Id Fixed.Length = Cost1.Fixed.Length Fixed(id) = Cost1.Fixed(id) Variable.Length = Cost1.Variable.Length Variable(id) = Cost1.Fixed(id)
Consulta FM Costes de un FactoryModel que no existe	FindCostComponentByFactoryModelView { FactoryModelId: Guid.New }	Id = no válido
Actualizar información mostrada	RefreshCostComponentView { CostComponentDto: Cost1 }	Id != no válido Id = Cost1.Id Production = Cost1.Production RefProduction = Cost1.RefProduction CostLabor = Cost1.CostLabor RefCostLabor = Cost1.RefCostLabor NumberWorkers = Cost1.NumberWorkers RefNumberWorkers = Cost1.RefNumberWorkers YearHoursLaborPerWorker = Cost1.YearHoursLaborPerWorker RefYearHourLaborPerWorker = Cost1.RefYearLaborPerWorker TotalCostLabor = Cost1.TotalCostLabor RefTotalCostLabor = Cost1.RefTotalCostlabor PotentialSavings = Cost1.PotentialSavings Comment = Cost1.Comment



		Type = Cost1.Type Active = Cost1.Active FactoryModel = Cost1.FactoryModel Fixed.Length = Cost1.Fixed.Length Fixed(id) = Cost1.Fixed(id) Variable.Length = Cost1.Variable.Length Variable(id) = Cost1.Fixed(id)
Actualizar información	UpdateCostComponentView { CostComponentDto: Cost1 { Comment = "es una prueba" Fixed[0].Family = "Fixed" Variable[0].Family = "Variable" } }	Id != no válido Id != Cost1.Id Production = Cost1.Production RefProduction = Cost1.RefProduction CostLabor = Cost1.CostLabor RefCostLabor = Cost1.RefCostLabor NumberWorkers = Cost1.NumberWorkers RefNumberWorkers = Cost1.RefNumberWorkers YearHoursLaborPerWorker = Cost1.YearHoursLaborPerWorker RefYearHourLaborPerWorker = Cost1.RefYearLaborPerWorker TotalCostLabor = Cost1.TotalCostLabor RefTotalCostLabor = Cost1.RefTotalCostlabor PotentialSavings != Cost1.PotentialSavings Comment != Cost1.Comment Type = Cost1.Type Active = Cost1.Active FactoryModel = Cost1.FactoryModel Fixed.Length = Cost1.Fixed.Length Fixed(id) = Cost1.Fixed(id) Fixed(family) = Cost1.Fixed(family) Variable.Length = Cost1.Variable.Length Variable(id) = Cost1.Fixed(id) Variable(family) = Cost1.Fixed(family)



Actualizar información de entidad no válida	<pre>UpdateCostComponentView { CostComponentDto: new Dto(){ Id = Guid.New } } UpdateCostComponentView { CostComponentDto: new Dto() } UpdateCostComponentView</pre>	Exception: <i>ConflictRequestException</i>
--	---	--

Tabla 178. Diseño Pruebas Unitarias: Vista de detalles y edición de FM Costes

Creación de FM Costes		
Prueba	Entrada	Resultado Esperado
Crear FM Costes	<pre>CostCreateComponentView { FactoryModelId = FactoryModel1 Type = "Cost" Active = true }</pre>	<pre>Id != no válido Production = 0 RefProduction = 0 CostLabor = 0 RefCostLabor = 0 NumberWorkers = 0 RefNumberWorkers = 0 YearHoursLaborPerWorker = 0 RefYearHourLaborPeWorker = 0 TotalCostLabor = 0 RefTotalCostLabor = 0 PotentialSavings = 0 Comment = no válido Type = "Cost" Active = true FactoryModel = FactoryModel1 Fixed.Length = 0 Variable.Lenth = 0</pre>
Crear FM Costes existiendo uno asignado al mismo FactoryModel	<pre>CostCreateComponentView { FactoryModelId = FactoryModel1 Type = "Cost" Active = true }</pre>	<pre>Id != no válido Production = 0 RefProduction = 0 CostLabor = 0 RefCostLabor = 0 NumberWorkers = 0 RefNumberWorkers = 0 YearHoursLaborPerWorker = 0 RefYearHourLaborPeWorker = 0 TotalCostLabor = 0 RefTotalCostLabor = 0 PotentialSavings = 0 Comment = no válido Type = "Cost" Active = true</pre>

		FactoryModel = FactoryModel1 Fixed.Length = 0 Variable.Length = 0
Crear otro FM	CostCreateComponentView { FactoryModelId = FactoryModel1 Type = "aaa" Active = true } CostCreateComponentView	Excepción: BadRequestException

Tabla 179. Diseño Pruebas Unitarias: Creación de FM Costes

Clonación de FM Costes		
Prueba	Entrada	Resultado Esperado
Clonar la entidad correctamente	CostCloneComponentView { Old = Cost1, New = { Id = Guid.New, FactoryModelId = Guid.New } }	Id != no válido Id != Cost1.Id Production = Cost1.Production RefProduction = Cost1.RefProduction CostLabor = Cost1.CostLabor RefCostLabor = Cost1.RefCostLabor NumberWorkers = Cost1.NumberWorkers RefNumberWorkers = Cost1.RefNumberWorkers YearHoursLaborPerWorker = Cost1.YearHoursLaborPerWorker RefYearHourLaborPerWorker = Cost1.RefYearLaborPerWorker TotalCostLabor = Cost1.TotalCostLabor RefTotalCostLabor = Cost1.RefTotalCostlabor PotentialSavings = Cost1.PotentialSavings Comment = Cost1.Comment Type = Cost1.Type Active = Cost1.Active FactoryModel != Cost1.FactoryModel Fixed.Length = Cost1.Fixed.Length Fixed(id) != Cost1.Fixed(id) Variable.Length = Cost1.Variable.Length Variable(id) != Cost1.Fixed(id)
Clonar entidad con datos incorrectos	CostCloneComponentView { Old = { Id = Guid.New, }, New = { Id = Guid.New, }	Excepción: BadRequestException

```

FactoryModelId = Guid.New
}
}
CostCloneComponentView {
Old = {
Id = Guid.New,
FactoryModelId = Guid.New
}
}
CostCloneComponentView

```

Tabla 180. Diseño Pruebas Unitarias: Clonación de FM Costes

Energía

Vista de detalles y edición de FM Energía		
Prueba	Entrada	Resultado Esperado
Consultar FM Costes de un FactoryModel del	FindEnergyComponentByFactoryModelView { FactoryModelId: FactoryModel1.Id }	Id != no válido Id = Energy1.Id Periods.Length = Energy1.Periods.Length Periods(id) = Energy1.Periods(id)
Consulta FM Costes de un FactoryModel del que no existe	FindEnergyComponentByFactoryModelView { FactoryModelId: Guid.New }	Id = no válido
Actualizar información mostrada	RefreshEnergyComponentView { EnergyComponentDto: Energy1 }	Id = válido Id = Energy1.Id Power = Energy1.Production RefPower = Energy1.RefPower PowerDiff = Energy1.PowerDiff PowerDiffPerc = Energy1.PowerDiffPerc Consumption = Energy1.Consumption RefConsumption = Energy1.RefConsumption ConsumptionDiff = Energy1.ConsumptionDiff ConsumptionDiffPerc = Energy1.ConsumptionDiffPerc Total = Energy1.Total RefTotal = Energy1.RefTotal YearlyConsumption = Energy1.YearlyConsumption



		<p>RefYearlyConsumption = Energy1.RefYearlyConsumption PotentialSavings = Energy1.PotentialSavings PotentialImprovementProd = Energy1.PotentialImprovementProd Comment = Energy1.Comment Type = Energy1.Type Active = Energy1.Active FactoryModel = Energy1.FactoryModel Periods.Length = Energy1.Periods.Length Periods(id) = Energy1.Periods(id)</p>
<p>Actualizar información</p>	<pre>UpdateEnergyComponentView { EnergyComponentDto: Energy1 { Comment = "es una prueba" Periods[0].Name = "Variable" Periods[0].Times[0].Value = 0,2 } }</pre>	<p>Id = válido Id = Energy1.Id Power = Energy1.Production RefPower = Energy1.RefPower PowerDiff = Energy1.PowerDiff PowerDiffPerc = Energy1.PowerDiffPerc Consumption = Energy1.Consumption RefConsumption = Energy1.RefConsumption ConsumptionDiff = Energy1.ConsumptionDiff ConsumptionDiffPerc = Energy1.ConsumptionDiffPerc Total = Energy1.Total RefTotal = Energy1.RefTotal YearlyConsumption = Energy1.YearlyConsumption RefYearlyConsumption = Energy1.RefYearlyConsumption PotentialSavings = Energy1.PotentialSavings PotentialImprovementProd = Energy1.PotentialImprovementProd Comment = Energy1.Comment Type = Energy1.Type Active = Energy1.Active FactoryModel = Energy1.FactoryModel</p>

		Periods.Length = <i>Energy1.Periods.Length</i> Periods(id) = <i>Energy1.Periods(id)</i> Periods(name) != <i>Energy1.Periods(name)</i>
Actualizar información de entidad no válida	UpdateEnergyComponentView { <i>EnergyComponentDto: new Dto()</i> { <i>Id = Guid.New</i> } } UpdateEnergyComponentView { <i>EnergyComponentDto: new Dto()</i> } UpdateEnergyComponentView	Exception: <i>ConflictRequestException</i>

Tabla 181. Diseño Pruebas Unitarias: Vista detalles y edición de FM Energía

Creación de FM Energías		
Prueba	Entrada	Resultado Esperado
Crear FM Energías	EnergyCreateComponentView { <i>FactoryModelId = FactoryModel1</i> <i>Type = "Cost"</i> <i>Active = true</i> }	Id = válido Power = 0 RefPower = 0 PowerDiff = 0 PowerDiffPerc = 0 Consumption = 0 RefConsumption = 0 ConsumptionDiff = 0 ConsumptionDiffPerc = 0 Total = 0 RefTotal = 0 YearlyConsumption = 0 RefYearlyConsumption = 0 PotentialSavings = 0 PotentialImprovementProd = 0 Comment = no válido Type = Energy1.Type Active = Energy1.Active FactoryModel = <i>Energy1.FactoryModel</i> Periods.Length = 3
Crear FM Energías existiendo uno asignado al mismo FactoryModel	EnergíasCreateComponentView { } <i>FactoryModelId = FactoryModel1</i> <i>Type = "Cost"</i> <i>Active = true</i> }	Id = válido Power = 0 RefPower = 0 PowerDiff = 0 PowerDiffPerc = 0 Consumption = 0

		RefConsumption = 0 ConsumptionDiff = 0 ConsumptionDiffPerc = 0 Total = 0 RefTotal = 0 YearlyConsumption = 0 RefYearlyConsumption = 0 PotentialSavings = 0 PotentialImprovementProd = 0 Comment = no válido Type = Energy1.Type Active = Energy1.Active FactoryModel = Energy1.FactoryModel Periods.Length = 3
Crear otro FM	<pre> EnergyCreateComponentView { FactoryModelId = FactoryModel1 Type = "aaa" Active = true } EnergyCreateComponentView </pre>	Excepción: BadRequestException

Tabla 182. Diseño Pruebas Unitarias: Creación de FM Calidad

Clonación de FM Energías		
Prueba	Entrada	Resultado Esperado
Clonar la entidad correctamente	<pre> EnergyCloneComponentView { Old = Energy1, New = { Id = Guid.New, FactoryModelId = Guid.New } } </pre>	Id = válido Id != Energy1.Id Power = Energy1.Production RefPower = Energy1.RefPower PowerDiff = Energy1.PowerDiff PowerDiffPerc = Energy1.PowerDiffPerc Consumption = Energy1.Consumption RefConsumption = Energy1.RefConsumption ConsumptionDiff = Energy1.ConsumptionDiff ConsumptionDiffPerc = Energy1.ConsumptionDiffPerc Total = Energy1.Total RefTotal = Energy1.RefTotal YearlyConsumption = Energy1.YearlyConsumption RefYearlyConsumption = Energy1.RefYearlyConsumption PotentialSavings = Energy1.PotentialSavings

		PotentialImprovementProd = <i>Energy1.PotentialImprovementProd</i> Comment = <i>Energy1.Comment</i> Type = <i>Energy1.Type</i> Active = <i>Energy1.Active</i> FactoryModel != <i>Energy1.FactoryModel</i> Periods.Length = <i>Energy1.Periods.Length</i> Periods(id) != <i>Energy1.Periods(id)</i>
Clonar entidad con datos incorrectos	<pre> EnergyCloneComponentView { Old = { Id = Guid.New, }, New = { Id = Guid.New, FactoryModelId = Guid.New } } EnergyCloneComponentView { Old = { Id = Guid.New, FactoryModelId = Guid.New } } EnergyCloneComponentView </pre>	Excepción: <i>BadRequestException</i>

Tabla 183. Diseño Pruebas Unitarias: Clonación de FM Energías

Producción

Vista de detalles y edición de FM Producción		
Prueba	Entrada	Resultado Esperado
Consultar FM Producción de un FactoryModel	<pre> FindProductionComponentByFactoryModelView { FactoryModelId: <i>FactoryModel1.Id</i> } </pre>	Id != no válido <i>Id = Production1.Id</i> Products.Length = <i>Production1.Products.Length</i> Products(id) = <i>Production1.Products(id)</i>
Consulta FM Producción de un FactoryModel que no existe	<pre> FindProductionComponentByFactoryModelView { FactoryModelId: <i>Guid.New</i> } </pre>	Id = no válido
Actualizar información mostrada	<pre> RefreshProductionComponentView { EnergyComponentDto: <i>Energy1</i> } </pre>	Id = válido Id = <i>Production1.Id</i>



		YearHours = <i>Production1.Production</i> TechnicalEfficDiff = <i>Production1.RefPower</i> UnitPerHourRatio = <i>Production1.PowerDiff</i> PotentialSavings = <i>Production1.PowerDiffPerc</i> PotentialImprovementProd = <i>Production1.Consumption</i> Comment != <i>Production1.Comment</i> Type = <i>Production1.Type</i> Active = <i>Production1.Active</i> FactoryModel = <i>Production1.FactoryModel</i> Products.Length = <i>Production1.Products.Length</i> Products (id) = <i>Production1.Products(id)</i> Products (name) != <i>Production1.Products(name)</i>
Actualizar información	UpdateProductionComponentView { <i>ProductionComponentDto: Energy1 {</i> <i>Comment = "es una prueba"</i> <i>Products[0].Name = "Porducto"</i> } }	Id = <i>válido</i> Id = <i>Production1.Id</i> YearHours = <i>Production1.Production</i> TechnicalEfficDiff = <i>Production1.RefPower</i> UnitPerHourRatio = <i>Production1.PowerDiff</i> PotentialSavings = <i>Production1.PowerDiffPerc</i> PotentialImprovementProd = <i>Production1.Consumption</i> Comment != <i>Production1.Comment</i> Type = <i>Production1.Type</i> Active = <i>Production1.Active</i> FactoryModel = <i>Production1.FactoryModel</i> Products.Length = <i>Production1.Products.Length</i> Products (id) = <i>Production1.Products(id)</i> Products (name) != <i>Production1.Products(name)</i>

Actualizar información de entidad no válida	<pre>UpdateProductionComponentView { ProductionComponentDto: new Dto(){ Id = Guid.New } } UpdateProductionComponentView { ProductionComponentDto: new Dto() } UpdateProductionComponentView</pre>	Exception: <i>ConflictRequestException</i>
--	---	--

Tabla 184. Diseño Pruebas Unitarias: Vista de detalles y edición de FM Producción

Creación de FM Producción		
Prueba	Entrada	Resultado Esperado
Crear FM Producción	<pre>ProductionCreateComponentView { FactoryModelId = FactoryModel1 Type = "Production" Active = true }</pre>	<pre>Id = válido YearHours = 0 TechnicalEfficDiff = 0 UnitPerHourRatio = 0 PotentialSavings = 0 PotentialImprovementProd = 0 Comment != válido Type = "Production" Active = true FactoryModel = FactoryModel Products.Length = 0</pre>
Crear FM Producción existiendo uno asignado al mismo FactoryModel	<pre>ProductionCreateComponentView { FactoryModelId = FactoryModel1 Type = "Production" Active = true }</pre>	<pre>Id = válido YearHours = 0 TechnicalEfficDiff = 0 UnitPerHourRatio = 0 PotentialSavings = 0 PotentialImprovementProd = 0 Comment != válido Type = "Production" Active = true FactoryModel = FactoryModel Products.Length = 0</pre>
Crear otro FM	<pre>ProductionCreateComponentView { FactoryModelId = FactoryModel1 Type = "aaa" Active = true } ProductionCreateComponentView</pre>	Excepción: <i>BadRequestException</i>

Tabla 185. Diseño Pruebas Unitarias: Creación de FM Producción

Clonación de FM Producción

Prueba	Entrada	Resultado Esperado
Clonar la entidad correctamente	<pre> ProductionCloneComponentView { Old = Production1, New = { Id = Guid.New, FactoryModelId = Guid.New } } </pre>	<pre> Id = válido Id != Production1.Id YearHours = Production1.Production TechnicalEfficDiff = Production1.RefPower UnitPerHourRatio = Production1.PowerDiff PotentialSavings = Production1.PowerDiffPerc PotentialImprovementProd = Production1.Consumption Comment = Production1.Comment Type = Production1.Type Active = Production1.Active FactoryModel != Production1.FactoryModel Products.Length = Production1.Products.Length Products (id) != Production1.Products(id) Products (name) != Production1.Products(name) </pre>
Clonar entidad con datos incorrectos	<pre> ProductionCloneComponentView { Old = { Id = Guid.New, }, New = { Id = Guid.New, FactoryModelId = Guid.New } } ProductionCloneComponentView { Old = { Id = Guid.New, FactoryModelId = Guid.New } } ProductionCloneComponentView </pre>	<pre> Excepción: BadRequestException </pre>

Tabla 186. Diseño Pruebas Unitarias: Clonación de FM Producción

Calidad

Vista de detalles y edición de FM Calidad		
Prueba	Entrada	Resultado Esperado



Consultar FM Calidad de un FactoryModel	FindQualityComponentByFactoryModelView { FactoryModelId: FactoryModel1.Id }	Id != no válido Id = Quality1.Id Products.Length = Quality1.Products.Length Products(id) = Quality1. Products (id)
Consulta FM Calidad de un FactoryModel que no existe	FindQualityComponentByFactoryModelView { FactoryModelId: Guid.New }	Id = no válido
Actualizar información mostrada	RefreshQualityComponentView { QualityComponentDto: Energy1 }	Id = válido Id = Quality1.Id PotentialSavings = Quality1.PowerDiffPerc PotentialImprovementProd = Quality1.Consumption Comment != Quality1.Comment Type = Quality1.Type Active = Quality1.Active FactoryModel = Quality1.FactoryModel Products.Length = Quality1.Products.Length Products (id) = Quality1.Products(id)
Actualizar información	UpdateQualityComponentView { QaulityComponentDto: Energy1 { Comment = "es una prueba" Products[0].Name = "Porducto" } }	Id = válido Id = Quality1.Id PotentialSavings = Quality1.PowerDiffPerc PotentialImprovementProd = Quality1.Consumption Comment != Quality1.Comment Type = Quality1.Type Active = Quality1.Active FactoryModel = Quality1.FactoryModel Products.Length = Quality1.Products.Length Products (id) = Quality1.Products(id) Products (name) != Quality1.Products(name)

Actualizar información de entidad no válida	<pre>UpdateQualityComponentView { QualityComponentDto: new Dto(){ Id = Guid.New } } UpdateProductionComponentView { QualityComponentDto: new Dto() } UpdateProductionComponentView</pre>	Exception: <i>ConflictRequestException</i>
--	--	--

Tabla 187. Diseño Pruebas Unitarias: Vista de detalles y edición de FM Calidad

Creación de FM Calidad		
Prueba	Entrada	Resultado Esperado
Crear FM Calidad	<pre>QualityCreateComponentView { FactoryModelId = FactoryModel1 Type = "Quality" Active = true }</pre>	<p>Id = válido PotentialSavings = 0 PotentialImprovementProd = 0 Comment != válido Type = "Quality" Active = true FactoryModel = FactoryModel Products.Length = 0</p>
Crear FM Calidad existiendo uno asignado al mismo FactoryModel	<pre>QualityCreateComponentView { FactoryModelId = FactoryModel1 Type = "Quality" Active = true }</pre>	<p>Id = válido YearHours = 0 TechnicalEfficDiff = 0 UnitPerHourRatio = 0 PotentialSavings = 0 PotentialImprovementProd = 0 Comment != válido Type = "Production" Active = true FactoryModel = FactoryModel Products.Length = 0</p>
Crear otro FM	<pre>QualityCreateComponentView { FactoryModelId = FactoryModel1 Type = "aaa" Active = true } EnergyCreateComponentView</pre>	Excepción: BadRequestException

Tabla 188. Diseño Pruebas Unitarias: Creación de FM Calidad

Clonación de FM Calidad		
Prueba	Entrada	Resultado Esperado

<p>Clonar la entidad correctamente</p>	<pre>QualityCloneComponentView { Old = Production1, New = { Id = Guid.New, FactoryModelId = Guid.New } }</pre>	<p>Id = válido Id != Quality1.Id PotentialSavings = Quality1.PowerDiffPerc PotentialImprovementProd = Quality1.Consumption Comment = Quality1.Comment Type = Quality1.Type Active = Quality1.Active FactoryModel != Quality1.FactoryModel Products.Length = Quality1.Products.Length Products (id) != Quality1.Products(id) Products (name) != Quality1.Products(name)</p>
<p>Clonar entidad con datos incorrectos</p>	<pre>QualityCloneComponentView { Old = { Id = Guid.New, }, New = { Id = Guid.New, FactoryModelId = Guid.New } } QualityCloneComponentView { Old = { Id = Guid.New, FactoryModelId = Guid.New } } ProductionCloneComponentView</pre>	<p>Excepción: BadRequestException</p>

Tabla 189. Diseño Pruebas Unitarias: Clonación de FM Calidad

Pruebas de Comportamiento

Caso de Uso: Inicio de sesión	
Prueba	Entrada
Iniciar sesión con datos correctos	Email: Javier.MartinezAlvarez@seresco.es Password: 123456
Prueba	Entrada
Iniciar sesión con datos incorrectos	Email: Javier.MartinezAlvarez@seresco.es Password: 12345
Prueba	Entrada
Iniciar sesión un usuario desactivado	Email: disable@seresco.es Password: 123456

