Universidad de Oviedo

IRIS TOKEN SERVER:

DESARROLLO DE UN SERVIDOR DE TOKENIZACIÓN PARA ADAPTACIÓN A NORMATIVA PCI DE APLICACIONES CON GESTIÓN DE TARJETAS DE CRÉDITO

ESCUELA DE INGENIERÍA INFORMÁTICA DE OVIEDO

TRABAJO FIN DE MÁSTER

DIRECTOR: CÉSAR FERNÁNDEZ ACEBAL

CODIRECTOR: ALBERTO MANUEL FERNÁNDEZ ÁLVAREZ

AUTOR: LILIANA VILLAR IGLESIAS

JULIO DE 2018

Resumen

Cuando un usuario proporciona los datos de su tarjeta de crédito a través de un medio electrónico pone su confianza en que el sistema que está gestionando estos datos tan sensibles, está tomando todas las medidas de seguridad necesarias para asegurar que ningún tercero malintencionado pueda acceder a ellos y provocarle un perjuicio.

Este es el principal objetivo de la normativa PCI – DSS, asegurar que todos aquellos sistemas que manipulen o almacenen datos relativos a tarjetas de crédito tomen las medidas de seguridad oportunas para asegurar su tratamiento.

Recientemente esta normativa ha pasado a ser de obligado cumplimiento para las agencias de viaje que cuentan con sistemas que permiten, a sus clientes, realizar el pago de sus reservas. En este contexto la empresa Grupo Iris se ve afectada por la entrada en vigor de esta normativa, pues su principal actividad está relacionada con el desarrollo de soluciones utilizadas, entre otros, por las agencias de viaje.

El objetivo de este proyecto es crear un sistema de tokenización que permita simplificar al máximo la tarea de adaptación de los sistemas desarrollados por Grupo Iris a la normativa PCI.

Palabras clave

Token, PCI, PAN, tarjeta de crédito, seguridad.

Abstract

When a user provides his credit card information through an electronic means, he puts his trust in that the system that is managing this sensitive data, has taking all necessary security measures to ensure that no malicious third party can access them and cause him harm.

This is the main objective of the PCI - DSS standar, to ensure that all those systems that manipulate or store data related to credit cards, take the appropriate security measures to ensure their treatment.

Recently, this regulation has become mandatory for travel agencies that have systems that allow their customers to pay their reserves. In this context, the company Grupo Iris is affected by the entry into force of this regulation, since its main activity is related to the development of solutions used, among others, by travel agencies.

The objective of this project is to create a tokenization system that allows for the simplification of the task of adapting the systems developed by Grupo Iris to the PCI standard.

Keywords

Token, PCI, PAN, credit card, security.

Índice general

Índice de figuras

Índice de tablas

1. INTRODUCCIÓN

1.1 JUSTIFICACIÓN DEL PROYECTO

La industria aérea sufre unas perdidas aproximadas de 1.000 millones de dólares anuales debido al fraude con tarjetas de crédito. Por eso la Asociación Internacional de Transporte Aéreo, IATA, ha decido obligar a todas sus agencias de viaje a cumplir con el Estándar de Seguridad de Datos para la Industria de Tarjeta de Pago o PCI DSS[1].

Dicha medida entró en vigor el 1 de marzo de 2018 y su intención es la de reforzar la seguridad de los pagos realizados por las agencias de viaje.

La empresa Grupo Iris se encarga del desarrollo de aplicaciones que son utilizadas por las agencias de viajes, para llevar a cabo la gestión de su actividad. Entre las tareas que las aplicaciones desarrolladas permiten realizan, se encuentran las de pagos a través de TPV y el almacenamiento de datos de pago para su procesamiento. Debido a esto, las aplicaciones desarrolladas y mantenidas por las empresa se ven afectadas por la entrada en vigor de la medida adoptada por IATA y debe adecuar sus desarrollos a las medidas establecidas por PCI, para así poder seguir sirviendo como una herramienta de trabajo a las agencias de viaje implicadas.

Tras realizar un análisis sobre la situación actual de estas aplicaciones, se detecta el uso de datos de pago en todas las aplicaciones y servicios relacionados (backups, logs, entornos de desarrollo, etc.). Dadas las circunstancias, resultaría muy complicado y tedioso verificar el cumplimiento de todos los requisitos PCI en todos los entornos en los que hay presencia de datos de pago sensibles.

Bajo este escenario surge la necesidad de crear este proyecto, como una solución que simplifique la adaptación que debe llevarse a cabo. Se propone emplear un sistema de tokenización, un servidor de tokens, de forma que todos los PAN (Primary Account Number) se encuentren almacenados en este servidor, mientras que el resto de datos de la tarjeta permanecen en los sistemas actuales y, en ellos, el PAN es sustituido por un token.

Esta disposición simplifica el esfuerzo de adaptación del resto de sistemas, concentrando los cambios en puntos concretos de la funcionalidad actual. El servidor de tokens y las aplicaciones que necesiten manejar el PAN serán las realmente afectadas por la normativa PCI, siendo más sencillo pasar la certificación al reducirse el ámbito.