Tabla 190. Diseño Pruebas de Comportamiento: Inicio de sesión

Caso de Uso: Dar de alta a usuarios	
Prueba	Entrada
Dar de alta a un usuario correctamente.	Email: prueba@email.com Password: 123456 Nombre: Usuario Apellidos: Prueba Comportamiento Roles: Administrador, Usuario, Observador
Prueba	Entrada
No introducir todos los campos necesarios.	Email: prueba@email.com Password: 123456 Nombre: Usuario Apellidos: Vacío Roles: Administrador, Usuario, Observador
Prueba	Entrada
El usuario cancela la operación.	N/A
Prueba	Entrada
El usuario ya existe en el sistema.	Email: prueba@email.com Password: 123456 Nombre: Usuario Apellidos: Prueba Comportamiento Roles: Administrador, Usuario, Observador

Tabla 191. Diseño Pruebas de Comportamiento: Dar de alta a usuarios

Caso de Uso: Consulta todos los usuarios
--



Prueba	Entrada
Consultar los usuarios del sistema	N/A

Tabla 192. Diseño Pruebas de Comportamiento: Consulta todos los usuarios

Caso de Uso: Modificar usuario	
Prueba	Entrada
Modificar un usuario	Usuario: prueba@email.com Nombre: Modificar Apellidos: Usuario
Prueba	Entrada
No introducir todos los campos necesarios.	Usuario: prueba@email.com Nombre: Modificar Apellidos: Vacío

Tabla 193. Diseño Pruebas de Comportamiento: Modificar usuario

Caso de Uso: Consulta detalles de un usuario	
Prueba	Entrada
Consultar detalles de un usuario	Usuario: prueba@email.com

Tabla 194. Diseño Pruebas de Comportamiento: Consulta detalles de un usuario

Caso de Uso: Consulta visiones	
Prueba	Entrada
Consultar listado de visiones	N/A

Tabla 195. Diseño Pruebas de Comportamiento: Consulta visiones

Caso de Uso: Añadir visión	
Prueba	Resultado Esperado
Dar de alta una visión.	Nombre: Visión prueba Departamento: CYS Sección: Nuevo software Área: PSA Localización: Oviedo Dirección: Calle Santa Susana
Prueba	Entrada

No introducir todos los campos necesarios.	Nombre: Vacío Departamento: CYS Sección: Nuevo software Área: PSA Localización: Oviedo Dirección: Calle Santa Susana
Prueba	Entrada
El usuario cancela la operación.	N/A

Tabla 196. Diseño Pruebas de Comportamiento: Añadir visión

Caso de Uso: Modificar visión	
Prueba	Entrada
Modificar una visión	Nombre: Modificación Visión Departamento: RRHH Sección: Nuevo personal
Prueba	Entrada
No introducir todos los campos necesarios.	Nombre: Modificación Visión Departamento: RRHH Sección: Vacío

Tabla 197. Diseño Pruebas de Comportamiento: Modificar visión

Caso de Uso: Clonar visión	
Prueba	Resultado Esperado
Clonar visión	Nuevo Nombre: Visión clonada
Prueba	Entrada
El usuario cancela la operación.	N/A

Tabla 198. Diseño Pruebas de Comportamiento: Clonar visión

Caso de Uso: Consulta máquinas	
Prueba	Entrada
Consultar listado de máquinas	Visión seleccionada: Visión clonada

Tabla 199. Diseño Pruebas de Comportamiento: Consulta máquinas

Caso de Uso: Añadir máquina



Prueba	Entrada
Dar de alta una máquina.	Nombre: Máquina creada Descripción: Producción software ASPA Versión: 1.0 Grupo: ArcelorMittal
Prueba	Entrada
No introducir todos los campos necesarios.	Nombre: Máquina creada Descripción: Producción software ASPA Versión: Vacío Grupo: ArcelorMittal
Prueba	Entrada
El usuario cancela la operación.	N/A

Tabla 200. Diseño Pruebas de Comportamiento: Añadir máquina

Caso de Uso: Modificar máquina	
Prueba	Entrada
Modificar una máquina	Nombre: Máquina modificada Versión: 1.1
Prueba	Entrada
No introducir todos los campos necesarios.	Descripción: Vacío

Tabla 201. Diseño Pruebas de Comportamiento: Modificar máquina

Caso de Uso: Clonar máquina	
Prueba	Entrada
Clonar máquina	Nuevo Nombre: Máquina clonada
Prueba	Entrada
El usuario cancela la operación.	N/A

Tabla 202. Diseño Pruebas de Comportamiento: Clonar máquina

Caso de Uso: Consulta FactoryModel	
Prueba	Entrada
Consultar listado de FactoryModels	Visión seleccionada: Visión clonada Máquina seleccionada: Máquina clonada

Tabla 203. Diseño Pruebas de Comportamiento: Consulta FactoryModels

Caso de Uso: Consulta componentes	
Prueba	Entrada
Consultar listado de componentes dado un FactoryModels	Visión seleccionada: Visión clonada Máquina seleccionada: Máquina clonada FactoryModel seleccionada: FactoryModel clonado

Tabla 204. Diseño Pruebas de Comportamiento: Consulta componentes

Caso de Uso: Añadir FactoryModels	
Prueba	Entrada
Dar de alta un FactoryModel.	Nombre: FactoryModel creado Descripción: Proyecto ASPA-IR Versión: 1.0
Prueba	Entrada
No introducir todos los campos necesarios.	Nombre: FactoryModel creado Descripción: Proyecto ASPA-IR Versión: <i>Vacío</i>
Prueba	Entrada
El usuario cancela la operación.	N/A

Tabla 205. Diseño Pruebas de Comportamiento: Añadir FactoryModel

Caso de Uso: Modificar FactoryModel	
Prueba	Entrada
Modificar una FactoryModel	Nombre: FactoryModel modificado Descripción: Proyecto ASPA-CVVS Versión: 1.1
Prueba	Entrada
No introducir todos los campos necesarios.	Nombre: FactoryModel modificado Descripción: <i>Vacío</i> Versión: 1.1

Tabla 206. Diseño Pruebas de Comportamiento: Modificar FactoryModel

Caso de Uso: Clonar FactoryModel	
Prueba	Entrada
Clonar FactoryModel	Nuevo Nombre: FactoryModel clonado
Prueba	Entrada



El usuario cancela la operación.	N/A
----------------------------------	-----

Tabla 207. Diseño Pruebas de Comportamiento: Clonar FactoryModel

Caso de Uso: Añadir costes	
Prueba	Entrada
Dar de alta un nuevo coste	<p>Jornadas (horas): 8.76 Coste Jornadas por Trabajador: 1752 Número de trabajadores: 400 Número totales anual: 100</p> <p><i>Variables</i></p> <p>Naturaleza: Embalaje Familia: Cartón Unidades: 10.5 Precio/Unidad: 2.5 Ref Precio/Unidad: 2.2</p> <p><i>Fijos</i></p> <p>Naturaleza: Alquiler Familia: Electricidad Total: 600 INFL: 25 SOST: 22</p>

Tabla 208. Diseño Pruebas de Comportamiento: Añadir costes

Caso de Uso: Consulta de costes	
Prueba	Entrada
Consultar listado de componentes Costes dado un FactoryModel	<p>Visión seleccionada: Visión clonada Máquina seleccionada: Máquina clonada FactoryModel seleccionada: FactoryModel clonado</p>

Tabla 209. Diseño Pruebas de Comportamiento: Consulta de costes

Caso de Uso: Añadir potencia	
Prueba	Entrada
Modificar valores de potencia añadiendo nuevos datos	<p><i>P1</i> KW: 70 Tarifa: 39.14 Ref kW: 20 Ref Tarifa: 39.14</p>

	<p>P2 KW: 20 Tarifa: 19.58 Ref kW: 35 Ref Tarifa: 19.59</p> <p>P3 KW: 50 Tarifa: 14.33 Ref kW: 60 Ref Tarifa: 14.33</p>
--	---

Tabla 210. Diseño Pruebas de Comportamiento: Añadir potencia

Caso de Uso: Añadir consumo	
Prueba	Entrada
<p>Modificar valores de consumo añadiendo nuevos datos</p>	<p><i>Consumo anual</i> kWh: 15483,2 Reducción: 0.1</p> <p><i>Discriminación horaria</i> P1 Invierno: 23 Verano: 35 Ref Invierno: 22 Ref Verano: 28 Precio kWh: 0.11 Ref Precio kWh: 0.11</p> <p>P2 Invierno: 66 Verano: 55 Ref Invierno: 70 Ref Verano: 60 Precio kWh: 0.09 Ref Precio kWh: 0.09</p> <p>P3 Invierno: 4 Verano: 10 Ref Invierno: 8 Ref Verano: 12 Precio kWh: 0.08 Ref Precio kWh: 0.08</p>

Tabla 211. Diseño Pruebas de Comportamiento: Añadir consumo

Caso de Uso: Consulta de energía	
Prueba	Entrada
Consultar listado de componentes Energía dado un FactoryModel	Visión seleccionada: Visión clonada Máquina seleccionada: Máquina clonada FactoryModel seleccionada: FactoryModel clonado

Tabla 212. Diseño Pruebas de Comportamiento: Consulta de energía

Caso de Uso: Añadir productos	
Prueba	Entrada
Dar de alta un nuevo producto	Variación eficiencia: 0.1 Variación eficiencia real: 0.11 Variación ratio: 0.15 <i>Productos</i> Producto: YA147896 Teórica: 2200 Técnica: 1750 Real: 1050 Horas totales: 2200 Coste Unidad: 50 Ref Horas totales: 51

Tabla 213. Diseño Pruebas de Comportamiento: Añadir productos

Caso de Uso: Consulta de producción	
Prueba	Entrada
Consultar listado de componentes Producción dado un FactoryModel	Visión seleccionada: Visión clonada Máquina seleccionada: Máquina clonada FactoryModel seleccionada: FactoryModel clonado

Tabla 214. Diseño Pruebas de Comportamiento: Consulta de producción

Caso de Uso: Añadir defectos	
Prueba	Entrada
Dar de alta un nuevo defecto	Producto seleccionado: YA147896 <i>Defectos</i> Descripción: Fallo de diseño

	Defectos: 10 Coste MP rec: 0 Ref Defectos: 5 Coste MP rec: 0
--	---

Tabla 215. Diseño Pruebas de Comportamiento: Añadir defecto

Caso de Uso: Consulta de calidad	
Prueba	Entrada
Consultar listado de componentes Calidad dado un FactoryModel	Visión seleccionada: Visión clonada Máquina seleccionada: Máquina clonada FactoryModel seleccionada: FactoryModel clonado

Tabla 216. Diseño Pruebas de Comportamiento: Consulta de calidad

Pruebas de Usabilidad y Accesibilidad

A continuación, se detallará el cuestionario que ha permitido evaluar si la aplicación es usable y accesible para un usuario sin conocimientos en la aplicación. Este cuestionario será respondido por el propio usuario de la prueba, a excepción, del último apartado:

Preguntas de carácter general

Preguntas para conocer el nivel de conocimiento que puede disponer el usuario.

¿Usa un ordenador frecuentemente? <ol style="list-style-type: none"> 1. Todos los días 2. Varias veces a la semana 3. Ocasionalmente 4. Nunca o casi nunca
¿Qué tipo de actividades realiza con el ordenador? <ol style="list-style-type: none"> 1. Es parte de mi trabajo o profesión 2. Lo uso básicamente para ocio 3. Solo empleo aplicaciones estilo Office 4. Únicamente leo el correo y navego ocasionalmente
¿Ha usado alguna vez software como el de esta prueba? <ol style="list-style-type: none"> 1. Sí, he empleado software similar 2. No, aunque si empleo otros programas que me ayudan a realizar tareas similares 3. No, nunca
¿Qué busca Vd. Principalmente en un programa? <ol style="list-style-type: none"> 1. Que sea fácil de usar 2. Que sea intuitivo 3. Que sea rápido



4. Que tenga todas las funciones necesarias

Actividades guiadas

Acciones guiadas por el responsable de la prueba cuyo objetivo es conocer si los usuarios saben realizar las acciones. Se evaluará si se ha completado, dificultades, adaptación que ha tenido haciendo esa prueba (dificultad) y tiempo empleado.

1. Añadir un usuario a la aplicación
Completado:
Dificultades:
Adaptación:
Tiempo:

2. Añadir visión
Completado:
Dificultades:
Adaptación:
Tiempo:

3. Clonar visión
Completado:
Dificultades:
Adaptación:
Tiempo:
Observaciones:

4. Clonar FactoryModel
Completado:
Dificultades:



<p>Adaptación:</p> <p>Tiempo:</p>
5. Modificar datos en FactoryModel: Energía
<p>Completado:</p> <p>Dificultades:</p> <p>Adaptación:</p> <p>Tiempo:</p>

Preguntas Cortas sobre la Aplicación y Observaciones

Preguntas efectuadas después de la prueba con el sistema con tal de conocer las sensaciones que ha tenido haciendo la prueba.

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Sabe dónde está dentro de la aplicación?</i>				
<i>¿Existe ayuda para las funciones en caso de que tenga dudas?</i>				
<i>¿Le resulta sencillo el uso de la aplicación?</i>				
<i>¿Le ha costado adaptarse al funcionamiento?</i>				
<i>¿Conoce la funcionalidad de la aplicación?</i>				
<i>¿Cree que la aplicación tiene un aspecto similar a una hoja de cálculos?</i>				
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Funciona cada tarea como Vd. Espera?</i>				
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>				
<i>¿Tiene la sensación de no conocer el estado de las acciones que realiza?</i>				
<i>¿Considera que los cálculos hechos son correctos?</i>				
<i>¿Considera que le puede aportar algo si realiza o realizase una actividad económica?</i>				
<i>¿Considera que está satisfecho con el funcionamiento de la app?</i>				
Calidad del 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>				
<i>La aplicación está adaptada para un entorno empresarial y comercial</i>				
Diseño de la Interfaz		Si	No	A veces
<i>¿Le resulta fácil de usar?</i>				
<i>¿El diseño de las pantallas es claro y atractivo?</i>				
<i>¿Cree que el programa está bien estructurado?</i>				
<i>¿Cree que el programa es usable?</i>				
<i>¿Modificaría alguna de las páginas?</i>				
Observaciones				

Cuestionario para el responsable de las Pruebas

Último apartado, donde el responsable de pruebas escribirá todos los detalles que se han detectado en la prueba. Se cumplimentará a la finalización de este.

Aspecto Observado	Notas
<i>El usuario comienza a trabajar de forma rápida por las tareas</i>	
<i>Tiempo en realizar cada tarea</i>	
<i>Errores leves cometidos</i>	
<i>Errores graves cometidos</i>	



Capítulo 5 CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN

FASE DE DESARROLLO

CSI



CSI 1: PREPARACIÓN DEL ENTORNO DE GENERACIÓN Y CONSTRUCCIÓN

En la siguiente sección se van a definir una serie de aspectos que se deberán tener en cuenta en el momento de la construcción del sistema.

Estándares y normas seguidos

Primeramente, se va a comentar las normas que se han seguido [15]:

- **ISO 12207:** Modelo de Ciclos de Vida del Software. Esta norma internacional deberá ser seguida para lograr tener un marco común a los procesos software.
- **ISO 9126:** Norma internacional que sirve para evaluar si los procesos software han seguido una linealidad en el proceso de desarrollo. Viene definida dentro de los marcos de Calidad del proceso, Calidad del Producto Software y Calidad en Uso.
- Cualquier otra norma que puede aportar de forma indirecta una ventaja para el proyecto.

Si se habla sobre estándares seguidos para la construcción del proyecto:

- **HTML5:** Estándar que define la última versión del lenguaje HTML. Incluye diversas mejoras permitiendo una mejor interoperabilidad entre diferentes navegadores. Se encuentra estandarizado por medio del W3C.
- **CSS:** Hojas de estilo reguladas por el W3C.
- **Convenciones .NET en *namespaces*, clases y otros aspectos de programación** [16]: Recomendaciones por parte de Microsoft para mantener un estilo de código único y fácilmente reconocible por el resto de los desarrolladores.

Para poder cumplir de una forma óptima las anteriores normas y estándares, se han hecho labores de investigación para adquirir una mejor visión.

Como medida complementaria, se han decidido seguir tutoriales relativos a las tecnologías usadas en el proyecto. En el caso de Angular se ha seguido el siguiente [17], y en el caso de .NET se ha seguido las indicaciones dadas en el puesto de trabajo y en [18].

Lenguajes de programación

A continuación, se describirán los lenguajes de programación usados para realizar el desarrollo de la aplicación.

- **C#:** Lenguaje de programación multiparadigma usado en entorno de desarrollo de .NET. Se considera la evolución de *VisualBasic* con una sintaxis similar a C/C++.
- **Javascript:** Lenguaje usado en el desarrollo web que permite crear *scripts* interpretados por el navegador. El uso que se hace en este proyecto se puede considerar residual.



- **Typescript:** Lenguaje usado por el cliente de la aplicación. Es un tipo de lenguaje creado por Microsoft que permite hacer uso de tipos sobre código *Javascript*. En una última instancia, se compilará a código JS.
- **CSS:** Lenguaje de diseño gráfico que permite definir y crear presentaciones para la web. En la aplicación se hace uso también de SASS, que se corresponde con un lenguaje de diseño gráfico más potente.
- **Archivos por lotes:** Lenguaje de programación usado en sistema operativo Windows para poder generar scripts con un cometido. Su uso se limita a despliegue.
- **Shell bash:** Lenguaje de programación usado en sistema operativo Linux para poder generar scripts con un cometido. Su uso se limita a despliegue.

Finalmente se enumerarán una serie de medidas tenidas en cuenta en la aplicación para su correcto funcionamiento.

	SI	NO	N/A
Se hace validación en el cliente y en el servidor	X		
Se han usado validadores nativos de un framework	X		
Se han validado las entradas JSON y XML con esquemas adecuados	X		
Se hacen conversiones de tipo adecuadas con captura estricta de excepciones a las entradas leídas como string	X		
Se comprueba el rango de los valores de entrada numéricos			X
Se comprueba la longitud mínima y máxima para strings	X		
Se comprueba el conjunto de valores admitidos para entradas que solo puedan poseer un nº finito de valores de forma estricta			X
Se usan expresiones regulares para validar cualquier otro tipo de dato estructurado que no encaje en las categorías anteriores	X		
Si hay entradas de texto libre Unicode, se normalizan a una codificación común	X		
Se usan librerías de codificación de entradas anti XSS	X		
Solo se permiten insertar datos en lugares considerados como seguros, usando codificación de datos adecuada a cada lugar	X		
Si se admiten entradas en formato HTML, se usa una librería para "limpiar" su contenido			X
Se usa el flag HTTPOnly		X	
Se implementa CSP	X		
Se usa la cabecera X-XSS-Protection	X		
Se usan frameworks de desarrollo modernos y actualizados	X		

Se toman medidas para validar los archivos que el usuario pueda subir			X
Se toman medidas para validar las direcciones de e-mail que vayan a usarse	X		

Tabla 217. Medidas de seguridad tomadas en el código de la aplicación

Herramientas y programas usados para el desarrollo

Para el correcto desarrollo del software, se ha hecho uso de las siguientes herramientas:

Visual Studio 2019 Enterprise



Fig 66. Logo de VisuaStudio

Entorno de trabajo disponible para los principales sistemas operativos compatible con .NET Core. Tiene un soporte muy bueno con lenguajes con C# o Typescript junto a todas las capacidades que dispone de tratamiento sobre Microsoft Azure. Nos permite crear nuevos proyectos, depurarlos y desplegarlos sobre el entorno *Kubernetes* donde se ejecuta.

.NET C#: Entorno de trabajo muy usado que nos permite construir un sistema software de una forma sencilla. Está basada en una arquitectura modular donde se usan los paquetes estrictamente necesarios. Es el *framework* usado en los principales módulos de la aplicación. A pesar de su largo recorrido en el tiempo, los microservicios han sido la parte más difícil del trabajo. La información es residual y hay que lograr encontrar una solución a base de prueba y error.

Visual Studio Code

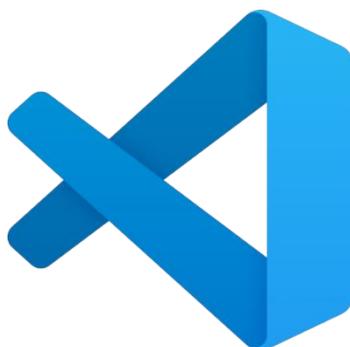


Fig 67. Logo: Visual Studio Code

Editor de código fuente desarrollado por Microsoft muy usado en la actualidad. Dispone de soporte para diferentes lenguajes de programación basándose en *plugins* que ampliar las funcionalidades.

Por defecto, dispone de soporte de depuración para código JS, integración con GIT o *IntelliSense* (resaltado de sintaxis en base al lenguaje).

El uso que se ha hecho de él es, principalmente, para el desarrollo del cliente de Angular. Una pequeña parte, se ha usado, para la generación de los *scripts* de despliegue.

Microsoft SQL Server Management Studio



Fig 68. Logo: Microsoft SQL Server Management Studio

Entorno de trabajo usado para hacer pruebas contra un servidor de base de datos de Microsoft SQL Server. Se corresponde con la solución oficial, con más aceptación de uso y que nos permite configurar *Profiles*, *Backups*...

Dotnet/Nuget

Herramienta usada para para gestión de proyectos basado en .NET Core. Permite la ejecución, compilación, gestión de dependencias o nuevas opciones aparecidas a partir de las dependencias instaladas. No tiene ningún tipo de configuración y es multiplataforma.

Docker

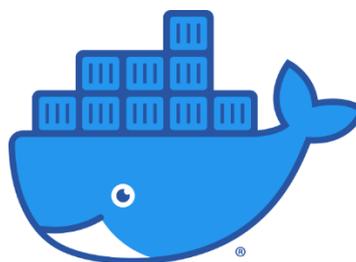


Fig 69. Logo: Docker

Herramienta de código abierto que nos permite automatizar despliegues de aplicaciones haciendo uso de contenedores. Un contenedor es una máquina virtual reducida que se ejecuta de forma independiente al sistema cuyo objetivo es tener un entorno cerrado y preconfigurado.

Kubernetes (KubeCtl)



Fig 70. Logo: Kubernetes

Herramienta usada para la automatización de despliegues de aplicaciones haciendo uso de contenedores. *Kubernetes* ejecuta los contenedores Docker permitiendo realizar la orquestación de forma sencilla. *Kubectl* se corresponde con la herramienta que nos permite hacer una gestión sobre el servidor Kubernetes.

Azure CLI

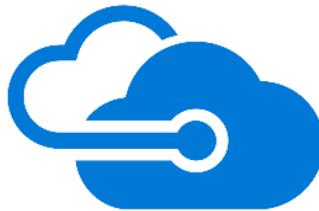


Fig 71. Logo: Azure CLI

Herramienta creada por Microsoft usada para el manejo y control de los sistemas de MS Azure. Permite realizar gestiones sobre una suscripción para añadir, quitar o modificar los servicios contratados. Es usado para manejo de recursos de los servicios.

Git



Fig 72. Logo: GIT



Herramienta que nos permite disponer de un control de versión de los cambios efectuado en el proyecto. El sistema se basa en la construcción de un repositorio local (y normalmente en la nube) que nos permite modificar cambios, visualizar los cambios, crear ramas, crear parches... En una primera instancia se ha desarrollado haciendo uso de un repositorio en la nube en GitHub, pero la intención es hacer una migración a Azure DevOps en un futuro.



CSI 2: GENERACIÓN DEL CÓDIGO DE LOS COMPONENTES Y PROCEDIMIENTOS

Dependencias entre módulos

Se han descompuesto los siguientes módulos una vez construido el sistema de información requerido:

- **Common:** Almacena librerías comunes que pueden compartir los paquetes para reutilizar código.
 - **WebApi:** Almacena interfaces, clases abstractas y clases extensoras usadas para todo lo relativo a ASP Core.
 - **Authentication:** Almacena interfaces, clases abstractas y clases extensoras relativas a la autenticación JWT necesaria para usar las API.
- **Client:** Paquete que contiene el módulo del cliente. Contiene un paquete ASP SSR que ejecuta un paquete Angular 9.
- **Services:** Espacio de nombres que almacena todos los servicios relativos a capa de Back-end.
 - **IdentityService:** Contiene todas las clases necesarias para la gestión de inicio de sesión, usuarios y roles de un usuario.
 - **VisionService:** Contiene todas las clases necesarias para la gestión de visiones.
 - **MachineService:** Contiene todas las clases necesarias para la gestión de máquinas y sus agrupaciones.
 - **FactoryModelService:** Contiene todas las clases necesarias para la gestión de FactoryModel y sus componentes.
 - **Components:** Espacio de nombres que contiene todos los FM Componentes.
 - **CostService:** Contiene todas las clases necesarias para la gestión del componente de costes.
 - **EnergyService:** Contiene todas las clases necesarias para la gestión del componente de energía.
 - **ProductionService:** Contiene todas las clases necesarias para la gestión del componente de producción.
 - **QualityService:** Contiene todas las clases necesarias para la gestión del componente de calidad.

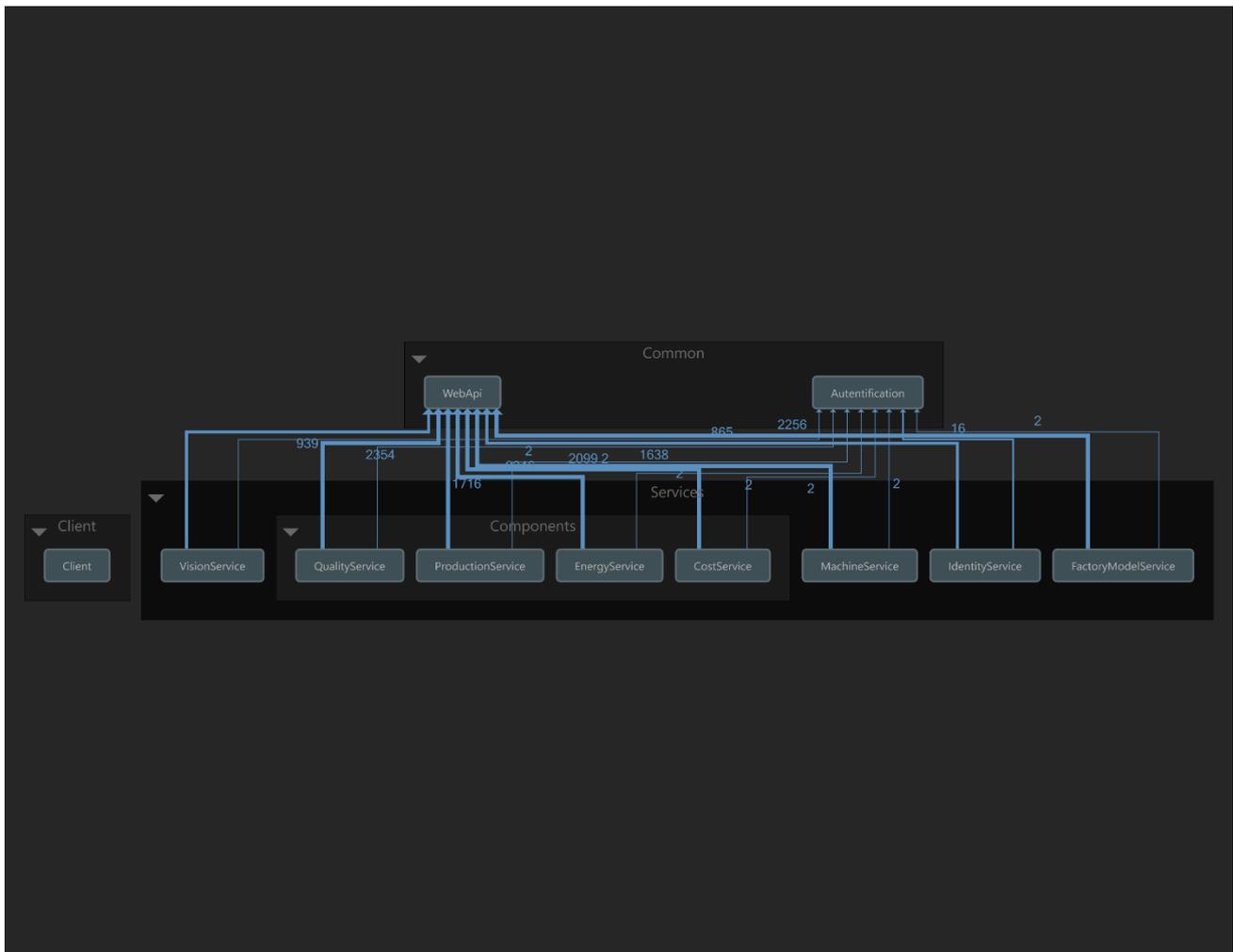


Fig 73. Diagrama de dependencias entre los diferentes módulos

CSI 3: EJECUCIÓN DE LAS PRUEBAS UNITARIAS

El resultado que se ha obtenido de la ejecución de 157 pruebas unitarias han sido todas positivas. Dentro de estas pruebas podemos distinguir:

- Listado de pruebas descritas en la fase de Diseño con sus correspondientes casos de prueba.
- Pruebas añadidas relacionadas con los controladores HTTP.
- Pruebas adicionales relacionadas con las clases e interfaces definidas en el módulo común al resto de subsistemas.

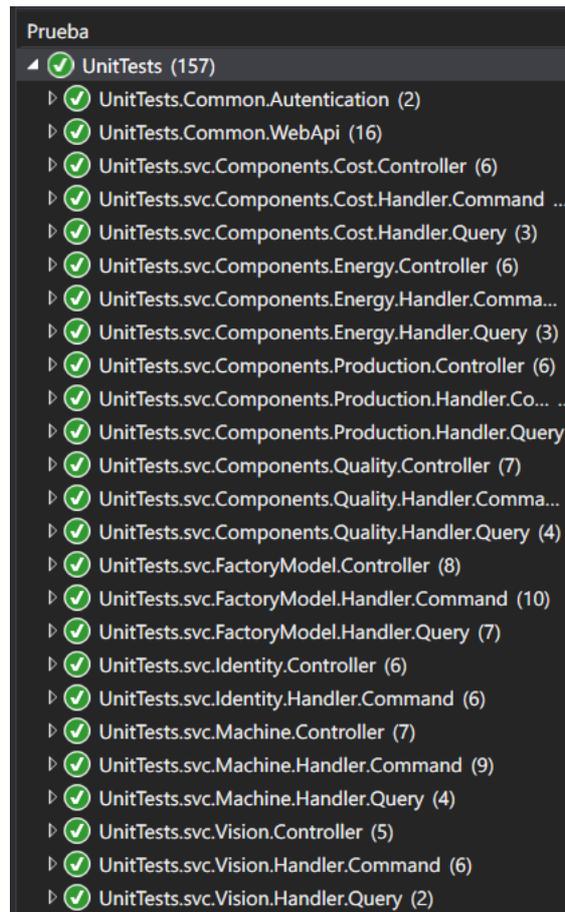


Fig 74. Resultados de las pruebas unitarias ejecutadas

El resultado de las pruebas se puede consultar a través del entorno de integración continua en MS Azure Pipelines. Los resultados obtenidos son los siguientes:

- **Line Coverage:** 83,6 %
- **Branch Coverage:** 78,8%

CSI 4: EJECUCIÓN DE LAS PRUEBAS DE INTEGRACIÓN

Caso de Uso: Inicio de sesión	
Prueba	Resultado Esperado
Iniciar sesión con datos correctos	Inicia sesión correctamente. Redirección a <i>layout</i> privado.
	Resultado Obtenido
	OK.
Prueba	Resultado Esperado
Iniciar sesión con datos incorrectos	Mensaje de error de usuario o contraseña incorrecto.
	Resultado Obtenido
	OK.
Prueba	Resultado Esperado
Iniciar sesión un usuario desactivado	Mensaje de error de usuario desactivado.
	Resultado Obtenido
	OK.

Tabla 218. Resultado Pruebas de Comportamiento: Inicio de sesión

Caso de Uso: Dar de alta a usuarios	
Prueba	Resultado Esperado
Dar de alta a un usuario correctamente.	<ul style="list-style-type: none"> – Mensaje de que se creado correctamente el usuario. – Nuevo usuario en el sistema.
	Resultado Obtenido
	OK.
Prueba	Resultado Esperado
No introducir todos los campos necesarios.	Mensaje de error de campo requerido.
	Resultado Obtenido
	OK.
Prueba	Resultado Esperado
El usuario cancela la operación.	Regreso a listado de usuarios.
	Resultado Obtenido
	OK.
Prueba	Resultado Esperado
El usuario crea un usuario que ya existe.	Mensaje de error de usuario ya existente.
	Resultado Obtenido
	OK.

Tabla 219. Resultado Pruebas de Comportamiento: Dar de alta a usuarios

Caso de Uso: Consulta todos los usuarios	
Prueba	Resultado Esperado
Consultar los usuarios del sistema	Listado de los usuarios del sistema.
	Resultado Obtenido
	OK.

Tabla 220. Resultado Pruebas de Comportamiento: Consulta todos los usuarios

Caso de Uso: Modificar usuario	
Prueba	Resultado Esperado
Modificar un usuario	<ul style="list-style-type: none"> – Mensaje de que se modificar correctamente el usuario. – Modificado usuario en el sistema.
	Resultado Obtenido
	OK.
Prueba	Resultado Esperado
No introducir todos los campos necesarios.	Mensaje de error de campo requerido.
	Resultado Obtenido
	OK.

Tabla 221. Resultado Pruebas de Comportamiento: Modificar usuario

Caso de Uso: Consulta detalles de un usuario	
Prueba	Resultado Esperado
Consultar detalles de un usuario	Página perfil del usuario que ha iniciado sesión.
	Resultado Obtenido
	OK.

Tabla 222. Resultado Pruebas de Comportamiento: Consulta detalles de un usuario

Caso de Uso: Consulta visiones	
Prueba	Resultado Esperado
Consultar listado de visiones	Listado de las visiones del sistema.
	Resultado Obtenido
	OK.

Tabla 223. Resultado Pruebas de Comportamiento: Consulta visiones

Caso de Uso: Añadir visión	
Prueba	Resultado Esperado
Dar de alta una visión.	<ul style="list-style-type: none"> – Mensaje de que se creado correctamente la visión. – Nueva visión en el sistema.

	Resultado Obtenido
	OK.
Prueba	Resultado Esperado
No introducir todos los campos necesarios.	Mensaje de error de campo requerido.
	Resultado Obtenido
	OK.
Prueba	Resultado Esperado
El usuario cancela la operación.	Regreso a listado de visiones.
	Resultado Obtenido
	OK.

Tabla 224. Resultado Pruebas de Comportamiento: Añadir visión

Caso de Uso: Modificar visión	
Prueba	Resultado Esperado
Modificar una visión	<ul style="list-style-type: none"> – Mensaje de que se modificar correctamente la visión. – Modificada visión en el sistema.
	Resultado Obtenido
	OK.
Prueba	Resultado Esperado
No introducir todos los campos necesarios.	Mensaje de error de campo requerido.
	Resultado Obtenido
	OK.

Tabla 225. Resultado Pruebas de Comportamiento: Modificar visión

Caso de Uso: Clonar visión	
Prueba	Resultado Esperado
Clonar visión	<ul style="list-style-type: none"> – Mensaje de confirmación que se ha realizado correctamente. – Nueva visión clonada del sistema.
	Resultado Obtenido
	OK.
Prueba	Resultado Esperado
El usuario cancela la operación.	Regreso a listado de visiones.
	Resultado Obtenido
	OK.

Tabla 226. Resultado Pruebas de Comportamiento: Clonar visión

Caso de Uso: Consulta máquinas	
Prueba	Resultado Esperado
Consultar listado de máquinas	Listado de las máquinas del sistema.
	Resultado Obtenido
	OK.

Tabla 227. Resultado Pruebas de Comportamiento: Consulta máquinas

Caso de Uso: Añadir máquina	
Prueba	Resultado Esperado
Dar de alta una máquina.	<ul style="list-style-type: none"> – Mensaje de que se creado correctamente la máquina. – Nueva máquina en el sistema.
	Resultado Obtenido
	OK.
Prueba	Resultado Esperado
No introducir todos los campos necesarios.	Mensaje de error de campo requerido.
	Resultado Obtenido
	OK.
Prueba	Resultado Esperado
El usuario cancela la operación.	Regreso a listado de visiones.
	Resultado Obtenido
	OK.

Tabla 228. Resultado Pruebas de Comportamiento: Añadir máquina

Caso de Uso: Modificar máquina	
Prueba	Resultado Esperado
Modificar una máquina	<ul style="list-style-type: none"> – Mensaje de que se modificar correctamente la máquina. – Modificada máquina en el sistema.
	Resultado Obtenido
	OK.
Prueba	Resultado Esperado
No introducir todos los campos necesarios.	Mensaje de error de campo requerido.
	Resultado Obtenido
	OK.

Tabla 229. Resultado Pruebas de Comportamiento: Modificar máquina

Caso de Uso: Clonar máquina	
Prueba	Resultado Esperado

Clonar máquina	<ul style="list-style-type: none"> - Mensaje de confirmación que se ha realizado correctamente. - Nueva máquina clonada del sistema.
	Resultado Obtenido OK.
Prueba	Resultado Esperado
El usuario cancela la operación.	Regreso a listado de máquinas.
	Resultado Obtenido OK.

Tabla 230. Resultado Pruebas de Comportamiento: Clonar máquina

Caso de Uso: Consulta FactoryModel	
Prueba	Resultado Esperado
Consultar listado de FactoryModels	Listado de las FactoryModels del sistema.
	Resultado Obtenido OK.

Tabla 231. Resultado Pruebas de Comportamiento: Consulta FactoryModels

Caso de Uso: Consulta componentes	
Prueba	Resultado Esperado
Consultar listado de componentes dado un FactoryModels	Listado de los componentes del FactoryModels.
	Resultado Obtenido OK.

Tabla 232. Resultado Pruebas de Comportamiento: Consulta componentes

Caso de Uso: Añadir FactoryModels	
Prueba	Resultado Esperado
Dar de alta un FactoryModel.	<ul style="list-style-type: none"> - Mensaje de que se creado correctamente el FactoryModel. - Nuevo FactoryModel en el sistema.
	Resultado Obtenido OK.
Prueba	Resultado Esperado
No introducir todos los campos necesarios.	Mensaje de error de campo requerido.
	Resultado Obtenido OK.
Prueba	Resultado Esperado

El usuario cancela la operación.	Regreso a listado de visiones.
	Resultado Obtenido
	OK.

Tabla 233. Resultado Pruebas de Comportamiento: Añadir FactoryModel

Caso de Uso: Modificar FactoryModel	
Prueba	Resultado Esperado
Modificar una FactoryModel	<ul style="list-style-type: none"> – Mensaje de que se modificar correctamente un FactoryModel. – Modificado FactoryModel en el sistema.
	Resultado Obtenido
	OK.
Prueba	Resultado Esperado
No introducir todos los campos necesarios.	Mensaje de error de campo requerido.
	Resultado Obtenido
	OK.

Tabla 234. Resultado Pruebas de Comportamiento: Modificar FactoryModel

Caso de Uso: Clonar FactoryModel	
Prueba	Resultado Esperado
Clonar FactoryModel	<ul style="list-style-type: none"> – Mensaje de confirmación que se ha realizado correctamente. – Nuevo FactoryModel clonado del sistema.
	Resultado Obtenido
	OK.
Prueba	Resultado Esperado
El usuario cancela la operación.	Regreso a listado de FactoryModel.
	Resultado Obtenido
	OK.

Tabla 235. Resultado Pruebas de Comportamiento: Clonar FactoryModel

Caso de Uso: Añadir costes	
Prueba	Resultado Esperado
Dar de alta un nuevo coste	Actualización de los cálculos del sistema.
	Resultado Obtenido
	OK.

Tabla 236. Resultado Pruebas de Comportamiento: Añadir costes



Caso de Uso: Consulta de costes	
Prueba	Resultado Esperado
Consultar listado de componentes Costes dado un FactoryModel	Cálculos relativos a ese componente de costes.
	Resultado Obtenido
	OK.

Tabla 237. Resultado Pruebas de Comportamiento: Consulta de costes

Caso de Uso: Añadir potencia	
Prueba	Resultado Esperado
Modificar valores de potencia añadiendo nuevos datos	Actualización de los cálculos del sistema.
	Resultado Obtenido
	OK.

Tabla 238. Resultado Pruebas de Comportamiento: Añadir potencia

Caso de Uso: Añadir consumo	
Prueba	Resultado Esperado
Modificar valores de consumo añadiendo nuevos datos	Actualización de los cálculos del sistema.
	Resultado Obtenido
	OK.

Tabla 239. Resultado Pruebas de Comportamiento: Añadir consumo

Caso de Uso: Consulta de energía	
Prueba	Resultado Esperado
Consultar listado de componentes Energía dado un FactoryModel	Cálculos relativos a ese componente de Energía.
	Resultado Obtenido
	OK.

Tabla 240. Resultado Pruebas de Comportamiento: Consulta de energía

Caso de Uso: Añadir productos	
Prueba	Resultado Esperado
Dar de alta un nuevo producto	Actualización de los cálculos del sistema.
	Resultado Obtenido
	OK.

Tabla 241. Resultado Pruebas de Comportamiento: Añadir productos

Caso de Uso: Consulta de producción



Prueba	Resultado Esperado
Consultar listado de componentes Producción dado un FactoryModel	Cálculos relativos a ese componente de producción.
	Resultado Obtenido
	OK.

Tabla 242. Resultado Pruebas de Comportamiento: Consulta de producción

Caso de Uso: Añadir defectos	
Prueba	Resultado Esperado
	Actualización de los cálculos del sistema.
Dar de alta un nuevo defecto	Resultado Obtenido
	OK.

Tabla 243. Resultado Pruebas de Comportamiento: Añadir defecto

Caso de Uso: Consulta de calidad	
Prueba	Resultado Esperado
Consultar listado de componentes Calidad dado un FactoryModel	Cálculos relativos a ese componente de calidad.
	Resultado Obtenido
	OK.

Tabla 244. Resultado Pruebas de Comportamiento: Consulta de calidad



CSI 5: EJECUCIÓN DE LAS PRUEBAS DEL SISTEMA

Pruebas de carga y rendimiento

Para realizar las siguientes pruebas se ha hecho uso de una máquina de **Azure Kubernetes Service** de tipo *Standard_F4S_V2* con un nodo. Se corresponde con una de las versiones más económicas que nos permite contratar Azure pero que debería de ser suficiente para la carga que pueda tener la aplicación. Dispone de las siguientes características:

- Intel Xeon Platinum 8168
- 8 GiB de Ram
- 4 vCPU

Al igual que las pruebas unitarias ejecutadas, se ha decidido montar un entorno de integración continua para poder ejecutar este tipo de pruebas.

Para ello, se ha usado la herramienta que proporciona Microsoft llamada **Azure Performance Test**. Se ha ejecutado las pruebas y hemos obtenido los siguientes resultados:

Primera prueba

- **Número de usuarios concurrentes:** 20
- **Tiempo de la prueba:** 10 minutos
- **Número de peticiones ejecutadas:** 48.874 peticiones
- **Número de peticiones completadas:** 29.722 peticiones
- **Número de peticiones falladas:** 19.152 peticiones
- **Porcentaje de peticiones completadas:** 60,81%
- **Tiempo medio de respuesta:** 0,21 s
- **Número de peticiones por segundo:** 81,46 peticiones/s

Segunda prueba

- **Número de usuarios concurrentes:** 12
- **Tiempo de la prueba:** 5 minutos
- **Número de peticiones ejecutadas:** 19.985 peticiones
- **Número de peticiones completadas:** 17.960 peticiones
- **Número de peticiones falladas:** 2.025 peticiones
- **Porcentaje de peticiones completadas:** 88.72%
- **Tiempo medio de respuesta:** 0,13 s
- **Número de peticiones por segundo:** 66.62 peticiones/s

Conclusiones

Detectamos después de ejecutar las pruebas, que la carga máxima que soporta la aplicación en el anterior nodo vendrá dada por un máximo de X usuarios. Estas medidas hay que tomarlas con cautela debido a que se ha detectado que hay parte de las peticiones que no llegan al servidor AKS. La razón reside en las restricciones que impone Azure contra ataques de DDOS. Se puede consultar más información en [19].

Adicionalmente, Azure nos permite escalar el sistema de forma automática según las necesidades permitiendo tener una mayor carga si fuera necesario. A priori, este número de usuarios concurrentes será suficiente para el ámbito que necesita el sistema.