[1] https://www.pcisecuritystandards.org/

1.2 OBJETIVOS DEL PROYECTO

El principal objetivo del proyecto es la creación de un sistema que permita adecuar aplicaciones, que actualmente almacenan y tratan información sensible de tarjetas de crédito, al cumplimiento del estándar PCI.

Para esto, se crea un sistema capaz de servir como almacén del PAN de las tarjetas de crédito, de forma que las aplicaciones deleguen en este nuevo sistema, el almacenamiento de dicha información. Ese sistema deberá permitir, por tanto, que otras aplicaciones soliciten el almacenamiento de un nuevo PAN cuando lo necesiten, lo que desencadenará la creación de un nuevo token que será remitido a la aplicación solicitante, para que lo almacene en lugar del PAN inicial.

Un requisito básico de este proyecto, es asegurar el correcto almacenamiento de los PAN. Es decir, el sistema no puede simplemente guardar el dato, sino que debe hacerlo bajo unas estrictas medidas de seguridad que cumplan con los requisitos establecidos en la normativa PCI. Puesto que las aplicaciones delegarán en este servidor la responsabilidad de almacenar este dato sensible, el sistema debe proporcionar un entorno seguro de almacenamiento.

Este dato no será únicamente guardado por el sistema si no que, para su correcto funcionamiento, las aplicaciones necesitarán, en ciertas ocasiones, acceder a los PAN para poder realizar sus actividades normales como la ejecución de pagos. Teniendo en cuenta este aspecto, se forma otro de los claros objetivos del proyecto, que consiste en asegurar el acceso al servidor de tokens.

El sistema debe realizar un correcto proceso de autenticación, asegurando por completo que únicamente la aplicación que ha registrado el token en el sistema, que cuentan con permiso para ello y que accede desde una zona de red segura, pueda obtener el valor del PAN.

1.3 ESTUDIO DE LA SITUACIÓN ACTUAL

1.3.1 Evaluación de alternativas

A continuación se realiza un breve análisis sobre las posibles alternativas que podrían haberse adoptado para conseguir la adaptación de las aplicaciones con las que cuenta la empresa a la normativa PCI – DSS

ADAPTACIÓN DE TODAS LAS APLICACIONES

Cuando surge la necesidad de cumplir con la normativa, la primera perspectiva que se tiene es esta. La norma obliga a tener en cuenta todas las aplicaciones que gestionan datos de tarjetas de crédito, por lo que el primer análisis que se hace es que será necesario modificar cada una de las aplicaciones con las que se cuenta, para adecuarse a los requisitos establecidos en el estándar.

Sin embargo, una vez analizado el alcance y la complejidad que se establece en la normativa PCI queda de manifiesto que el esfuerzo que habría que llevar a cabo, tanto en tiempo como en recursos, para poder ejecutar esta opción, sería exageradamente elevado. De aquí nace la necesidad de plantearse otras alternativas que simplifiquen, en la medida de lo posible el proceso de adaptación a la normativa.

SISTEMA DE TOKENIZACIÓN COMERCIAL

Desde la propia normativa PCI se establece que una de las posibles soluciones que simplifican el esfuerzo de adaptación es la utilización de tokens que sustituyan el PAN en todos aquellos lugares en los que hasta el momento se estuviese almacenando.

Una vez planteada esta opción, aparece la posibilidad de utilizar uno de los múltiples sistemas de tokenización existentes en el mercado. En concreto se analizó el posible uso de los siguientes:

- RSA Data Protection Manager[2]
- SafeNet Tokenization Manager[3]
- Voltage Secure Staless[4]

Sin embargo, puesto que no se trata de una adaptación temporal, sino que los sistemas de la empresa van a tener que estar condicionados a esta normativa a muy largo plazo, utilizar uno de estos sistemas conllevaría un coste incremental a lo largo del tiempo. Además, al tratarse de una solución comercial, serían las aplicaciones las que tendrían que adaptarse al sistema de tokenización y no al contrario.

Por todo esto se opta finalmente por el desarrollo de un sistema de tokenización propio, que permita diseñarlo de tal forma que se adapte lo mejor posible a las aplicaciones existentes, facilitando el trabajo que posteriormente habrá que realizar sobre cada una de ellas.

[2] http://www.emc.com/security/rsa-data-protection-manager.htm
[3] http://www.safenet-inc.com/data-protection/data-encryption/tokenization-manager/
[4] http://www.voltage.com/products/securedata-enterprise/tokenization/

1.4 SOLUCIÓN PROPUESTA

Se propone la creación de un sistema de tokenización que se encargue de almacenar, de forma segura, los PAN de las tarjetas de crédito que se manejan en diferentes aplicaciones y proporcionar a estas un token que sea utilizado como sustituto del valor inicial.

A continuación se muestra como interactuarán las diferentes aplicaciones involucradas con el servidor de tokens propuesto.