Prueba de Usabilidad y accesibilidad

Guía de usabilidad

Una vez finalizado el proceso de construcción, se pasará la guía de usabilidad desarrollada por Yusef Hassan Montero [13]:

Criterios	¿Cumplido?
<u>Generales</u>	
¿Cuáles son los objetivos del sitio web? ¿Son concretos y bien definidos? ¿Los contenidos y servicios que ofrece se corresponden con esos objetivos?	SI
¿Tiene una URL correcta, clara y fácil de recordar? ¿Y las URL de sus páginas internas? ¿Son claras y permanentes?	SI
¿Muestra de forma precisa y completa qué contenidos o servicios ofrece realmente el sitio web? El diseño de la página de inicio debe ser diferente al resto de páginas y cumplir la función de 'escaparate' del sitio.	SI
¿La estructura general del sitio web está orientada al usuario? Los sitios web deben estructurarse pensando en el usuario, sus objetivos y necesidades. La estructura interna de la empresa u organización, cómo funciona o se organiza no interesan al usuario.	SI
¿El look & feel general se corresponde con los objetivos, características, contenidos y servicios del sitio web? Ciertas combinaciones de colores ofrecen imágenes más o menos formales, serias o profesionales.	SI
¿Es coherente el diseño general del sitio web? Se debe mantener una coherencia y uniformidad en las estructuras y colores de todas las páginas. Esto sirve para que el usuario no se desoriente en su navegación.	SI
¿Es reconocible el diseño general del sitio web? Cuánto más se parezca el sitio web al resto de sitios web, más fácil será de usar.	NO



¿El sitio web se actualiza periódicamente? ¿Indica cuándo se actualiza? Las fechas que se muestren en la página deben corresponderse con actualizaciones, noticias, eventos...no con la fecha del sistema del usuario.	-
<u>Identidad e Información</u>	
¿Se muestra claramente la identidad de la empresa-sitio a través de todas las páginas?	SI
El Logotipo, ¿es significativo, identificable y suficientemente visible?	SI
El eslogan o <i>tagline</i> , ¿expresa realmente qué es la empresa y qué servicios ofrece?	-
¿Se ofrece algún enlace con información sobre la empresa, sitio web, 'webmaster'...?	SI
¿Se proporciona mecanismos para ponerse en contacto con la empresa? (email, teléfono, dirección postal, fax...)	SI
¿Se proporciona información sobre la protección de datos de carácter personal de los clientes o los derechos de autor de los contenidos del sitio web?	SI
En artículos, noticias, informes... ¿Se muestra claramente información sobre el autor, fuentes y fechas de creación y revisión del documento?	-
<u>Lenguaje y Redacción</u>	
¿El sitio web habla el mismo lenguaje que sus usuarios? Se debe evitar usar un lenguaje corporativista. Así mismo, hay que prestarle especial atención al idioma, y ofrecer versiones del sitio en diferentes idiomas cuando sea necesario.	SI
¿Emplea un lenguaje claro y conciso?	SI
¿Es amigable, familiar y cercano? Es decir, lo contrario a utilizar un lenguaje constantemente imperativo, mensajes crípticos, o tratar con "desprecio" al usuario.	SI
¿1 párrafo = 1 idea? Cada párrafo es un objeto informativo. Trasmite ideas, mensajes...Se deben evitar párrafos vacíos o varios mensajes en un mismo párrafo.	SI
<u>Rotulado</u>	
Los rótulos, ¿son significativos? Ejemplo: evitar rótulos del tipo "haga clic aquí".	SI
¿Usa rótulos estándar? Siempre que exista un "estándar" comúnmente aceptado para el caso concreto, como "Mapa del Sitio" o "Acerca de...".	-
¿Usa un único sistema de organización, bien definido y claro? No se deben mezclar diferentes. Los sistemas de organización son: alfabético, geográfico, cronológico, temático, orientado a tareas, orientado al público y orientado a metáforas.	SI
¿Utiliza un sistema de rotulado controlado y preciso? Por ejemplo, si un enlace tiene el rótulo "Quiénes somos", no puede dirigir a una página cuyo encabezamiento sea "Acerca de"	SI



El título de las páginas, ¿Es correcto? ¿Ha sido planificado? Relacionado con la capacidad para poder buscar y encontrar el sitio <i>web</i> .	SI
<u>Estructura y Navegación</u>	
La estructura de organización y navegación, ¿Es la más adecuada? Hay varios tipos de estructuras: jerárquicas, hipertextual, facetada...	SI
En el caso de estructura jerárquica, ¿Mantiene un equilibrio entre Profundidad y Anchura?	SI
En el caso de ser puramente hipertextual, ¿Están todos los clústeres de nodos comunicados? Aquí se mide la distancia entre nodos.	-
¿Los enlaces son fácilmente reconocibles como tales? ¿Su caracterización indica su estado (visitados, activos...)? Los enlaces no sólo deben reconocerse como tales, sino que su caracterización debe indicar su estado, y ser reconocidos como una unidad	SI
En menús de navegación, ¿Se ha controlado el número de elementos y de términos por elemento para no producir sobrecarga memorística? No se deben superar los 7±2 elementos, ni los 2 o, como mucho, 3 términos por elemento.	SI
¿Es predecible la respuesta del sistema antes de hacer clic sobre el enlace? Relacionado con el nivel de significación del rótulo del enlace, aunque también con: el uso de globos de texto, información contextual, la barra de estado del navegador...	SI
¿Se ha controlado que no haya enlaces que no lleven a ningún sitio? Enlaces que no llevan a ningún sitio: Los enlaces rotos, y los que enlazan con la misma página que se está visualizando (por ejemplo, enlaces a la "home" desde la misma página de inicio)	SI
¿Existen elementos de navegación que orienten al usuario acerca de dónde está y cómo deshacer su navegación? ...como <i>breadcrumbs</i> , enlaces a la página de inicio...recuerde que el logo también es recomendable que enlace con la página de inicio.	SI
Las imágenes enlace, ¿se reconocen como clicables? ¿Incluyen un atributo 'title' describiendo la página de destino? En este sentido, también hay que cuidar que no haya imágenes que parezcan enlaces y en realidad no lo sean.	-
¿Se ha evitado la redundancia de enlaces?	SI
¿Se ha controlado que no haya páginas "huérfanas"? Páginas huérfanas: que, aun siendo enlazadas desde otras páginas, éstas no enlacen con ninguna.	SI
<u>Layout de la Página</u>	
¿Se aprovechan las zonas de alta jerarquía informativa de la página para contenidos de mayor relevancia? (como por ejemplo la zona central)	SI
¿Se ha evitado la sobrecarga informativa? Esto se consigue haciendo un uso correcto de colores, efectos tipográficos y agrupaciones para discriminar información. Los grupos diferentes de objetos informativos de una página deben ser 7±2.	SI



¿Es una interfaz limpia, sin ruido visual?	SI
¿Existen zonas en "blanco" entre los objetos informativos de la página para poder descansar la vista?	-
¿Se hace un uso correcto del espacio visual de la página? Es decir, que no se desaproveche demasiado espacio con elementos de decoración, o grandes zonas en "blanco", y que no se adjudique demasiado espacio a elementos de menor importancia.	SI
¿Se utiliza correctamente la jerarquía visual para expresar las relaciones del tipo "parte de" entre los elementos de la página? (La jerarquía visual se utiliza para orientar al usuario)	N/A
¿Se ha controlado la longitud de página? Se debe evitar en la medida de lo posible el <i>scrolling</i> . Si la página es muy extensa, se debe fraccionar.	SI
<i>Búsqueda (si es necesario, por la extensión del sitio, incorporar un buscador interno)</i>	
¿Se encuentra fácilmente accesible? Es decir: directamente desde la página home, y a ser posible desde todas las páginas del sitio, y colocado en la zona superior de la página.	SI
¿Es fácilmente reconocible como tal?	SI
¿Permite la búsqueda avanzada? (siempre y cuando, por las características del sitio web, fuera de utilidad que la ofreciera)	-
¿Muestra los resultados de la búsqueda de forma comprensible para el usuario?	-
¿La caja de texto es lo suficientemente ancha?	-
¿Asiste al usuario en caso de no poder ofrecer resultados para una consultada dada?	-
<i>Elementos Multimedia</i>	
¿Las fotografías están bien recortadas? ¿Son comprensibles? ¿Se ha cuidado su resolución?	SI
¿Las metáforas visuales son reconocibles y comprensibles por cualquier usuario? (prestar especial atención a usuarios de otros países y culturas)	SI
¿El uso de imágenes o animaciones proporciona algún tipo de valor añadido?	NO
¿Se ha evitado el uso de animaciones cíclicas?	SI
<i>Ayuda</i>	
Si posee una sección de Ayuda, ¿Es verdaderamente necesaria? Siempre que se pueda prescindir de ella simplificando los elementos de navegación e interacción, debe omitirse esta sección.	-
En enlace a la sección de Ayuda, ¿Está colocado en una zona visible y "estándar"? La zona de la página más normal para incluir el enlace a la sección de Ayuda es la superior derecha.	-
¿Se ofrece ayuda contextual en tareas complejas? (transferencias bancarias, formularios de registro...)	-



Si posee FAQs, ¿Es correcta tanto la elección como la redacción de las preguntas? ¿Y las respuestas?	-
<u>Accesibilidad</u>	
¿El tamaño de fuente se ha definido de forma relativa, o por lo menos, la fuente es lo suficientemente grande como para no dificultar la legibilidad del texto?	SI
¿El tipo de fuente, efectos tipográficos, ancho de línea y alineación empleadas facilitan la lectura?	SI
¿Existe un alto contraste entre el color de fuente y el fondo?	SI
¿Incluyen las imágenes atributos 'alt' que describan su contenido?	SI
¿Es compatible el sitio web con los diferentes navegadores? ¿Se visualiza correctamente con diferentes resoluciones de pantalla? Se debe prestar atención a: JScript, CSS, tablas, fuentes...	SI (mínimo requerido 1080p)
¿Puede el usuario disfrutar de todos los contenidos del sitio web sin necesidad de tener que descargar e instalar <i>plugins</i> adicionales?	SI
¿Se ha controlado el peso de la página? Se deben optimizar las imágenes, controlar el tamaño del código JScript...	NO
¿Se puede imprimir la página sin problemas? Leer en pantalla es molesto, por lo que muchos usuarios preferirán imprimir las páginas para leerlas. Se debe asegurar que se puede imprimir la página (no salen partes cortadas), y que el resultado es legible.	SI
<u>Control y Retroalimentación</u>	
¿Tiene el usuario todo el control sobre el interfaz? Se debe evitar el uso de ventanas pop-up, ventanas que se abren a pantalla completa, banners intrusivos...	SI
¿Se informa constantemente al usuario acerca de lo que está pasando? Si el usuario tiene que esperar hasta que se termine una operación, se debe mostrar un mensaje indicándole y que debe esperar, con el tiempo de espera estimado o una barra de progreso.	SI
¿Se informa al usuario de lo que ha pasado? Por ejemplo, cuando un usuario valora un artículo o responde a una encuesta, se le debe informar de que su voto ha sido procesado correctamente.	SI
Cuando se produce un error, ¿se informa de forma clara y no alarmista al usuario de lo ocurrido y de cómo solucionar el problema? Siempre es mejor intentar evitar que se produzcan errores a tener que informar al usuario del error.	SI
¿Posee el usuario libertad para actuar? NO restringir la libertad del usuario: Uso de animaciones que no pueden ser "saltadas", páginas en las que desaparecen los botones de navegación, no impida al usuario poder usar el botón derecho de su ratón...	SI



¿Se ha controlado el tiempo de respuesta? Esto tiene que ver con el peso de cada página (accesibilidad) y tiene relación con el tiempo que tarda el servidor en finalizar una tarea y responder. El tiempo máximo que esperará un usuario son 10 segundos	SI
<u>Aclaraciones</u>	
¿Se ha evaluado adecuadamente la orientación del usuario? (Donde estoy, como volver, que he visitado, que va a pasar) ²	SI
¿Se ha usado correctamente la publicidad? ³	-

Tabla 245. Checklist de pruebas de Usabilidad

Cuestionarios

Una vez definido el anterior cuestionario, y obtener buenos resultados, se han hecho pruebas con usuarios reales para avalarlo.

Usuario 1

Preguntas de carácter general

<p>¿Usa un ordenador frecuentemente?</p> <p>5. Todos los días 6. Varias veces a la semana 7. Ocasionalmente 8. Nunca o casi nunca</p>
<p>¿Qué tipo de actividades realiza con el ordenador?</p> <p>5. Es parte de mi trabajo o profesión 6. Lo uso básicamente para ocio 7. Solo empleo aplicaciones estilo Office 8. Únicamente leo el correo y navego ocasionalmente</p>
<p>¿Ha usado alguna vez software como el de esta prueba?</p> <p>4. Sí, he empleado software similar 5. No, aunque si empleo otros programas que me ayudan a realizar tareas similares 6. No, nunca</p>
<p>¿Qué busca Vd. Principalmente en un programa?</p> <p>5. Que sea fácil de usar 6. Que sea intuitivo 7. Que sea rápido 8. Que tenga todas las funciones necesarias</p>

Actividades guiadas

6. Añadir un usuario a la aplicación



Completado: Si

Dificultades: Bajas

Adaptación: Rápida

Tiempo: 2 minutos

7. Añadir visión

Completado: Si

Dificultades: Bajas

Adaptación: Rápida

Tiempo: 1 minutos

8. Clonar visión

Completado: No

Dificultades: No encuentra el botón de clonar

Adaptación: Mala

Tiempo: 3 minutos

Observaciones: Cambiar el color de los botones de clonar

9. Clonar FactoryModel

Completado: Si

Dificultades: No

Adaptación: Rápida

Tiempo: 2,5 minutos

10. Modificar datos en FactoryModel: Energía

Completado: Si



Dificultades: No

Adaptación: Rápida

Tiempo: 2 minutos

Preguntas Cortas sobre la Aplicación y Observaciones

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Sabe dónde está dentro de la aplicación?</i>	X			
<i>¿Existe ayuda para las funciones en caso de que tenga dudas?</i>				X
<i>¿Le resulta sencillo el uso de la aplicación?</i>	X			
<i>¿Le ha costado adaptarse al funcionamiento?</i>				X
<i>¿Conoce la funcionalidad de la aplicación?</i>	X			
<i>¿Cree que la aplicación tiene un aspecto similar a una hoja de cálculos?</i>	X			
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Funciona cada tarea como Vd. Espera?</i>	X			
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>				X
<i>¿Tiene la sensación de no conocer el estado de las acciones que realiza?</i>		X		
<i>¿Considera que los cálculos hechos son correctos?</i>	X			
<i>¿Considera que le puede aportar algo si realiza o realizase una actividad económica?</i>	X			
<i>¿Considera que está satisfecho con el funcionamiento de la app?</i>		X		
Calidad del Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
<i>El tipo y tamaño de letra es</i>		X		
<i>Los iconos e imágenes usados son</i>		X		
<i>Los colores empleados son</i>		X		
<i>La aplicación está adaptada para un entorno empresarial y comercial</i>	X			
Diseño de la Interfaz		Si	No	A veces
<i>¿Le resulta fácil de usar?</i>		X		
<i>¿El diseño de las pantallas es claro y atractivo?</i>		X		



¿Cree que el programa está bien estructurado?	X		
¿Cree que el programa es usable?	X		
¿Modificaría alguna de las páginas?			X
Observaciones			
<ul style="list-style-type: none"> - Cambio en la forma de mostrar la internalización. - Cambio en el color de los botones de activación de un FM Component. 			

Cuestionario para el responsable de las Pruebas

Aspecto Observado	Notas
<i>El usuario comienza a trabajar de forma rápida por las tareas</i>	El usuario sabe interpretar la mayoría de los aspectos de la aplicación.
<i>Tiempo en realizar cada tarea</i>	El usuario ha realizado las tareas de forma rápida. A excepción de la tarea de Clonar una visión. Ha comentado que no ha encontrado el botón para ejecutar la acción.
<i>Errores leves cometidos</i>	Al pedir internalizaciones la aplicación, el usuario no ha sabido realizarlo.
<i>Errores graves cometidos</i>	Ninguno.

Usuario 2

Preguntas de carácter general

<p>¿Usa un ordenador frecuentemente?</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>¿Qué tipo de actividades realiza con el ordenador?</p> <ol style="list-style-type: none"> 1. Es parte de mi trabajo o profesión 2. Lo uso básicamente para ocio 3. Solo empleo aplicaciones estilo Office 4. Únicamente leo el correo y navego ocasionalmente
<p>¿Ha usado alguna vez software como el de esta prueba?</p> <ol style="list-style-type: none"> 1. Sí, he empleado software similar 2. No, aunque si empleo otros programas que me ayudan a realizar tareas similares 3. No, nunca
<p>¿Qué busca Vd. Principalmente en un programa?</p> <ol style="list-style-type: none"> 1. Que sea fácil de usar 2. Que sea intuitivo 3. Que sea rápido



4. Que tenga todas las funciones necesarias

Actividades guiadas

1. Añadir un usuario a la aplicación

Completado: Sí

Dificultades: Baja

Adaptación: Rápida

Tiempo: 1 minuto

2. Añadir visión

Completado: Sí

Dificultades: Baja

Adaptación: Rápida

Tiempo: 1 minuto

3. Clonar visión

Completado: Sí

Dificultades: Baja

Adaptación: Rápida

Tiempo: 1 minuto

4. Clonar FactoryModel

Completado: Sí

Dificultades: Media

Adaptación: Rápida

Tiempo: 2 minutos

5. Modificar datos en FactoryModel: Energía



Completado: Sí

Dificultades: Media

Adaptación: Media

Tiempo: 5 minutos

Observaciones: Añadir un texto de “Modificar” al lado del botón. Añadir una leyenda de los colores empleados en las celdas.

Preguntas Cortas sobre la Aplicación y Observaciones

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Sabe dónde está dentro de la aplicación?	X			
¿Existe ayuda para las funciones en caso de que tenga dudas?		X		
¿Le resulta sencillo el uso de la aplicación?	X			
¿Le ha costado adaptarse al funcionamiento?				X
¿Conoce la funcionalidad de la aplicación?	X			
¿Cree que la aplicación tiene un aspecto similar a una hoja de cálculos?	X			
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Funciona cada tarea como Vd. Espera?	X			
¿El tiempo de respuesta de la aplicación es muy grande?				X
¿Tiene la sensación de no conocer el estado de las acciones que realiza?				X
¿Considera que los cálculos hechos son correctos?	X			
¿Considera que le puede aportar algo si realiza o realizase una actividad económica?	X			
¿Considera que está satisfecho con el funcionamiento de la app?	X			
Calidad del 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			



<i>La aplicación está adaptada para un entorno empresarial y comercial</i>	X			
Diseño de la Interfaz		Si	No	A veces
<i>¿Le resulta fácil de usar?</i>	X			
<i>¿El diseño de las pantallas es claro y atractivo?</i>	X			
<i>¿Cree que el programa está bien estructurado?</i>	X			
<i>¿Cree que el programa es usable?</i>	X			
<i>¿Reestructuraría alguna de las páginas?</i>			X	
Observaciones				
Cualquier comentario del usuario				

Cuestionario para el responsable de las Pruebas

Aspecto Observado	Notas
<i>El usuario comienza a trabajar de forma rápida por las tareas</i>	Sí.
<i>Tiempo en realizar cada tarea</i>	Normal.
<i>Errores leves cometidos</i>	Al editar un campo de una visión en la tabla, tras darle a "Guardar cambios", visualmente sigue saliendo el campo como si fuese editable, no obstante, si se hace click accede a los detalles de la misma.
<i>Errores graves cometidos</i>	Ninguno.

Usuario 3

Preguntas de carácter general

¿Usa un ordenador frecuentemente?
<ol style="list-style-type: none"> 1. Todos los días 2. Varias veces a la semana 3. Ocasionalmente 4. Nunca o casi nunca
¿Qué tipo de actividades realiza con el ordenador?
<ol style="list-style-type: none"> 1. Es parte de mi trabajo o profesión 2. Lo uso básicamente para ocio 3. Solo empleo aplicaciones estilo Office 4. Únicamente leo el correo y navego ocasionalmente
¿Ha usado alguna vez software como el de esta prueba?
<ol style="list-style-type: none"> 1. Sí, he empleado software similar 2. No, aunque si empleo otros programas que me ayudan a realizar tareas similares 3. No, nunca
¿Qué busca Vd. Principalmente en un programa?



1. Que sea fácil de usar
2. Que sea intuitivo
3. Que sea rápido
4. **Que tenga todas las funciones necesarias**

Actividades guiadas

1. Añadir un usuario a la aplicación

Completado: Si

Dificultades: Encontrar el apartado de Administración de Usuarios

Adaptación: Difícil

Tiempo: 5 minutos

2. Añadir visión

Completado: Si

Dificultades: Ninguna

Adaptación: Rápida

Tiempo: 1 minuto

3. Clonar visión

Completado: Sí

Dificultades: Entendimiento de la palabra clonar

Adaptación: Costosa

Tiempo: 4 minutos

4. Clonar FactoryModel

Completado: No

Dificultades: No encuentra la lista de FactoryModel

Adaptación: Varios niveles de complejidad para llegar al listado



Tiempo: 6 minutos

5. Modificar datos en FactoryModel: Energía

Completado: Sí

Dificultades: Ninguna

Adaptación: Fácil

Tiempo: 2 minutos

Preguntas Cortas sobre la Aplicación y Observaciones

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Sabe dónde está dentro de la aplicación?</i>	X			
<i>¿Existe ayuda para las funciones en caso de que tenga dudas?</i>				X
<i>¿Le resulta sencillo el uso de la aplicación?</i>		X		
<i>¿Le ha costado adaptarse al funcionamiento?</i>	X			
<i>¿Conoce la funcionalidad de la aplicación?</i>			X	
<i>¿Cree que la aplicación tiene un aspecto similar a una hoja de cálculos?</i>	X			
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Funciona cada tarea como Vd. Espera?</i>		X		
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>	X			
<i>¿Tiene la sensación de no conocer el estado de las acciones que realiza?</i>				X
<i>¿Considera que los cálculos hechos son correctos?</i>	X			
<i>¿Considera que le puede aportar algo si realiza o realizase una actividad económica?</i>	X			
<i>¿Considera que está satisfecho con el funcionamiento de la app?</i>		X		
Calidad del Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
<i>El tipo y tamaño de letra es</i>		X		
<i>Los iconos e imágenes usados son</i>		X		
<i>Los colores empleados son</i>	X			

La aplicación está adaptada para un entorno empresarial y comercial	X			
Diseño de la Interfaz		Si	No	A veces
¿Le resulta fácil de usar?				X
¿El diseño de las pantallas es claro y atractivo?				X
¿Cree que el programa está bien estructurado?	X			
¿Cree que el programa es usable?	X			
¿Modificaría alguna de las páginas?			X	
Observaciones				
Permitir al usuario almacenar la información relativa al idioma y moneda en su cuenta.				

Cuestionario para el responsable de las Pruebas

Aspecto Observado	Notas
El usuario comienza a trabajar de forma rápida por las tareas	El usuario ha sabido ejecutar las tareas una vez que ya las ha realizado. No ha conseguido entender donde se encuentra el apartado de administración.
Tiempo en realizar cada tarea	Relativamente rápido para el nivel de conocimiento de la aplicación que disponía.
Errores leves cometidos	No encontrar que la tabla es <i>clickable</i> .
Errores graves cometidos	Ninguno

Prueba de Código

A continuación, se mostrará un resumen de los resultados obtenidos con relación al código desarrollado como salida del proyecto:

Resumen	
Prueba	Resultado medio
Índice de mantenimiento	85,66
Complejidad ciclomática	22,31
Profundidad de herencia	3,16
Líneas de código	10,04

Tabla 246. Resultado pruebas de código

Todos los resultados se encuentran dentro de los valores que se dieron como aceptables. Para obtener un mayor detalle de los resultados, se adjunta un documento Excel con el informe detallado con los datos donde se puede comprobar las métricas de código llamada Pruebas de Código en formato MS Excel.



CSI 6: ELABORACIÓN DE LOS MANUALES DE USUARIO

Manual de Instalación-Ejecución

El manual de usuario se puede consultar en un archivo anexo externo llamado **“Manual de Instalación-Ejecución”** en formato PDF.

Manual de Usuario

El manual de usuario se puede consultar en un archivo anexo externo llamado **“Manual de Usuario”** en formato PDF.



CSI 8: CONSTRUCCIÓN DE LOS COMPONENTES Y PROCEDIMIENTOS DE MIGRACIÓN Y CARGA INICIAL DE DATOS

En la aplicación, el despliegue de la carga de datos inicial se hace de forma automática si el sistema detecta que no existe la base de datos. Si por consiguiente, se quiere ejecutar una restauración de los datos iniciales, se dispone de un script llamado `deploy/migrations` que permite hacerlo.



Capítulo 6 IMPLANTACIÓN Y ACEPTACIÓN DEL SISTEMA

FASE DE DESARROLLO

IAS



IAS 1: ESTABLECIMIENTO DEL PLAN DE IMPLANTACIÓN

El plan de implantación para esta aplicación debe de cumplir con los siguientes prerequisites:

- Dado que una empresa puede administrar todos los sistemas relativos al producto, el plan de actuación debe de estar preparado para que cualquier usuario pueda ejecutarlos. Para solucionar este inconveniente, se decide hacer uso de *scripts* automáticos parametrizables para la implantación del sistema.
- La formación necesaria de la persona encargada del despliegue consistirá en nociones básicas de *scripting* y de los servicios Azure (Administración de un servicio).

Se pasa a definir el sistema involucrado y sus posibles relaciones:

- El sistema estará formado por los servicios desplegados en el AKS. El sistema se mantendrá desplegado siempre en el sistema que proporciona la empresa al Cliente o en un servicio desplegado por el propio Cliente donde la empresa será administradora del sistema.
- Dado que el sistema es un producto único, no tiene ninguna relación con el resto del sistema y no es necesario disponer de acceso a ningún sistema de información de la empresa contratante.
- El sistema se debe de mantener actualizado por los responsables de soporte dados por el Cliente de forma transparente a la empresa contratante.
- El sistema deberá ser desplegado por al menos una persona que disponga de permisos de Administrador sobre el *cluster* de AKS.

Finalmente se comentarán los criterios que se seguirán para que el sistema pueda ser aceptado por el Cliente y la empresa contratante:

- El sistema debe de funcionar correctamente en el dominio donde se desea desplegar.
- En caso de que la empresa contratante quiera, el sistema debe de poder soportar una carga de trabajo definida por la propia empresa. En caso negativo, se debe de realizar un escalado del servidor AKS.



IAS 4: CARGA DE DATOS AL ENTORNO DE OPERACIÓN

Los diferentes microservicios son transparentes con la carga inicial de datos a la base de datos. El esquema de la base de datos y las tablas maestras son cargadas en la primera iteración del arranque del sistema de forma transparente.

Si por alguna razón, se necesita regenerar algún sistema de datos de alguno de los servicios, se dispone de un script **deploy/migrations** que realiza el mismo proceso anteriormente descrito.

IAS 5: PRUEBAS DE IMPLANTACIÓN DEL SISTEMA

Tomando con referencia los aspectos tratados en [20] y [21], a continuación, se muestran las pruebas de seguridad seguidas antes de ejecutar test de vulnerabilidades sobre el servidor desplegado:

	SI	NO	N/A
El banner del servidor ya no informa sobre su versión o da información sobre el SO subyacente	X		
Se ha deshabilitado el listado de directorios, salvo que sea una funcionalidad buscada (en cuyo caso solo está activa donde sea estrictamente necesario)	X		
Se ha deshabilitado la cabecera ETag		X	
El servidor NO se ejecuta como usuario root	X		
El directorio de instalación del servidor tiene permisos adecuados	X		
No se permite a los usuarios sobrescribir la configuración del servidor		X	
Se limita al mínimo el nº de métodos HTTP permitidos	X		
Se deshabilita el método HTTP TRACE	X		
Se configura la cabecera X-Frame-Options para que solo se permitan IFrames desde el mismo dominio de la web	X		
Se deshabilitan los Server-Side Includes, salvo que sean necesarios para la funcionalidad		X	
Se habilita la cabecera X-XSS-Protection	X		
Se deshabilita el soporte de HTTP 1.0	X		
El Timeout por defecto se configura a unos 60 segundos	X		
Se usa SSL con una clave de longitud 2048 por lo menos			X
Se usa SSL con un algoritmo de cifrado seguro			X
Se deshabilita el soporte de SSLv2 y SSLv3, admitiéndose solo TLSv1.2 o cualquier versión posterior no considerada vulnerable			X
Se usa el WAF mod_security y sus core rules	X		
Se hace log de forma adecuada			X
Solo se usan los módulos estrictamente necesarios	X		

Tabla 247. Medidas de seguridad del servidor instalado

Por último, se ejecutará una prueba de vulnerabilidades sobre el servidor de desarrollo desplegado para tal efecto. Los resultados obtenidos son los siguientes:



- **Ataques de inyección SQL:** Los campos introducidos por el usuario se encuentran protegidos contra ataques de inyección de SQL tanto en el servidor como en el cliente.
- **Ataques XSS:** Se comentarán los resultados en base a su tipo:
 - **Reflected Cross-Site Scripting:** Cuando se introduce algo de código por parámetros es tratado como un error.
 - **Stored Cross-Site Scripting:** Se ha procedido a guardar un script de JS en el SGBD y el resultado ha sido un fracaso.
 - **XEE:** En este caso, el ataque por inyección XML no es posible. No se trata ningún elemento en XML y el servidor solo puede trabajar con ficheros con tipo *application/json* con una sintaxis de JSON.



IAS 7: PREPARACIÓN DEL MANTENIMIENTO DEL SISTEMA

El sistema, tal como se encuentra planteado, no requiere ningún mantenimiento adicional a la suscripción necesaria de un servicio AKS.

Las actualizaciones se desarrollan de forma automática al sistema, a través de los *pipelines* generadas en el momento del despliegue del sistema al Cliente.



Capítulo 7 APÉNDICES



PROBLEMAS ENCONTRADOS DURANTE EL DESARROLLO

A continuación, se detallará la lista de principales problemas que se han encontrado durante el desarrollo del sistema.

Comunicación entre microservicios

Alcanzar un sistema de comunicación automático que funcione de forma abstracta al sistema fue difícil de lograr. Existe muy poca información relativa sobre el tema de cómo se puede hacer una implementación en .NET evitando cargar esos datos sobre un servicio alojado en MS Azure.

Hay multitud de librerías que permiten mandar mensajes a una cola, pero la mayoría, no estaban adaptadas a .NET Core en el momento inicial de desarrollo teniendo que buscar otras soluciones.

Finalmente, se decidió hacer uso de sistema basado en mensajes de **RabbitMQ** usando la librería **RawRabbit**.

Clonación de entidades

Aunque clonar entidades es algo bastante esencial en aplicaciones de gestión, no es una tarea sencilla al llevarla a cabo en una arquitectura basada en microservicios. Debido a su dificultad, durante el desarrollo, se llegó a plantear la posibilidad de migrar parte del sistema a un solo microservicio. Finalmente, se logró alcanzar una solución, aunque no es la más adecuada.

El sistema se planteó como un sistema síncrono donde el encargado de la clonación era el microservicio que se encontraba escuchando el evento. Esto tiene varios problemas: no se puede conocer si un microservicio está disponible, se crea acoplamiento indirecto entre ellos y, según muchos expertos, se puede considerar un *anti-patrón*. El siguiente diagrama, viene a definir este sistema.

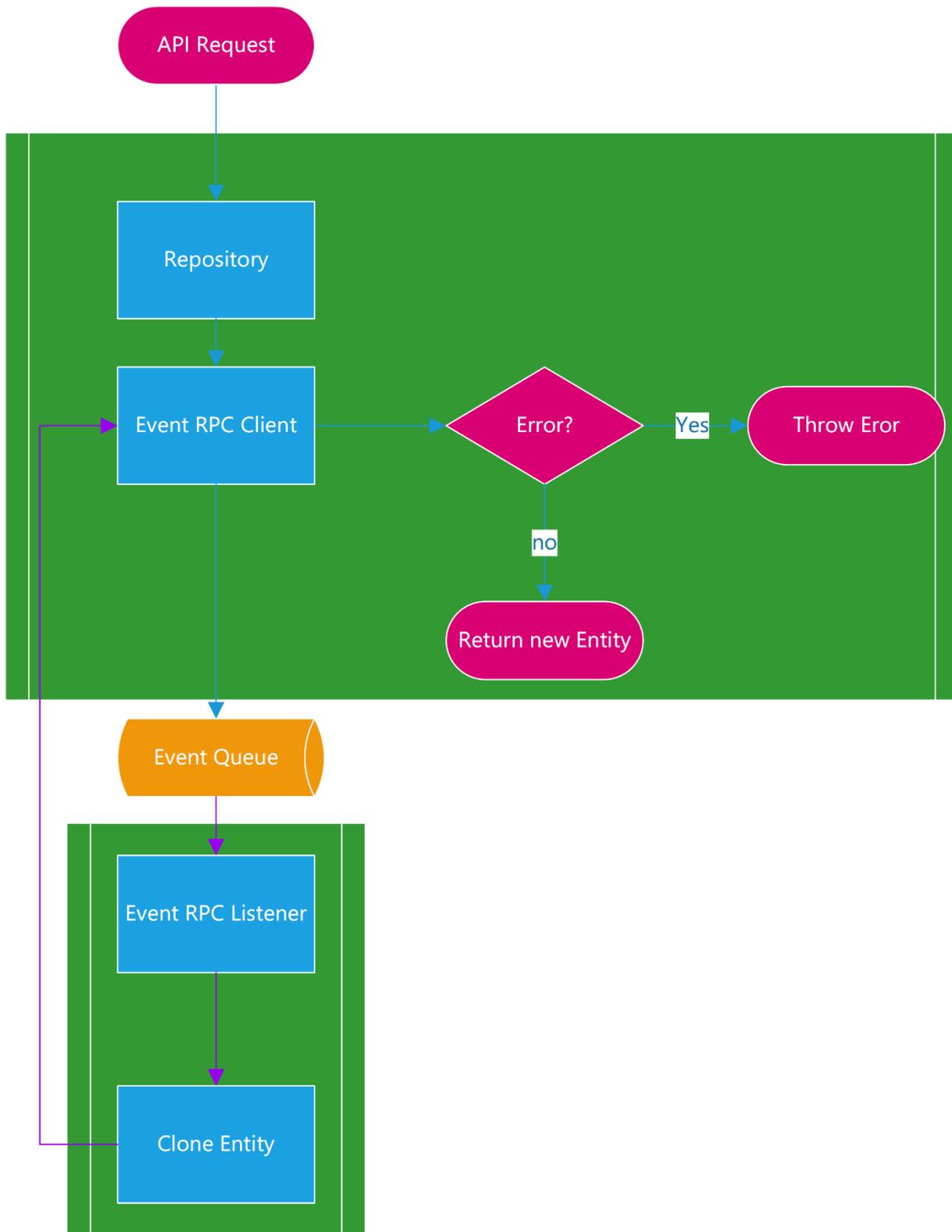


Fig 75. Diagrama de flujo del primer diseño para la clonación

La opción finalmente adaptada, fue hacer un sistema asíncrono donde cada microservicio clona todas sus entidades y les envía un mensaje a los microservicios que están escuchando con los datos que este ha generado.



AMPLIACIONES

Aunque el sistema se encuentra en una fase de producción, a lo largo del desarrollo, se han planteado varias ampliaciones por parte del Cliente/equipo de desarrollo.

Nuevos componentes de FactoryModel

Debido al carácter que tiene la aplicación de parametrización, es importante, que la aplicación no sea estática y se mantenga actualizada con nueva funcionalidad. La funcionalidad viene dada por los componentes que forman un FactoryModel. Posibles ideas de nuevos componentes podrían ser:

- **Mantenimiento:** En empresas donde la tarea de fabricación y los costes de mantenimiento, puede ser realmente útil tenerlos en cuenta como un componente independiente.
- **Costes de mano de obra:** Al igual que el anterior, es interesante contemplar un módulo de coste de mano de obra en mayor detalle.
- **Parametrización del módulo de Energía:** Actualmente, la aplicación está haciendo uso de las directrices marcadas por los contratos eléctricos españoles de tipo 3.X. Aunque la aplicación, soporta diferentes contratos, no se encuentran configurados. Quedaría seleccionar y definir que contratos eléctricos son interesantes.
- **OEE:** En actividades económicas donde es parte fundamental el equipamiento, sería interesante implementar un control de OEE (Eficiencia general de los equipos). Sería una ampliación al módulo de Producción actualmente desarrollado.
- **Existencias:** Gestión de almacenes con el objetivo de obtener la relación óptima entre el coste de almacenamiento y el espacio disponible.
- **Inmovilizado:** Módulo destinado al ajuste de la infraestructura a la producción efectiva, evitando posibles excesos de planta.

Vista detalle de una visión y exportación de los datos finales

Una vez hechos los cálculos y cerrado una visión, la aplicación se convierte en demasiado compleja para consultar datos. Por dicha razón, una posible idea sería realizar una tabla que disponga de todas las visiones donde poder consultar de forma rápida los FactoryModel publicados.

Además, sería interesante permitir a un usuario exportar los datos a un sistema de Hojas de Cálculo para disponer de una mejor integración con su sistema de contabilidad.



Soporte para *ActiveDirectory*

Window/Azure Active Directory es un sistema muy usado en entornos empresarias que permiten iniciar sesión en varios servicios haciendo uso del servidor central de autenticación. Sería interesante también, tratar con un sistema de inicio en sesión automático en la aplicación haciendo uso de él.

Separación en módulos del Cliente Web

El módulo del Cliente Web se encuentra desarrollado en Angular. Aunque en un primer momento se planteó como una arquitectura monolítica, es interesante, modularizar la aplicación para mejorar los tiempos de espera y que sea más parametrizable en el despliegue.

En una segunda iteración, se podría llevar rehacer los módulos para convertirlos en módulos independientes haciendo uso de *Front-end microservices – Micro Frontends*.

Nueva implementación de Tablas de Hojas de Calculo

Actualmente, existe una versión muy sencilla de hoja de cálculos. Aunque para una primera aproximación es suficiente, sería apropiado diseñar e implementar un componente más complejo que realice las tareas de tabla de datos de forma óptima.

Mejora en la clonación de Entidades

Actualmente, se puede clonar entidades de forma correcta. La implementación que se ha hecho no es la más adecuada y tiene varios problemas de mantenimiento. Una mejor opción sería diseñar un sistema que permita hacer clonaciones más sencillas de forma más abstracta al programador.

Dashboard

Una forma ideal de mostrar los datos de una forma visual es disponer de un *dashboard* que nos muestre toda la información relativa a la aplicación. Se convertiría de esta forma, en la pantalla principal del sistema.



CONCLUSIONES

En esta sección, se va a hacer recapitulación de lo tratado a lo largo del documento. Después de haber finalizado el proyecto, se puede considerar que se ha alcanzado el objetivo principal del proyecto. Se ha conseguido completar el desarrollo de un sistema software que nos permita simplificar y facilitar la implementación de técnicas de *Cost Deployment*. Se ha completado con éxito un análisis del sistema pedido por el Cliente con su posterior validación y aceptación. Se ha diseñado un sistema mantenible, parametrizable y muy configurable cumplimiento con uno de los principales atributos de calidad que fueron marcados por el Cliente, mantenibilidad.

Este documento ha ido evolucionando y validándose a lo largo del desarrollo, para conocer posibles fallos y defectos que pudieran aparecer. Tanto el equipo de desarrollo como el Cliente se encuentran satisfechos con el resultado obtenido.

Al iniciar el proyecto, se buscaba disponer de un sistema web que implementará técnicas de *Cost Deployment*. El primer objetivo que se buscó fue adquirir una visión y conocimiento de la fuente de dominio. Aunque es la base del proyecto, fue necesario ampliar la visión del dominio a temas mucho más concretos como análisis económicos, fuentes de costes de una empresa, conocimientos en energías eléctricas o en la manufactura y producción de recursos. Visto el resultado del proyecto, se considera que se han adquirido los conocimientos para realizar un sistema consistente en todos estos aspectos.

Después de revisar el resultado del software analizado, diseñado e implementado, se ha llegado a la conclusión de que las herramientas que se han usado han sido las mejores opciones para ello. Desde un primer momento se puso como restricción que el software tenía que vivir y residir en la nube, haciendo uso de servicios actualmente punteros. Aunque realizar el desarrollo de un sistema basada en microservicios no es algo sencillo, fue la mejor elección que se pudo hacer en tema de diseño. Nos ha aportado ventajas como independencia entre módulos, parametrización, reutilización de software o una mayor capacidad de soporte para el Cliente con actualizaciones de mejora del producto o con actualizaciones incrementales para arreglo de fallos.

En conclusión, todos los interesados del proyecto consideran que los resultados han sido muy destacables y que se han cumplido de forma muy satisfactoria y con éxito los objetivos inicialmente marcados.

CONTENIDO ENTREGADO EN LOS ANEXOS

Junto a dicho documento, se adjunta un fichero comprimido que contiene el código fuente, diversos scripts y documentación derivada del desarrollo del sistema. La estructura en detalle que tiene dicho fichero será la siguiente:

Directorio	Contenido
./	Contiene un fichero README .TXT explicando la estructura de los entregable.
./FactoryModel	Contiene toda la estructura de directorios del proyecto para desarrollo. Contiene la solución de Visual Studio que permite ejecutar el proyecto.
./FactoryModel/src	Contiene el código fuente de la aplicación.
./FactoryModel/src/Client	Contiene el código fuente de un proyecto de Visual Studio con el cliente desarrollado en Angular. El código se puede encontrar en la carpeta ClientApp.
./FactoryModel/src/Common	Librerías usadas por los diferentes servicios que conforman el <i>back-end</i> de la aplicación.
./FactoryModel/src/Services	Contiene todos los servicios <i>back-end</i> del sistema. Cada carpeta almacena un proyecto que funciona de forma independiente a los demás.
./FactoryModel/test	Contiene las pruebas unitarias ejecutados a la aplicación. Está formado por un proyecto categorizado por el servicio o librería que se quiere probar.
./FactoryModel/deploy	<i>Scripts</i> utilizados para la configuración del servidor AKS de forma automática. Tiene compatibilidad con sistemas Windows (ps1) y con sistemas Unix (sh). Se puede adquirir un mayor detalle de que hace cada fichero en el fichero de documentación Manual de Instalación.
./Documentación	Contiene la documentación asociada al proyecto en PDF. Se detallará a continuación.

Tabla 248. Estructura general de los ficheros anexos entregados

Para finalizar, se detallarán los ficheros externos que complementan a este documento:

Fichero	Contenido
./Documentación/Fórmulas.xlsx	Fichero MS Excel que contiene las fórmulas usadas para la definición de los componentes de un FactoryModel. Disponen de unos cálculos de ejemplo.
./Documentación/Planificación.xlsx	Fichero MS Excel que contiene la planificación final del desarrollo del sistema software.



./Documentación/Planificación Inicial.xlsx	Fichero MS Excel que contiene la planificación inicial del desarrollo del sistema software.
./Documentación/Presupuesto.xlsx	Fichero MS Excel que contiene el presupuesto final de costes y del cliente del proyecto.
./Documentación/Presupuesto.pdf	Fichero PDF con un resumen del presupuesto final de costes y del cliente del proyecto.
./Documentación/Presupuesto Inicial.xlsx	Fichero MS Excel que contiene el presupuesto inicial de costes y del cliente del proyecto.
./Documentación/Presupuesto Inicial.pdf	Fichero PDF con un resumen del presupuesto inicial de costes y del cliente del proyecto.
./Documentación/Manual de Usuario.pdf	Fichero PDF con el manual de usuario del sistema de información.
./Documentación/Manual de Instalación-Ejecución.pdf	Fichero PDF con el manual de instalación y ejecución del sistema de información.
./Documentación/Presentación.pptx	Fichero MS PowerPoint usado para la defensa del trabajo fin de grado.
./Documentación/Pruebas de código.xlsx	Fichero MS Excel que contiene los resultados obtenidos de las pruebas de código realizadas al sistema.

Tabla 249. Ficheros anexos del proyecto



REFERENCIAS BIBLIOGRÁFICAS

- [1] J. M. Redondo, «Documentos-modelo para Trabajos de Fin de Grado/Master de la Escuela de Informática de Oviedo,» 17 6 2019. [En línea]. Available: https://www.researchgate.net/publication/327882831_Plantilla_de_Proyectos_de_Fin_de_Carrera_de_la_Escuela_de_Informatica_de_Oviedo.
- [2] hsanchez. [En línea]. Available: <https://www.monografias.com/trabajos14/implantacion-datos/implantacion-datos.shtml>. [Último acceso: 07 05 2020].
- [3] T. Netland, «better operations,» [En línea]. Available: <https://better-operations.com/2016/09/15/manufacturing-cost-deployment/>. [Último acceso: 05 05 2020].
- [4] D. H. Knutsen, «Roadmap for Manufacturing Cost Deployment».
- [5] R. O. Matilda Garbe, «Evaluation of Manufacturing Cost Models,» 2014.
- [6] H. & K. T. Yamashina, Manufacturing cost deployment, 2002.
- [7] L. Gracia, «¿Qué es AKS (Azuree Kubernetes Service)?,» 26 07 2018. [En línea]. Available: <https://unpocodejava.com/2018/07/26/que-es-aks-azure-kubernetes-service/>. [Último acceso: 28 10 2019].
- [8] Sophia, «Angular vs React vs Vue: Which is the Best Choice for 2019?,» [En línea]. Available: <https://hackernoon.com/angular-vs-react-vs-vue-which-is-the-best-choice-for-2019-16ce0deb3847>. [Último acceso: 11 10 2019].
- [9] Microsoft, «Implementación de un bus de eventos con RabbitMQ para el entorno de desarrollo o de prueba,» 14 02 2020. [En línea]. Available: <https://docs.microsoft.com/es-es/dotnet/architecture/microservices/multi-container-microservice-net-applications/rabbitmq-event-bus-development-test-environment>.
- [10] Fortytwo, «Two-Factor Authentication REST API,» [En línea]. Available: <https://www.fortytwo.com/apis/two-factor-authentication/>. [Último acceso: 28 10 2019].
- [11] I. Landajuela, «MVVMBasicDemo1: Patrón de diseño MVVM con aplicaciones basadas en WPF Resultado de imagen de WPF (Windows Presentation Foundation),» 26 07 2017. [En línea]. Available: <https://ikerlandajuela.wordpress.com/2017/07/26/mvmbasicdemo1-patron-de-diseno-mvvm-con-aplicaciones-basadas-en-wpf-resultado-de-imagen-de-wpf-windows-presentation-foundation/>. [Último acceso: 28 10 2019].
- [12] Cucumber, «Gherkin syntax,» [En línea]. Available: <https://cucumber.io/docs/gherkin/>.



- [13] H. Montero, «No solo usabilidad: revista sobre personas, diseño y tecnología,» 30 3 2003. [En línea]. Available: <http://www.nosolousabilidad.com/articulos/heuristica.htm>.
- [14] M. Docs, «Implementación del nivel de aplicación de microservicios mediante la API web,» 30 01 2020. [En línea]. Available: <https://docs.microsoft.com/es-es/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/microservice-application-layer-implementation-web-api>. [Último acceso: 18 02 2020].
- [15] F. Arciniega, «Normas y Estándares de calidad para el desarrollo de Software,» [En línea]. Available: <https://fernandoarciniega.com/normas-y-estandares-de-calidad-para-el-desarrollo-de-software/>. [Último acceso: 16 04 2020].
- [16] M. Docs, «Convenciones de código de C# (Guía de programación de C#),» 20 07 2015. [En línea]. Available: <https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>.
- [17] F. Herrera, «Udemy,» [En línea]. Available: <https://www.udemy.com/course/angular-2-fernando-herrera/>. [Último acceso: 5 11 2019].
- [18] Microsoft, «Microservicios de .NET: Arquitectura para aplicaciones .NET en contenedor,» [En línea]. Available: <https://docs.microsoft.com/es-es/dotnet/architecture/microservices/>. [Último acceso: 22 04 2020].
- [19] Microsoft, «DDOS Protection Overview,» [En línea]. Available: <https://docs.microsoft.com/es-es/azure/virtual-network/ddos-protection-overview>. [Último acceso: 09 05 2020].
- [20] I. Llana, J. M. Redondo y L. Vinuesa, «Towards Lightweight Mobile Pentesting Tools to Quickly Assess Machine Security Levels,» *IEEE Latin American Transactions*, p. pp, 2019.
- [21] J. M. Redondo, *Introducción Práctica a la Administración Segura de Servidores Apache Bajo Linux*, Oviedo, Asturias: Servicio de Publicaciones de la Universidad de Oviedo, 2019.



GNU FREE DOCUMENTATION LICENSE

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not



explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.



The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.



It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.*
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.*
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.*
- D. Preserve all the copyright notices of the Document.*
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.*
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.*
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.*
- H. Include an unaltered copy of this License.*
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.*
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.*



- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.*
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.*
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.*
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.*
- O. Preserve any Warranty Disclaimers.*

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique



number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those



notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <https://www.gnu.org/licenses/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.



11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C) YEAR YOUR NAME.  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.3  
or any later version published by the Free Software Foundation;  
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.  
A copy of the license is included in the section entitled "GNU  
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the  
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.





ANEXOS

ANEXO I: PRESUPUESTO

El proyecto se ha dividido en seis capítulos cuya relación se corresponde con cada fase de desarrollo del software y una adicional correspondiente a las licencias software necesarias.

Resumen por capítulos		
Capítulo	Descripción	Total importe
01	Planificación inicial	3.376,41 €
02	Análisis del sistema de información	5.278,05 €
03	Diseño del sistema de información	2.124,68 €
04	Construcción del sistema de información	9.077,47 €
05	Implantación y aceptación del sistema	776,81 €
06	Licencias	7.094,28 €
TOTAL PRESUPUESTO		27.727,69 €

Tabla 250. Presupuesto resumen del cliente

El citado presupuesto se desarrolla mediante seis capítulos compuesto cada uno de ellos por varias partidas numeradas ordinalmente de forma creciente.

Presupuesto de cliente					
Capítulo	Item	Descripción	Hr.	Importe	Total
01		Planificación inicial	65,00		3.376,41 €
	01	Adquisición del dominio	8,00	459,55 €	
	02	Inicio de Plan de Sistemas de Información	20,00	1.095,99 €	
	03	Definición de la organización	8,00	463,39 €	
	04	Especificación del Ámbito y Alcance	8,00	434,55 €	
	05	Estudio de la Arquitectura Tecnológica	11,00	538,38 €	
	06	Estudio y valoración de Alternativas	10,00	384,56 €	
02		Análisis del sistema de información	109,00		5.278,05 €
	01	Definición del sistema	4,00	211,51 €	
	02	Análisis de requisitos	25,00	1.201,74 €	
	03	Análisis de casos de uso	20,00	961,39 €	
	04	Análisis de Clases	15,00	769,11 €	
	05	Identificación de Subsistemas de Análisis	10,00	490,31 €	
	06	Definición de Interfaces de usuario	5,00	163,44 €	
	07	Especificación del Plan de Pruebas	10,00	509,54 €	
	08	Definición del presupuesto	20,00	971,01 €	
03		Diseño del sistema de información	40,00		2.124,68 €
	01	Diseño físico de datos	5,00	288,42 €	
	02	Diseño de los microservicios	20,00	1.067,15 €	



	03	Especificación Técnica del Plan de Pruebas	15,00	769,11 €	
04		Construcción del sistema de información	272,00		9.077,47 €
	001	<i>Back-end startup</i>	30,00	903,71 €	
	02	<i>Front-end startup</i>	20,00	653,75 €	
	03	Desarrollo de la funcionalidad	175,00	6.164,45 €	
	04	Pruebas	47,00	1.355,56 €	
05		Implantación y aceptación del sistema	14,50		776,81 €
	01	Elaboración de Manuales	6,50	336,49 €	
	02	Pruebas de Implantación del Sistema	3,00	173,05 €	
	003	Validación del software	5,00	267,27 €	
06		Licencias			7.094,28 €
	01	Licencias software		7.094,28 €	
TOTAL PRESUPUESTO					27.727,69 €

Tabla 251. Presupuesto del cliente

“Asciende el presente presupuesto líquido a la expresa cantidad de veintisiete mil setecientos veintisiete con sesenta y nueve euros (27.727,69 €)”

El citado importe se verá incrementado con el IVA vigente en momento de la facturación.

Se puede consultar toda la información relativa al **Presupuesto de Costes** en el archivo anexo externo llamado **“Presupuesto”** – en formato Excel y PDF.



ANEXO II: REQUISITOS FUNCIONALES

Este anexo viene complementado con el fichero incluido como anexo externo “**Fórmulas**” que incluye los cálculos pertinentes a la lógica de los componentes de un FactoryModel.

Inicio de sesión

- RF.LG.2.** El sistema permitirá iniciar sesión a los usuarios registrados previamente en el sistema.
 - RF.LG.2.1.** Se permitirá acceder de forma anónima.
 - RF.LG.2.2.** Se dispondrá de un formulario en el que introducir los datos.
 - RF.LG.2.2.1.** El formulario deberá solicitar al usuario su correo electrónico.
 - RF.LG.2.2.1.1.** No debe aceptarse un valor vacío.
 - RF.LG.2.2.1.2.** El formato que debe de seguir es de un email.
 - RF.LG.2.2.1.3.** Se devolverá un mensaje de error si no se cumple con el formato.
 - RF.LG.2.2.2.** El formulario deberá solicitar al usuario su contraseña.
 - RF.LG.2.2.2.1.** No debe de aceptarse un valor vacío.
 - RF.LG.2.2.2.2.** La contraseña deberá ocultarse.
 - RF.LG.2.2.2.3.** Se devolverá un mensaje de error si no se cumple con el formato.
 - RF.LG.2.2.3.** El formulario deberá solicitar si el usuario quiere recordar su contraseña.
 - RF.LG.2.2.3.1.** Por defecto, tomará el valor desactivado.
 - RF.LG.2.2.3.2.** La duración de la sesión será de X segundos desde el inicio de sesión.
 - RF.LG.2.3.** El sistema deberá comprobar que el usuario está dado de alta.
 - RF.LG.2.3.1.** El sistema comprobará si el email existe en el sistema.
 - RF.LG.2.3.2.** El sistema comprobará si el acceso al usuario ha sido revocado.
 - RF.LG.2.3.3.** El sistema comprobará si el usuario y la contraseña se corresponden.
 - RF.LG.2.3.4.** Se devolverá un mensaje de error si el sistema no consigue iniciar sesión de forma satisfactoria.

Gestión de usuarios

Dar de alta a usuarios

- RF.US.1.** El sistema permitirá dar de alta a nuevos usuarios en el sistema.
 - RF.US.1.1.** Se permitirá acceder a un usuario con un rol de Administrador.
 - RF.US.1.2.** Se dispondrá de un formulario que permita introducir los datos.
 - RF.US.1.2.1.** El formulario deberá de solicitar el correo electrónico.
 - RF.US.1.2.1.1.** No se aceptará un valor vacío.
 - RF.US.1.2.1.2.** El formato que debe de ser de un email.
 - RF.US.1.2.1.3.** No puede existir otro usuario con el mismo correo electrónico.



- RF.US.1.2.1.4.** Se devolverá un mensaje de error si no se cumple con el formato.
- RF.US.1.2.2.** El formulario deberá de solicitar la contraseña al usuario.
 - RF.US.1.2.2.1.** No debe aceptarse un valor vacío.
 - RF.US.1.2.2.2.** No debe de aceptarse valores inferiores a X caracteres.
 - RF.US.1.2.2.3.** Se devolverá un mensaje de error si no se cumple con el formato.
- RF.US.1.2.3.** El formulario deberá de solicitar la confirmación de contraseña al usuario.
 - RF.US.1.2.3.1.** No debe de aceptarse un valor vacío.
 - RF.US.1.2.3.2.** Debe de ser exactamente igual al campo contraseña.
 - RF.US.1.2.3.3.** Se devolverá un mensaje de error si no se cumple con el formato.
- RF.US.1.2.4.** El formulario deberá de solicitar el nombre del usuario.
 - RF.US.1.2.4.1.** No debe aceptarse un valor vacío.
 - RF.US.1.2.4.2.** Se devolverá un mensaje de error si no se cumple con el formato.
- RF.US.1.2.5.** El formulario deberá de solicitar los apellidos del usuario.
 - RF.US.1.2.5.1.** No debe aceptarse un valor vacío.
 - RF.US.1.2.5.2.** Se devolverá un mensaje de error si no se cumple con el formato.
- RF.US.1.2.6.** El formulario deberá de solicitar los roles del usuario.
 - RF.US.1.2.6.1.** Un usuario debe de disponer de X roles.
 - RF.US.1.2.6.2.** Un usuario puede pertenecer a todos los roles.
 - RF.US.1.2.6.3.** Se devolverá un mensaje de error si no se escoge al menos un rol.
- RF.US.1.2.7.** El formulario deberá de permitir cancelar la operación.
- RF.US.1.2.8.** El formulario deberá de permitir finalizar el alta del usuario.
 - RF.US.1.2.8.1.** Se devolverá un mensaje si se ha producido un error en la creación.

Consultar lista usuarios

- RF.US.2.** El sistema permitirá consultar los diferentes usuarios dados de alta en el sistema.
 - RF.US.2.1.** Se permitirá acceder a un usuario con un rol de Administrador.
 - RF.US.2.2.** El sistema nos proporcionará información de los usuarios dados de alta con los siguientes campos.
 - RF.US.2.2.1.** Correo electrónico.
 - RF.US.2.2.2.** Nombre.
 - RF.US.2.2.3.** Apellidos.
 - RF.US.2.2.4.** Usuario revocado.
 - RF.US.2.2.5.** Fecha de modificación del usuario.
 - RF.US.2.3.** El sistema nos permitirá filtrar por los campos anteriormente descritos.
 - RF.US.2.4.** El sistema nos permitirá ordenar por los campos anteriormente descritos.

Modificar usuarios

- RF.US.3.** El sistema permitirá modificar datos de los usuarios en el sistema.
 - RF.US.3.1.** Se permitirá acceder a un usuario con un rol de Administrador.
 - RF.US.3.2.** Se dispondrá de un formulario que permita introducir los datos.
 - RF.US.3.2.1.** El formulario deberá mostrar el correo electrónico.
 - RF.US.3.2.1.1.** No debe permitirse modificarse.
 - RF.US.3.2.2.** El formulario permitirá modificar el nombre del usuario.



- RF.US.3.2.2.1.** No debe aceptarse un valor vacío.
- RF.US.3.2.2.2.** Se devolverá un mensaje de error si no se cumple con el formato.
- RF.US.3.2.3.** El formulario permitirá modificar los apellidos del usuario.
 - RF.US.3.2.3.1.** No debe aceptarse un valor vacío.
 - RF.US.3.2.3.2.** Se devolverá un mensaje de error si no se cumple con el formato.
- RF.US.3.2.4.** El formulario permitirá cambiar la revocación a un usuario.
- RF.US.3.2.5.** El formulario permitirá modificar los roles del usuario.
 - RF.US.3.2.5.1.** Un usuario debe de disponer de X roles.
 - RF.US.3.2.5.2.** Un usuario puede pertenecer a todos los roles.
 - RF.US.3.2.5.3.** Se devolverá un mensaje de error si no se cumple con los requisitos.
- RF.US.3.2.6.** El formulario deberá de permitir cancelar la operación.
- RF.US.3.2.7.** El formulario deberá de permitir finalizar la modificación del usuario.
 - RF.US.3.2.7.1.** Se devolverá un mensaje si se ha producido un error en la modificación.

Consultar detalles de un usuario

- RF.US.4.** El sistema permitirá consultar la información en detalle de un usuario.
 - RF.US.4.1.** Se permitirá acceder a un usuario que ha iniciado sesión.
 - RF.US.4.2.** El sistema nos proporcionará información del usuario dado de alta.
 - RF.US.4.2.1.** Correo electrónico.
 - RF.US.4.2.2.** Nombre.
 - RF.US.4.2.3.** Apellidos.
 - RF.US.4.2.4.** Usuario revocado.
 - RF.US.4.2.5.** Fecha de modificación del usuario.
 - RF.US.4.3.** El sistema dará la información en formato solo lectura.

Gestión de visiones

Consultar visiones

- RF.VS.2.** El sistema permitirá consultar las diferentes visiones dados de alta en el sistema.
 - RF.VS.2.1.** Se permitirá acceder a un usuario que ha iniciado sesión.
 - RF.VS.2.2.** El sistema nos proporcionará información de las visiones con los siguientes campos.
 - RF.VS.2.2.1.** Nombre
 - RF.VS.2.2.2.** Departamento.
 - RF.VS.2.2.3.** Área.
 - RF.VS.2.2.4.** Ahorros potenciales
 - RF.VS.2.2.5.** Fecha de modificación de la visión.
 - RF.VS.2.3.** El sistema nos permitirá filtrar por los campos anteriormente descritos.
 - RF.VS.2.4.** El sistema nos permitirá ordenar por los campos anteriormente descritos.



Añadir visiones

RF.VS.3. El sistema permitirá dar de alta nuevas visiones.

RF.VS.3.1. Se permitirá acceder a un usuario que ha iniciado sesión con un rol con permisos para modificar.

RF.VS.3.2. Se dispondrá de un formulario que permita introducir los datos.

RF.VS.3.2.1. El formulario deberá de solicitar el nombre.

RF.VS.3.2.1.1. No debe aceptarse un valor vacío.

RF.VS.3.2.1.2. El nombre deberá ser único.

RF.VS.3.2.1.3. Se devolverá un mensaje de error si no se cumple con el formato.

RF.VS.3.2.2. El formulario deberá de solicitar la descripción.

RF.VS.3.2.3. El formulario deberá de solicitar el departamento.

RF.VS.3.2.3.1. No debe aceptarse un valor vacío.

RF.VS.3.2.3.2. Se devolverá un mensaje de error si no se cumple con el formato.

RF.VS.3.2.4. El formulario deberá de solicitar el área.

RF.VS.3.2.4.1. No debe aceptarse un valor vacío.

RF.VS.3.2.4.2. Se devolverá un mensaje de error si no se cumple con el formato.

RF.VS.3.2.5. El formulario deberá de solicitar la localización.

RF.VS.3.2.5.1. No debe aceptarse un valor vacío.

RF.VS.3.2.5.2. Se devolverá un mensaje de error si no se cumple con el formato.

RF.VS.3.2.6. El formulario deberá de solicitar la dirección.

RF.VS.3.2.6.1. No debe aceptarse un valor vacío.

RF.VS.3.2.6.2. Se devolverá un mensaje de error si no se cumple con el formato.

RF.VS.3.2.7. El formulario deberá de permitir cancelar la operación.

RF.VS.3.2.8. El formulario deberá de permitir finalizar el alta de la visión.

RF.VS.3.2.8.1. Se devolverá un mensaje si se ha producido un error en la creación.

RF.VS.3.3. Se mostrará un mensaje en caso de:

RF.VS.3.3.1. Si se produce un error, el sistema mostrará un mensaje advirtiendo que no se ha podido completar con éxito.

RF.VS.3.3.2. Si la operación se realiza con éxito, el sistema notificará que la operación ha sido realizada con éxito.

Modificar visiones

RF.VS.4. El sistema permitirá modificar datos de las visiones del sistema.

RF.VS.4.1. Se permitirá acceder a un usuario que ha iniciado sesión con un rol con permisos para modificar.

RF.VS.4.2. Se dispondrá de un formulario que permita introducir los datos.

RF.VS.4.2.1. El formulario deberá de solicitar el nombre.

RF.VS.4.2.1.1. No debe aceptarse un valor vacío.

RF.VS.4.2.1.2. El nombre deberá ser único.

RF.VS.4.2.1.3. Se devolverá un mensaje de error si no se cumple con el formato.

RF.VS.4.2.2. El formulario deberá de solicitar la descripción.

RF.VS.4.2.3. El formulario deberá de solicitar el departamento.



- RF.VS.4.2.3.1.** No debe aceptarse un valor vacío.
- RF.VS.4.2.3.2.** Se devolverá un mensaje de error si no se cumple con el formato.
- RF.VS.4.2.4.** El formulario deberá de solicitar el área.
- RF.VS.4.2.4.1.** No debe aceptarse un valor vacío.
- RF.VS.4.2.4.2.** Se devolverá un mensaje de error si no se cumple con el formato.
- RF.VS.4.2.5.** El formulario deberá de solicitar la localización.
- RF.VS.4.2.5.1.** No debe aceptarse un valor vacío.
- RF.VS.4.2.5.2.** Se devolverá un mensaje de error si no se cumple con el formato.
- RF.VS.4.2.6.** El formulario deberá de solicitar la dirección.
- RF.VS.4.2.6.1.** No debe aceptarse un valor vacío.
- RF.VS.4.2.6.2.** Se devolverá un mensaje de error si no se cumple con el formato.
- RF.VS.4.2.7.** El formulario deberá de permitir cancelar la operación.
- RF.VS.4.2.8.** El formulario deberá de permitir finalizar el alta de la visión.
 - RF.VS.4.2.8.1.** Se devolverá un mensaje si se ha producido un error en la modificación.
- RF.VS.4.3.** Se mostrará un mensaje en caso de:
 - RF.VS.4.3.1.** Si se produce un error, el sistema mostrará un mensaje advirtiendo que no se ha podido completar con éxito.
 - RF.VS.4.3.2.** Si la operación se realiza con éxito, el sistema notificará que la operación ha sido realizada con éxito.

Clonar visiones

- RF.VS.5.** El sistema permitirá dar clonar visiones ya existentes.
 - RF.VS.5.1.** Se permitirá acceder a un usuario que ha iniciado sesión con un rol con permisos para modificar.
 - RF.VS.5.2.** Se dispondrá de un formulario que permita introducir los datos.
 - RF.VS.5.2.1.** El formulario deberá de solicitar el nombre.
 - RF.VS.5.2.1.1.** No debe aceptarse un valor vacío.
 - RF.VS.5.2.1.2.** El nombre deberá ser único.
 - RF.VS.5.2.1.3.** El nombre por defecto será el nombre del original.
 - RF.VS.5.2.1.4.** Se devolverá un mensaje de error si no se cumple con el formato.
 - RF.VS.5.2.2.** El formulario deberá de permitir cancelar la operación.
 - RF.VS.5.2.3.** El formulario deberá de permitir finalizar la clonación de la visión.
 - RF.VS.5.2.3.1.** Se devolverá un mensaje si se ha producido un error en la clonación.
 - RF.VS.5.3.** Se mostrará un mensaje en caso de:
 - RF.VS.5.3.1.** Si se produce un error, el sistema mostrará un mensaje advirtiendo que no se ha podido completar con éxito.
 - RF.VS.5.3.2.** Si la operación se realiza con éxito, el sistema notificará que la operación ha sido realizada con éxito.

Vista de resumen de visión

- RF.VS.6.** El sistema permitirá consultar las diferentes visiones dados de alta en el sistema.



- RF.VS.6.1.** El sistema permitirá acceder a un usuario que ha iniciado sesión.
- RF.VS.6.2.** El sistema nos proporcionará información de las visiones con los siguientes campos.
 - RF.VS.6.2.1.** Nombre
 - RF.VS.6.2.2.** Departamento.
 - RF.VS.6.2.3.** Área.
 - RF.VS.6.2.4.** Ahorros potenciales
 - RF.VS.6.2.5.** Fecha de modificación de la visión.
- RF.VS.6.3.** El sistema nos permitirá filtrar por los campos anteriormente descritos.
- RF.VS.6.4.** El sistema nos permitirá ordenar por los campos anteriormente descritos.
- RF.VS.7.** El sistema permitirá consultar los FactoryModel publicados de la visión seleccionada.
 - RF.VS.7.1.** El sistema permitirá acceder a un usuario que ha iniciado sesión.
 - RF.VS.7.2.** El sistema nos proporcionará la siguiente información del FactoryModel:
 - RF.VS.7.2.1.** Nombre
 - RF.VS.7.2.2.** Descripción
 - RF.VS.7.2.3.** Ahorros potenciales
 - RF.VS.7.3.** El sistema nos permitirá acceder a una vista más detallada de la información.

Gestión de maquinas

Consultar máquinas

- RF.MQ.5.** El sistema permitirá consultar las diferentes máquinas dados de alta en el sistema de una visión.
 - RF.MQ.5.1.** Se permitirá acceder a un usuario que ha iniciado sesión.
 - RF.MQ.5.2.** El sistema nos proporcionará información de las máquinas dadas de alta con los siguientes campos.
 - RF.MQ.5.2.1.** Nombre.
 - RF.MQ.5.2.2.** Descripción.
 - RF.MQ.5.2.3.** Agrupación.
 - RF.MQ.5.2.4.** Sección.
 - RF.MQ.5.2.5.** Departamento.
 - RF.MQ.5.2.6.** Área.
 - RF.MQ.5.2.7.** Versión.
 - RF.MQ.5.2.8.** Fecha de modificación de la máquina.
 - RF.MQ.5.3.** El sistema nos permitirá filtrar por los campos anteriormente descritos.
 - RF.MQ.5.4.** El sistema nos permitirá ordenar por los campos anteriormente descritos.

Añadir maquinas

- RF.MQ.6.** El sistema permitirá dar de alta nuevas máquinas.
 - RF.MQ.6.1.** Se permitirá acceder a un usuario que ha iniciado sesión con un rol con permisos para modificar.
 - RF.MQ.6.2.** Se dispondrá de un formulario que permita introducir los datos.



RF.MQ.6.2.1. El formulario deberá de solicitar el nombre.

RF.MQ.6.2.1.1. No debe aceptarse un valor vacío.

RF.MQ.6.2.1.2. El nombre deberá ser único dentro de la visión.

RF.MQ.6.2.1.3. Se devolverá un mensaje de error si no se cumple con el formato.

RF.MQ.6.2.2. El formulario deberá de solicitar la descripción.

RF.MQ.6.2.3. El formulario deberá de solicitar la versión.

RF.MQ.6.2.3.1. No debe aceptarse un valor vacío.

RF.MQ.6.2.3.2. Se devolverá un mensaje de error si no se cumple con el formato.

RF.MQ.6.2.4. El formulario deberá de solicitar el grupo al que pertenece.

RF.MQ.6.2.4.1. No debe aceptarse un valor vacío.

RF.MQ.6.2.4.2. Se devolverá un mensaje de error si no se cumple con el formato.

RF.MQ.6.2.5. El formulario deberá de permitir cancelar la operación.

RF.MQ.6.2.6. El formulario deberá de permitir finalizar el alta de la máquina.

RF.MQ.6.2.6.1. Se devolverá un mensaje si se ha producido un error en la creación.

RF.MQ.6.3. Se mostrará un mensaje en caso de:

RF.MQ.6.3.1. Si se produce un error, el sistema mostrará un mensaje advirtiendo que no se ha podido completar con éxito.

RF.MQ.6.3.2. Si la operación se realiza con éxito, el sistema notificará que la operación ha sido realizada con éxito.

Modificar maquinas

RF.MQ.7. El sistema permitirá modificar datos de las maquinas del sistema.

RF.MQ.7.1. Se permitirá acceder a un usuario que ha iniciado sesión con un rol con permisos para modificar.

RF.MQ.7.2. Se dispondrá de un formulario que permita introducir los datos.

RF.MQ.7.2.1. El formulario deberá de solicitar el nombre.

RF.MQ.7.2.1.1. No debe aceptarse un valor vacío.

RF.MQ.7.2.1.2. El nombre deberá ser único dentro de la visión.

RF.MQ.7.2.1.3. Se devolverá un mensaje de error si no se cumple con el formato.

RF.MQ.7.2.2. El formulario deberá de solicitar la descripción.

RF.MQ.7.2.3. El formulario deberá de solicitar la versión.

RF.MQ.7.2.3.1. No debe aceptarse un valor vacío.

RF.MQ.7.2.3.2. Se devolverá un mensaje de error si no se cumple con el formato.

RF.MQ.7.2.4. El formulario deberá de solicitar el grupo al que pertenece.

RF.MQ.7.2.5. No debe aceptarse un valor vacío.

RF.MQ.7.2.5.1. Se devolverá un mensaje de error si no se cumple con el formato.

RF.MQ.7.2.6. El formulario deberá de permitir cancelar la operación.

RF.MQ.7.2.7. El formulario deberá de permitir finalizar el alta de la máquina.

RF.MQ.7.2.7.1. Se devolverá un mensaje si se ha producido un error en la modificación.

RF.MQ.7.3. Se mostrará un mensaje en caso de:

RF.MQ.7.3.1. Si se produce un error, el sistema mostrará un mensaje advirtiendo que no se ha podido completar con éxito.



RF.MQ.7.3.2. Si la operación se realiza con éxito, el sistema notificará que la operación ha sido realizada con éxito.

Clonar máquinas

RF.MQ.8. El sistema permitirá dar clonar maquinas ya existentes.

RF.MQ.8.1. Se permitirá acceder a un usuario que ha iniciado sesión con un rol con permisos para modificar.

RF.MQ.8.2. Se dispondrá de un formulario que permita introducir los datos.

RF.MQ.8.2.1. El formulario deberá de solicitar el nombre.

RF.MQ.8.2.1.1. No debe aceptarse un valor vacío.

RF.MQ.8.2.1.2. El nombre deberá ser único.

RF.MQ.8.2.1.3. El nombre por defecto será el nombre del original.

RF.MQ.8.2.1.4. Se devolverá un mensaje de error si no se cumple con el formato.

RF.MQ.8.2.2. El formulario deberá de solicitar la versión.

RF.MQ.8.2.2.1. No debe aceptarse un valor vacío.

RF.MQ.8.2.2.2. La versión por defecto será la versión del original.

RF.MQ.8.2.2.3. Se devolverá un mensaje de error si no se cumple con el formato.

RF.MQ.8.2.3. El formulario deberá de permitir cancelar la operación.

RF.MQ.8.2.4. El formulario deberá de permitir finalizar la clonación de la visión.

RF.MQ.8.2.4.1. Se devolverá un mensaje si se ha producido un error en la clonación.

RF.MQ.8.3. Se mostrará un mensaje en caso de:

RF.MQ.8.3.1. Si se produce un error, el sistema mostrará un mensaje advirtiendo que no se ha podido completar con éxito.

RF.MQ.8.3.2. Si la operación se realiza con éxito, el sistema notificará que la operación ha sido realizada con éxito.

Gestión de FactoryModels

Consultar FactoryModels

RF.FM.5. El sistema permitirá consultar las diferentes *FactoryModels* dados de alta en el sistema de una MAQUINA.

RF.FM.5.1. Se permitirá acceder a un usuario que ha iniciado sesión.

RF.FM.5.2. El sistema nos proporcionará información de los usuarios dados de alta con los siguientes campos.

RF.FM.5.2.1. Nombre.

RF.FM.5.2.2. Descripción.

RF.FM.5.2.3. Versión.

RF.FM.5.2.4. Estado del FactoryModel.

RF.FM.5.2.5. Ahorros potenciales.



RF.FM.5.2.6. Fecha de modificación del *FactoryModel*.

RF.FM.5.3. El sistema nos permitirá filtrar por los campos anteriormente descritos.

RF.FM.5.4. El sistema nos permitirá ordenar por los campos anteriormente descritos.

Añadir *FactoryModels*

RF.FM.6. El sistema permitirá dar de alta nuevos *FactoryModels*.

RF.FM.6.1. Se permitirá acceder a un usuario que ha iniciado sesión con un rol con permisos para modificar.

RF.FM.6.2. Se dispondrá de un formulario que permita introducir los datos.

RF.FM.6.2.1. El formulario deberá de solicitar el nombre.

RF.FM.6.2.1.1. No debe aceptarse un valor vacío.

RF.FM.6.2.1.2. El nombre deberá ser único dentro de la visión.

RF.FM.6.2.1.3. Se devolverá un mensaje de error si no se cumple con el formato.

RF.FM.6.2.2. El formulario deberá de solicitar la descripción.

RF.FM.6.2.3. El formulario deberá de solicitar la versión.

RF.FM.6.2.3.1. No debe aceptarse un valor vacío.

RF.FM.6.2.3.2. Se devolverá un mensaje de error si no se cumple con el formato.

RF.FM.6.2.4. El formulario deberá de solicitar el grupo al que pertenece.

RF.FM.6.2.4.1. No debe aceptarse un valor vacío.

RF.FM.6.2.4.2. Se devolverá un mensaje de error si no se cumple con el formato.

RF.FM.6.2.5. El formulario deberá de permitir cancelar la operación.

RF.FM.6.2.6. El formulario deberá de permitir finalizar el alta de la máquina.

RF.FM.6.2.6.1. Se mostrará un mensaje en caso de:

RF.FM.6.2.6.1.1 Si se produce un error, el sistema mostrará un mensaje advirtiendo que no se ha podido completar con éxito.

RF.FM.6.2.6.1.2 Si la operación se realiza con éxito, el sistema notificará que la operación ha sido realizada con éxito.

RF.FM.6.3. El *FactoryModel* se añadirá con un estado de *Borrador*.

Modificar *FactoryModels*

RF.FM.7. El sistema permitirá modificar datos de los *FactoryModel* del sistema.

RF.FM.7.1. Se permitirá acceder a un usuario que ha iniciado sesión con un rol con permisos para modificar.

RF.FM.7.2. Se dispondrá de un formulario que permita introducir los datos.

RF.FM.7.2.1.1. El formulario deberá de solicitar el nombre.

RF.FM.7.2.1.1.1 No debe aceptarse un valor vacío.

RF.FM.7.2.1.1.2 El nombre deberá ser único dentro de la visión.

RF.FM.7.2.1.1.3 Se devolverá un mensaje de error si no se cumple con el formato.

RF.FM.7.2.1.2. El formulario deberá de solicitar la descripción.

RF.FM.7.2.1.3. El formulario deberá de solicitar la versión.



- RF.FM.7.2.1.3.1** No debe aceptarse un valor vacío.
- RF.FM.7.2.1.3.2** Se devolverá un mensaje de error si no se cumple con el formato.
- RF.FM.7.2.1.4.** El formulario deberá de permitir cancelar la operación.
- RF.FM.7.2.1.5.** El formulario deberá de permitir finalizar el alta del *FactoryModel*.
 - RF.FM.7.2.1.5.1** Se mostrará un mensaje en caso de:
 - RF.FM.7.2.1.5.1.1.** Si se produce un error, el sistema mostrará un mensaje advirtiendo que no se ha podido completar con éxito.
 - RF.FM.7.2.1.5.1.2.** Si la operación se realiza con éxito, el sistema notificará que la operación ha sido realizada con éxito.

Clonar *FactoryModels*

- RF.FM.8.** El sistema permitirá dar clonar *FactoryModel* ya existentes.
 - RF.FM.8.1.** Se permitirá acceder a un usuario que ha iniciado sesión con un rol con permisos para modificar.
 - RF.FM.8.2.** Se dispondrá de un formulario que permita introducir los datos.
 - RF.FM.8.2.1.** El formulario deberá de solicitar el nombre.
 - RF.FM.8.2.1.1.** No debe aceptarse un valor vacío.
 - RF.FM.8.2.1.2.** El nombre deberá ser único.
 - RF.FM.8.2.1.3.** El nombre por defecto será el nombre del original.
 - RF.FM.8.2.1.4.** Se devolverá un mensaje de error si no se cumple con el formato.
 - RF.FM.8.2.2.** El formulario deberá de solicitar la versión.
 - RF.FM.8.2.2.1.** No debe aceptarse un valor vacío.
 - RF.FM.8.2.2.2.** La versión por defecto será la versión del original.
 - RF.FM.8.2.2.3.** Se devolverá un mensaje de error si no se cumple con el formato.
 - RF.FM.8.2.3.** El formulario deberá de permitir cancelar la operación.
 - RF.FM.8.2.4.** El formulario deberá de permitir finalizar la clonación de la visión.
 - RF.FM.8.3.** Se mostrará un mensaje en caso de:
 - RF.FM.8.3.1.** Si se produce un error, el sistema mostrará un mensaje advirtiendo que no se ha podido completar con éxito.
 - RF.FM.8.3.2.** Si la operación se realiza con éxito, el sistema notificará que la operación ha sido realizada con éxito.

Publicar *FactoryModels*

- RF.FM.9.** El sistema permitirá publicar un *FactoryModel*.
 - RF.FM.9.1.** Estará restringidos para los usuarios con un rol con permisos de escritura.
 - RF.FM.9.2.** El *FactoryModel* pasará al siguiente estado:
 - RF.FM.9.2.1.** Se mantendrán todas las propiedades intactas exceptuando el estado. Este pasará al Estado final Publicado.
 - RF.FM.9.2.2.** Solo puede existir un *FactoryModel* por cada máquina.



RF.FM.9.2.3. Un FactoryModel publicado será de solo lectura, al igual que todos sus componentes.

RF.FM.9.2.4. Esta acción solo se podrá realizar una vez.

RF.FM.9.3. Se mostrará un mensaje en caso de:

RF.FM.9.3.1. Si se produce un error, el sistema mostrará un mensaje advirtiendo que no se ha podido completar con éxito.

RF.FM.9.3.2. Si la operación se realiza con éxito, el sistema notificará que la operación ha sido realizada con éxito.

Gestión de los componentes de un FactoryModels

Seleccionar los componentes disponibles en un FactoryModel

RF.FMC.3. El sistema permitirá seleccionar los componentes activados en el sistema.

RF.FMC.3.1. Se permitirá modificar a un usuario que ha iniciado sesión con un rol con permisos para modificar.

RF.FMC.3.2. El sistema nos proporcionará información de los componentes que se encuentran disponibles en el sistema a través del *Service Discovery*.

RF.FMC.3.3. Se podrá modificar el estado de un componente.

RF.FMC.3.3.1. El usuario podrá activar un componente.

RF.FMC.3.3.2. El usuario podrá desactivar un componente.

RF.FMC.3.4. Se guardará el estado una vez el usuario ha decidido guardar los cambios.

RF.FMC.3.4.1. Si el guardado no se completa con éxito, se mostrará un mensaje de error.

Consultar los componentes disponibles en un FactoryModel

RF.FMC.4. El sistema permitirá consultar los componentes activados en el sistema.

RF.FMC.4.1. Se permitirá acceder a un usuario que ha iniciado sesión.

RF.FMC.4.2. El sistema nos proporcionará información de los componentes que se encuentran disponibles en el sistema a través del *Service Discovery*.

RF.FMC.4.3. Se mostrará haciendo uso de elementos gráficos la información por componentes.

RF.FMC.4.3.1. Deberá proporcionar el nombre del componente.

RF.FMC.4.3.2. Deberá proporcionar los ahorros potenciales.

RF.FMC.4.4. El sistema proporcionará un resumen del componente:

RF.FMC.4.4.1. El sistema proporcionará la siguiente información del componente:

RF.FMC.4.4.1.1. Nombre.

RF.FMC.4.4.1.2. Ahorros potenciales.

RF.FMC.4.4.1.3. Se diferenciará un valor positivo y uno negativo.

RF.FMC.4.4.1.4. Estado actual del servicio desplegado.



- RF.FMC.4.4.1.5.** Estado del componente.
- RF.FMC.4.4.2.** El sistema estará preparado para los siguientes componentes:
 - RF.FMC.4.4.2.1.** Costes
 - RF.FMC.4.4.2.2.** Energía
 - RF.FMC.4.4.2.3.** Productividad
 - RF.FMC.4.4.2.4.** Calidad
- RF.FMC.4.4.3.** El sistema permitirá activar y desactivar un componente.
 - RF.FMC.4.4.3.1.** En caso de existir dependencias, se activará o desactivará el componente padre de forma adecuada.
 - RF.FMC.4.4.3.1.1** Se mostrará un mensaje de advertencia.
- RF.FMC.4.4.4.** El sistema permitirá acceder a la vista en detalles y edición de un componente.
- RF.FMC.4.5.** Se guardará el estado actual una vez el usuario ha concluido de realizar las operaciones.

Componente de Costes

Vista de detalles y edición

- RF.CS.2.** El sistema permitirá consultar los siguientes elementos:
 - RF.CS.2.1.** Todos los valores deben de disponer un valor de referencia y un valor de referencia ajustada.
 - RF.CS.2.2.** El sistema permitirá modificar los valores si no está publicado el FactoryModel.
 - RF.CS.2.3.** Se dispondrá de los siguientes elementos:
 - RF.CS.2.3.1.** Coste de mano de obra.
 - RF.CS.2.3.1.1.** El valor por defecto será de X.
 - RF.CS.2.3.2.** Número de trabajadores.
 - RF.CS.2.3.3.** Horas anuales de mano de obra.
 - RF.CS.2.3.4.** Producción.
 - RF.CS.2.3.5.** Potenciales ganancias.
 - RF.CS.2.3.5.1.** Campo calculado por el servidor.
 - RF.CS.2.3.6.** Costes variables.
 - RF.CS.2.3.6.1.** Deberá tener un aspecto similar a una hoja de cálculos.
 - RF.CS.2.3.6.1.1** Dispondrá de un sistema de rehacer.
 - RF.CS.2.3.6.1.2** Dispondrá de un sistema de deshacer.
 - RF.CS.2.3.6.2.** Todo dato debe de ser validado.
 - RF.CS.2.3.6.3.** Se dispondrá de los siguientes campos:
 - RF.CS.2.3.6.3.1** Naturaleza.
 - RF.CS.2.3.6.3.1.1.** Campo introducido por el usuario.
 - RF.CS.2.3.6.3.2** Familia.
 - RF.CS.2.3.6.3.2.1.** Campo introducido por el usuario.

- RF.CS.2.3.6.3.3** Unidades.
 - RF.CS.2.3.6.3.3.1.** Campo introducido por el usuario.
 - RF.CS.2.3.6.3.3.2.** Número entero positivo.
- RF.CS.2.3.6.3.4** Precio/unidad
 - RF.CS.2.3.6.3.4.1.** Campo introducido por el usuario.
 - RF.CS.2.3.6.3.4.2.** Número decimal con dos decimales.
- RF.CS.2.3.6.3.5** Total/Unidad
 - RF.CS.2.3.6.3.5.1.** Campo calculado.
 - RF.CS.2.3.6.3.5.2.** Número decimal con dos decimales.
- RF.CS.2.3.6.3.6** Total Producción.
 - RF.CS.2.3.6.3.6.1.** Campo calculado.
 - RF.CS.2.3.6.3.6.2.** Número decimal con dos decimales.
- RF.CS.2.3.6.3.7** Inflación.
- RF.CS.2.3.6.3.8** Sostenibilidad.
- RF.CS.2.3.6.3.9** Referencia Precio/Unidad.
 - RF.CS.2.3.6.3.9.1.** Campo calculado haciendo uso del valor de Precio/Unidad en caso de ser 0.
 - RF.CS.2.3.6.3.9.2.** Podrá ser introducido por el usuario.
 - RF.CS.2.3.6.3.9.3.** Número decimal con dos decimales.
- RF.CS.2.3.6.3.10** Referencia Total/Unidad.
 - RF.CS.2.3.6.3.10.1.** Campo calculado.
 - RF.CS.2.3.6.3.10.2.** Número decimal con dos decimales.
- RF.CS.2.3.6.3.11** Referencia Total Producción.
 - RF.CS.2.3.6.3.11.1.** Campo calculado.
 - RF.CS.2.3.6.3.11.2.** Número decimal con dos decimales.
- RF.CS.2.3.6.4.** Permitirá añadir una nueva fila.
- RF.CS.2.3.6.5.** Permitirá eliminar una fila.
- RF.CS.2.3.6.6.** Se mostrarán como mínimo X filas.
- RF.CS.2.3.6.7.** Se podrá descargar los datos de la tabla en un formato compatible con Excel.
- RF.CS.2.3.7.** Costes fijos.
 - RF.CS.2.3.7.1.** Deberá tener un aspecto similar a una hoja de cálculos.
 - RF.CS.2.3.7.1.1** Dispondrá de un sistema de rehacer.
 - RF.CS.2.3.7.1.2** Dispondrá de un sistema de deshacer.
 - RF.CS.2.3.7.2.** Todo dato debe de ser validado.
 - RF.CS.2.3.7.3.** Se dispondrá de los siguientes campos:
 - RF.CS.2.3.7.3.1** Naturaleza.
 - RF.CS.2.3.7.3.1.1.** Campo introducido por el usuario.
 - RF.CS.2.3.7.3.2** Familia.
 - RF.CS.2.3.7.3.2.1.** Campo introducido por el usuario.
 - RF.CS.2.3.7.3.3** Unidades.
 - RF.CS.2.3.7.3.4** Precio/unidad
 - RF.CS.2.3.7.3.5** Total/Unidad
 - RF.CS.2.3.7.3.5.1.** Campo calculado.



- RF.CS.2.3.7.3.5.2.** Número decimal con dos decimales.
- RF.CS.2.3.7.3.6** Total Producción.
 - RF.CS.2.3.7.3.6.1.** Campo introducido por el usuario.
 - RF.CS.2.3.7.3.6.2.** Número decimal con dos decimales.
- RF.CS.2.3.7.3.7** Inflación.
 - RF.CS.2.3.7.3.7.1.** Campo introducido por el usuario.
 - RF.CS.2.3.7.3.7.2.** Número porcentual con dos decimales.
- RF.CS.2.3.7.3.8** Sostenibilidad.
 - RF.CS.2.3.7.3.8.1.** Campo introducido por el usuario.
 - RF.CS.2.3.7.3.8.2.** Número porcentual con dos decimales.
- RF.CS.2.3.7.3.9** Referencia Precio/Unidad.
 - RF.CS.2.3.7.3.9.1.** Campo calculado haciendo uso del valor de Precio/Unidad en caso de ser 0.
 - RF.CS.2.3.7.3.9.2.** Podrá ser introducido por el usuario.
 - RF.CS.2.3.7.3.9.3.** Número decimal con dos decimales.
- RF.CS.2.3.7.3.10** Referencia Total/Unidad.
 - RF.CS.2.3.7.3.10.1.** Campo calculado.
 - RF.CS.2.3.7.3.10.2.** Número decimal con dos decimales.
- RF.CS.2.3.7.3.11** Referencia Total Producción.
 - RF.CS.2.3.7.3.11.1.** Campo calculado.
 - RF.CS.2.3.7.3.11.2.** Número decimal con dos decimales.
- RF.CS.2.3.7.4.** Permitirá añadir una nueva fila.
- RF.CS.2.3.7.5.** Permitirá eliminar una fila.
- RF.CS.2.3.7.6.** Se mostrarán como mínimo X filas.
- RF.CS.2.3.7.7.** Se podrá descargar los datos de la tabla en un formato compatible con Excel.
- RF.CS.2.3.8.** Comentario.
 - RF.CS.2.3.8.1.** Por defecto, será vacío.
 - RF.CS.2.3.8.2.** Podrá ser introducido por el usuario.
- RF.CS.2.4.** El sistema debe de permitir recargar los cálculos.
 - RF.CS.2.4.1.** Si se realiza de forma correcta, el sistema actualizará los datos por los actuales.
 - RF.CS.2.4.2.** En caso de fallar, se mostrará un mensaje de error.
- RF.CS.2.5.** El sistema debe de permitir guardar los cambios.
 - RF.CS.2.5.1.** Se guardarán los siguientes elementos:
 - RF.CS.2.5.1.1.** Coste de mano de obra.
 - RF.CS.2.5.1.2.** Número de trabajadores.
 - RF.CS.2.5.1.3.** Horas anuales de mano de obra.
 - RF.CS.2.5.1.4.** Producción.
 - RF.CS.2.5.1.5.** Costes variables.
 - RF.CS.2.5.1.6.** Costes fijos.
 - RF.CS.2.5.1.7.** Comentario.
 - RF.CS.2.5.2.** El sistema debe permitir realizar la petición de guardado de datos.
 - RF.CS.2.5.2.1.** Se deberá de actualizar la información contenida en la tabla anteriormente.



- RF.CS.2.5.2.2.** Se procederá a realizar el guardado.
 - RF.CS.2.5.2.2.1** Si se realiza de forma correcta, el sistema nos informará.
 - RF.CS.2.5.2.2.2** En caso de fallar, se mostrará un error.

Componente de Energía

Vista de detalles y edición.

- RF.ES.1.** El sistema permitirá consultar los siguientes elementos:
 - RF.ES.1.1.** El sistema permitirá modificar los valores si no está publicado el FactoryModel.
 - RF.ES.1.2.** Deberá de disponer de un formato parecido a una hoja de cálculo.
 - RF.ES.1.3.** Se dispondrá de los siguientes elementos:
 - RF.ES.1.3.1.** Resumen de cálculos.
 - RF.ES.1.3.1.1.** Potencia contratada
 - RF.ES.1.3.1.1.1** Campo calculado.
 - RF.ES.1.3.1.1.2** Número decimal con dos decimales.
 - RF.ES.1.3.1.1.3** Dispondrá de campo en referencia y referencia ajustada.
 - RF.ES.1.3.1.2.** Consumo anual.
 - RF.ES.1.3.1.2.1** Campo calculado.
 - RF.ES.1.3.1.2.2** Número decimal con dos decimales.
 - RF.ES.1.3.1.2.3** Dispondrá de campo en referencia y referencia ajustada.
 - RF.ES.1.3.1.3.** Total
 - RF.ES.1.3.1.3.1** Campo calculado.
 - RF.ES.1.3.1.3.2** Número decimal con dos decimales.
 - RF.ES.1.3.1.3.3** Dispondrá de campo en referencia y referencia ajustada.
 - RF.ES.1.3.1.4.** Ahorro de potencia contratada
 - RF.ES.1.3.1.4.1** En unidades monetarias
 - RF.ES.1.3.1.4.1.1.** Campo calculado.
 - RF.ES.1.3.1.4.1.2.** Número decimal con dos decimales.
 - RF.ES.1.3.1.4.2** En porcentaje de incremento.
 - RF.ES.1.3.1.4.2.1.** Campo calculado.
 - RF.ES.1.3.1.4.2.2.** Número decimal con dos decimales.
 - RF.ES.1.3.1.5.** Ahorro de consumo
 - RF.ES.1.3.1.5.1** En unidades monetarias
 - RF.ES.1.3.1.5.1.1.** Campo calculado.
 - RF.ES.1.3.1.5.1.2.** Número decimal con dos decimales.
 - RF.ES.1.3.1.5.2** En porcentaje de incremento.
 - RF.ES.1.3.1.5.2.1.** Campo calculado.
 - RF.ES.1.3.1.5.2.2.** Número decimal con dos decimales.
 - RF.ES.1.3.1.6.** Ahorro total
 - RF.ES.1.3.1.6.1** En unidades monetarias



- RF.ES.1.3.1.6.1.1.** Campo calculado.
- RF.ES.1.3.1.6.1.2.** Número decimal con dos decimales.
- RF.ES.1.3.1.6.2** En porcentaje de incremento.
- RF.ES.1.3.1.6.2.1.** Campo calculado.
- RF.ES.1.3.1.6.2.2.** Número decimal con dos decimales.
- RF.ES.1.3.2.** Término de potencia.
 - RF.ES.1.3.2.1.** Todo dato debe de ser validado.
 - RF.ES.1.3.2.2.** Todos los valores deben de disponer un valor de referencia y un valor de referencia ajustada.
 - RF.ES.1.3.2.3.** Se dispondrá de los siguientes campos:
 - RF.ES.1.3.2.3.1** Potencia en kW.
 - RF.ES.1.3.2.3.1.1.** Campo introducido por el usuario.
 - RF.ES.1.3.2.3.1.2.** Número decimal con dos decimales.
 - RF.ES.1.3.2.3.1.3.** Número decimal mayor a 0.
 - RF.ES.1.3.2.3.2** Tarifa en unidades monetarias y kW.
 - RF.ES.1.3.2.3.2.1.** Campo introducido por el usuario.
 - RF.ES.1.3.2.3.2.2.** Número decimal con dos decimales.
 - RF.ES.1.3.2.3.2.3.** Número decimal mayor a 0.
 - RF.ES.1.3.2.3.3** Total
 - RF.ES.1.3.2.3.3.1.** Campo calculado.
 - RF.ES.1.3.2.3.3.2.** Número decimal con dos decimales.
 - RF.ES.1.3.2.4.** El sistema mostrará los periodos:
 - RF.ES.1.3.2.4.1** Existirá un periodo por termino de potencia diferente.
 - RF.ES.1.3.2.4.2** El sistema debe de ser configurable y parametrizable.
 - RF.ES.1.3.2.4.3** Debe de existir un mínimo de X periodos.
- RF.ES.1.3.3.** Término de energía.
 - RF.ES.1.3.3.1.** Todo dato debe de ser validado.
 - RF.ES.1.3.3.2.** Todos los valores deben de disponer un valor de referencia y un valor de referencia ajustada.
 - RF.ES.1.3.3.3.** Se dispondrá de los siguientes campos:
 - RF.ES.1.3.3.3.1** Consumo anual en kW.
 - RF.ES.1.3.3.3.1.1.** Campo introducido por el usuario.
 - RF.ES.1.3.3.3.1.2.** Número decimal con dos decimales.
 - RF.ES.1.3.3.3.1.3.** Número decimal mayor a 0.
 - RF.ES.1.3.3.3.2** Discriminación horaria.
 - RF.ES.1.3.3.3.2.1.** Campo introducido por el usuario.
 - RF.ES.1.3.3.3.2.2.** Número decimal con dos decimales.
 - RF.ES.1.3.3.3.2.3.** Número decimal mayor a 0.
 - RF.ES.1.3.3.3.2.3.1.** El sistema pedirá la siguiente información:
 - RF.ES.1.3.3.3.2.3.1.1.** Se dispondrá de tantos periodos de consumo como la parametrización determine.
 - RF.ES.1.3.3.3.2.3.2.** Se mostrará el nombre del periodo.
 - RF.ES.1.3.3.3.2.3.3.** Cada periodo debe de sumar el 100%.
 - RF.ES.1.3.3.3.3** Tarifa de coste de energía.



- RF.ES.1.3.3.3.1.** Campo introducido por el usuario.
- RF.ES.1.3.3.3.2.** Número decimal con dos decimales.
- RF.ES.1.3.3.3.3.** Número decimal mayor a 0.
- RF.ES.1.3.4.** Comentario.
 - RF.ES.1.3.4.1.** Por defecto, será vacío.
 - RF.ES.1.3.4.2.** Podrá ser introducido por el usuario.
- RF.ES.1.4.** El sistema debe de permitir recargar los cálculos.
 - RF.ES.1.4.1.** Si se realiza de forma correcta, el sistema actualizará los datos por los actuales.
 - RF.ES.1.4.2.** En caso de fallar, se mostrará un mensaje de error.
- RF.ES.1.5.** El sistema debe de permitir guardar los cambios.
 - RF.ES.1.5.1.** Se guardarán los siguientes elementos:
 - RF.ES.1.5.1.1.** Potencias contrada.
 - RF.ES.1.5.1.2.** Tarifa de potencias contratadas.
 - RF.ES.1.5.1.3.** Consumo anual.
 - RF.ES.1.5.1.4.** Discriminación horaria por periodos.
 - RF.ES.1.5.1.5.** Tarifa de consumo eléctrico.
 - RF.ES.1.5.2.** El sistema debe permitir realizar la petición de guardado de datos.
 - RF.ES.1.5.2.1.** Se deberá de actualizar la información contenida en la tabla anteriormente.
 - RF.ES.1.5.2.2.** Se procederá a realizar el guardado.
 - RF.ES.1.5.2.2.1** Si se realiza de forma correcta, el sistema nos informará.
 - RF.ES.1.5.2.2.2** En caso de fallar, se mostrará un error.

Componente de Producción

Vista de detalles y edición

- RF.PS.1.** El sistema permitirá consultar los siguientes elementos:
 - RF.PS.1.1.** El sistema permitirá modificar los valores si no está publicado el FactoryModel.
 - RF.PS.1.2.** Se dispondrá de los siguientes elementos:
 - RF.PS.1.2.1.** Horas anuales
 - RF.PS.1.2.1.1.** El valor por defecto será de X.
 - RF.PS.1.2.1.2.** Campo introducido por el usuario.
 - RF.PS.1.2.1.3.** Número decimal con dos decimales.
 - RF.PS.1.2.1.4.** Número positivo.
 - RF.PS.1.2.2.** Variación de eficiencia técnica.
 - RF.PS.1.2.2.1.** Campo introducido por el usuario.
 - RF.PS.1.2.2.2.** Número decimal con dos decimales.
 - RF.PS.1.2.2.3.** Número positivo.
 - RF.PS.1.2.2.4.** Porcentaje.
 - RF.PS.1.2.3.** Variación de eficiencia real.



- RF.PS.1.2.3.1.** Campo introducido por el usuario.
- RF.PS.1.2.3.2.** Número decimal con dos decimales.
- RF.PS.1.2.3.3.** Número positivo.
- RF.PS.1.2.3.4.** Porcentaje.
- RF.PS.1.2.4.** Variación del Ratio de UD por unidades de tiempo.
 - RF.PS.1.2.4.1.** Campo introducido por el usuario.
 - RF.PS.1.2.4.2.** Número decimal con dos decimales.
 - RF.PS.1.2.4.3.** Número positivo.
 - RF.PS.1.2.4.4.** Porcentaje.
 - RF.PS.1.2.4.5.** Potenciales ganancias
 - RF.PS.1.2.4.5.1** En unidades monetarias
 - RF.PS.1.2.4.5.1.1.** Campo calculado.
 - RF.PS.1.2.4.5.1.2.** Número decimal con dos decimales.
 - RF.PS.1.2.4.5.2** En porcentaje de incremento.
 - RF.PS.1.2.4.5.2.1.** Campo calculado.
 - RF.PS.1.2.4.5.2.2.** Número decimal con dos decimales.
- RF.PS.1.2.5.** Productos
 - RF.PS.1.2.5.1.** Todos los valores deben de disponer un valor de referencia y un valor de referencia ajustada.
 - RF.PS.1.2.5.2.** Gestión similar a una tabla Excel:
 - RF.PS.1.2.5.2.1** El sistema permitirá añadir una nueva fila.
 - RF.PS.1.2.5.2.2** El sistema permitirá eliminar una fila.
 - RF.PS.1.2.5.2.3** El sistema mostrará como mínimo X filas.
 - RF.PS.1.2.5.3.** Se podrá descargar los datos de la tabla en un formato compatible con Excel.
 - RF.PS.1.2.5.4.** Se dispondrá de la siguiente información:
 - RF.PS.1.2.5.4.1** Nombre del producto.
 - RF.PS.1.2.5.4.1.1.** Campo introducido por el usuario.
 - RF.PS.1.2.5.4.1.2.** Será introducido por el usuario en referencia.
 - RF.PS.1.2.5.4.1.3.** Campo calculado en referencia ajustada.
 - RF.PS.1.2.5.4.1.4.** No se aceptará un valor vacío.
 - RF.PS.1.2.5.4.2** Producción teórica.
 - RF.PS.1.2.5.4.2.1.** Será introducido por el usuario en referencia.
 - RF.PS.1.2.5.4.2.2.** Campo calculado en referencia ajustada.
 - RF.PS.1.2.5.4.2.3.** Número decimal positivo con dos decimales.
 - RF.PS.1.2.5.4.3** Producción técnica.
 - RF.PS.1.2.5.4.3.1.** Será introducido por el usuario en referencia.
 - RF.PS.1.2.5.4.3.2.** Campo calculado en referencia ajustada.
 - RF.PS.1.2.5.4.3.3.** Número decimal positivo con dos decimales.
 - RF.PS.1.2.5.4.4** Producción real.
 - RF.PS.1.2.5.4.4.1.** Será introducido por el usuario en referencia.
 - RF.PS.1.2.5.4.4.2.** Campo calculado en referencia ajustada.
 - RF.PS.1.2.5.4.4.3.** Número decimal positivo con dos decimales.
 - RF.PS.1.2.5.4.5** Pérdida Técnica.



- RF.PS.1.2.5.4.5.1.** Campo calculado.
- RF.PS.1.2.5.4.5.2.** Número decimal con dos decimales.
- RF.PS.1.2.5.4.6** Eficiencia Técnica.
 - RF.PS.1.2.5.4.6.1.** Campo calculado.
 - RF.PS.1.2.5.4.6.2.** Número decimal con dos decimales.
- RF.PS.1.2.5.4.7** Pérdida Real.
 - RF.PS.1.2.5.4.7.1.** Campo calculado.
 - RF.PS.1.2.5.4.7.2.** Número decimal con dos decimales.
- RF.PS.1.2.5.4.8** Eficiencia Real.
 - RF.PS.1.2.5.4.8.1.** Campo calculado.
 - RF.PS.1.2.5.4.8.2.** Número decimal con dos decimales.
- RF.PS.1.2.5.4.9** Horas totales de trabajo.
 - RF.PS.1.2.5.4.9.1.** Campo introducido por el usuario en referencia.
 - RF.PS.1.2.5.4.9.2.** Campo calculado en referencia ajustada.
 - RF.PS.1.2.5.4.9.3.** Número decimal con dos decimales.
- RF.PS.1.2.5.4.10** Porcentaje de horas totales de trabajo
 - RF.PS.1.2.5.4.10.1.** Campo calculado.
 - RF.PS.1.2.5.4.10.2.** Número decimal con dos decimales.
- RF.PS.1.2.5.4.11** Unidad por unidad temporal
 - RF.PS.1.2.5.4.11.1.** Campo calculado.
 - RF.PS.1.2.5.4.11.2.** Número decimal con dos decimales.
- RF.PS.1.2.5.4.12** Precio por unidad
 - RF.PS.1.2.5.4.12.1.** Campo introducido por el usuario.
 - RF.PS.1.2.5.4.12.2.** Número decimal con dos decimales.
- RF.PS.1.2.5.4.13** Total
 - RF.PS.1.2.5.4.13.1.** Campo calculado.
 - RF.PS.1.2.5.4.13.2.** Número decimal con dos decimales.
- RF.PS.1.2.6.** Comentario.
 - RF.PS.1.2.6.1.** Por defecto, será vacío.
 - RF.PS.1.2.6.2.** Podrá ser introducido por el usuario.
- RF.PS.1.3.** El sistema debe de permitir recargar los cálculos.
 - RF.PS.1.3.1.** Si se realiza de forma correcta, el sistema actualizará los datos por los actuales.
 - RF.PS.1.3.2.** En caso de fallar, se mostrará un mensaje de error.
- RF.PS.1.4.** El sistema debe de permitir guardar los cambios.
 - RF.PS.1.4.1.** Se guardarán los siguientes elementos:
 - RF.PS.1.4.1.1.** Coste de mano de obra.
 - RF.PS.1.4.1.2.** Número de trabajadores.
 - RF.PS.1.4.1.3.** Horas anuales de mano de obra.
 - RF.PS.1.4.1.4.** Producción.
 - RF.PS.1.4.1.5.** Costes variables.
 - RF.PS.1.4.1.6.** Costes fijos.
 - RF.PS.1.4.1.7.** Comentario.
 - RF.PS.1.4.2.** El sistema debe permitir realizar la petición de guardado de datos.



RF.PS.1.4.2.1. Se deberá de actualizar la información contenida en la tabla anteriormente.

RF.PS.1.4.2.2. Se procederá a realizar el guardado.

RF.PS.1.4.2.2.1 Si se realiza de forma correcta, el sistema nos informará.

RF.PS.1.4.2.2.2 En caso de fallar, se mostrará un error.

Componente de Calidad

Vista de detalles y edición

RF.QS.1. El sistema permitirá consultar los siguientes elementos:

RF.QS.1.1. El sistema permitirá modificar los valores si no está publicado el FactoryModel.

RF.QS.1.2. Todo valor dispondrá de referencia y referencia ajustada.

RF.QS.1.3. Se dispondrá de los siguientes elementos:

RF.QS.1.4. Nombre del producto.

RF.QS.1.4.1. Se encargará de cargar la información necesaria del producto.

RF.QS.1.5. Ahorro total

RF.QS.1.5.1. En unidades monetarias

RF.QS.1.5.1.1. Campo calculado.

RF.QS.1.5.1.2. Número decimal con dos decimales.

RF.QS.1.5.2. En porcentaje de incremento.

RF.QS.1.5.2.1. Campo calculado.

RF.QS.1.5.2.2. Número decimal con dos decimales.

RF.QS.1.6. Vista de productos.

RF.QS.1.6.1. Gestión similar a una tabla Excel:

RF.QS.1.6.1.1. El sistema permitirá añadir una nueva fila.

RF.QS.1.6.1.2. El sistema permitirá eliminar una fila.

RF.QS.1.6.1.3. El sistema mostrará como mínimo X filas.

RF.QS.1.6.2. El sistema dispondrá de los siguientes elementos:

RF.QS.1.6.2.1. Nombre del producto

RF.QS.1.6.2.1.1 Campo calculado.

RF.QS.1.6.2.2. Producción real.

RF.QS.1.6.2.2.1 Campo calculado.

RF.QS.1.6.2.2.2 Número decimal con dos decimales.

RF.QS.1.6.2.3. Número de defectos que tiene el producto.

RF.QS.1.6.2.3.1 Campo calculado.

RF.QS.1.6.2.3.2 Número decimal con dos decimales.

RF.QS.1.6.2.4. Ratio de defectos.

RF.QS.1.6.2.4.1 Campo calculado.

RF.QS.1.6.2.4.2 Número decimal con dos decimales.

RF.QS.1.6.2.5. Importe de defectos en unidades monetarias.



- RF.QS.1.6.2.5.1** Campo calculado.
- RF.QS.1.6.2.5.2** Número decimal con dos decimales.
- RF.QS.1.6.2.6.** Reproducción de defectos unidad por producción.
 - RF.QS.1.6.2.6.1** Campo calculado.
 - RF.QS.1.6.2.6.2** Número decimal con dos decimales.
- RF.QS.1.6.2.7.** Ahorro
 - RF.QS.1.6.2.7.1** Campo calculado.
 - RF.QS.1.6.2.7.2** Número decimal con dos decimales.
 - RF.QS.1.6.2.7.3** Debe de existir únicamente en referencia ajustada.
- RF.QS.1.7.** Vista de defectos
 - RF.QS.1.7.1.** Gestión similar a una tabla Excel:
 - RF.QS.1.7.1.1.** El sistema permitirá añadir una nueva fila.
 - RF.QS.1.7.1.2.** El sistema permitirá eliminar una fila.
 - RF.QS.1.7.1.3.** El sistema mostrará como mínimo X filas.
 - RF.QS.1.7.2.** El sistema dispondrá de los siguientes elementos:
 - RF.QS.1.7.2.1.** Descripción del defecto
 - RF.QS.1.7.2.1.1** Campo introducido por el usuario.
 - RF.QS.1.7.2.2.** Número de defectos.
 - RF.QS.1.7.2.2.1** Campo introducido por el usuario.
 - RF.QS.1.7.2.2.2** Número decimal con dos decimales.
 - RF.QS.1.7.2.3.** Precio unidad
 - RF.QS.1.7.2.3.1** Campo calculado.
 - RF.QS.1.7.2.3.2** Número decimal con dos decimales.
 - RF.QS.1.7.2.4.** Coste recuperable por mano de obra.
 - RF.QS.1.7.2.4.1** Campo introducido por el usuario.
 - RF.QS.1.7.2.4.2** Número decimal con dos decimales.
 - RF.QS.1.7.2.5.** Importe de defectos en unidades monetarias.
 - RF.QS.1.7.2.5.1** Campo calculado.
 - RF.QS.1.7.2.5.2** Número decimal con dos decimales.
- RF.QS.1.8.** Comentario.
 - RF.QS.1.8.1.** Por defecto, será vacío.
 - RF.QS.1.8.2.** Podrá ser introducido por el usuario.
- RF.QS.1.9.** El sistema debe de permitir recargar los cálculos.
 - RF.QS.1.9.1.** Si se realiza de forma correcta, el sistema actualizará los datos por los actuales.
 - RF.QS.1.9.2.** En caso de fallar, se mostrará un mensaje de error.
- RF.QS.1.10.** El sistema debe de permitir guardar los cambios.
 - RF.QS.1.10.1.** Se guardarán los siguientes elementos:
 - RF.QS.1.10.1.1.** Coste de mano de obra.
 - RF.QS.1.10.1.2.** Número de trabajadores.
 - RF.QS.1.10.1.3.** Horas anuales de mano de obra.
 - RF.QS.1.10.1.4.** Producción.
 - RF.QS.1.10.1.5.** Costes variables.
 - RF.QS.1.10.1.6.** Costes fijos.
 - RF.QS.1.10.1.7.** Comentario.



-
- RF.QS.1.10.2.** El sistema debe permitir realizar la petición de guardado de datos.
- RF.QS.1.10.2.1.** Se deberá de actualizar la información contenida en la tabla anteriormente.
- RF.QS.1.10.2.2.** Se procederá a realizar el guardado.
- RF.QS.1.10.2.2.1** Si se realiza de forma correcta, el sistema nos informará.
- RF.QS.1.10.2.2.2** En caso de fallar, se mostrará un error.