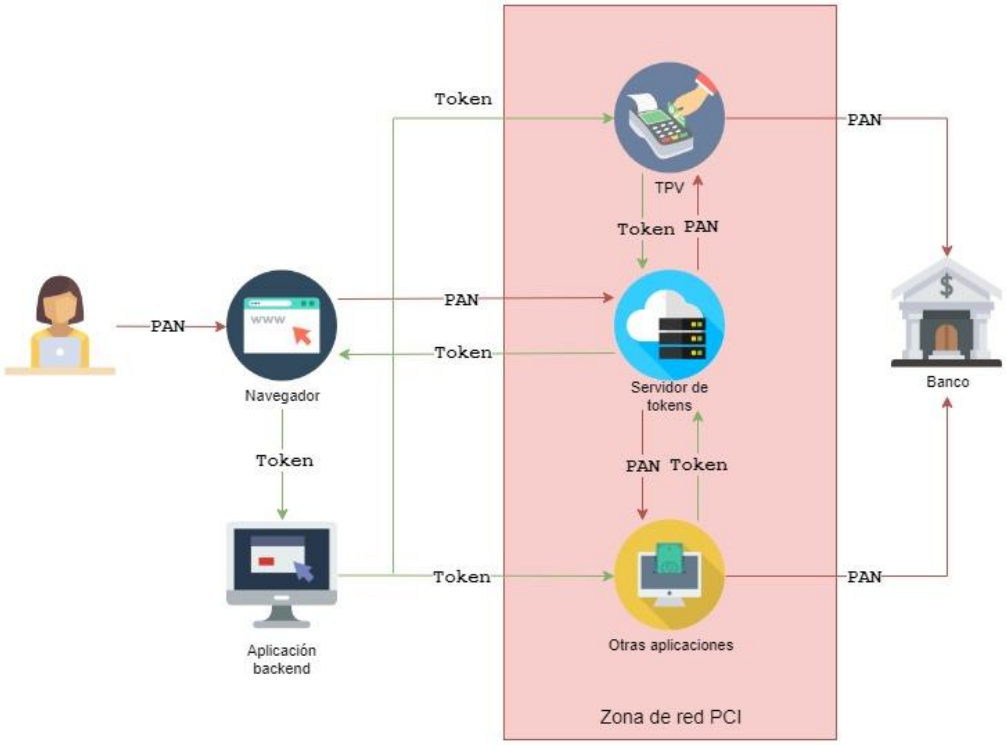

Figura 1. Diagrama de interacción del sistema

En la anterior figura se puede observar cómo se distribuye el uso del PAN y el token a lo largo de todo el sistema.

- El usuario final introduce el PAN de su tarjeta de crédito en el formulario que la aplicación backend ha dispuesto para ello.
- Desde el navegador el PAN es interceptado por el servidor de tokens que se encarga de sustituir el valor del PAN por el del token. De esta forma el navegador le devuelve a la aplicación backend el token y, en ningún caso, el PAN.
- En aquellas situaciones en las que la aplicación backend necesite llevar a cabo acciones en las que es necesario contar con el valor del PAN, no lo hará directamente, sino que se comunicará con las aplicaciones que se encuentran dentro de la zona de red PCI, puesto que este es el único lugar donde la transformación de token a PAN puede tener lugar. Por lo tanto, la aplicación backend le proporcionará, por ejemplo, al TPV el valor del token y este, ubicado en la zona PCI, interactuará con el servidor de tokens para obtener el valor del PAN y poder llevar a cabo la operación deseada.

Lo que se consigue con esta distribución es que el valor del PAN nunca salga hacia una zona desprotegida o desautorizada. Las aplicaciones que se ejecutan en una red menos estrictamente protegida, únicamente trabajan con el valor del token.

El sistema estará conformado por varios interfaces. Cada uno de ellos ofrece funcionalidad diferente, y la mínima necesaria, a cada uno de los sistemas externos involucradas. Además, por cada uno de ellos se establecerán los controle de acceso precisos.

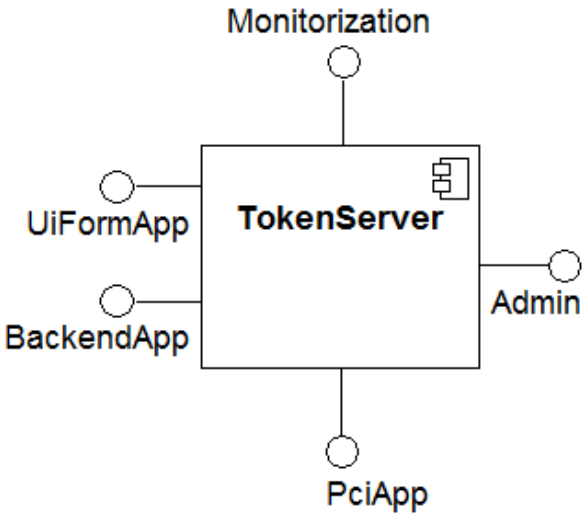

Figura 2. Interfaces del sistema.

UIFORMAPP

Interfaz de servicios destinada a recibir peticiones de conversión de PAN a token desde formularios en aplicaciones externas. En el formulario habrá al menos un campo para la PAN y se interceptará el evento de entrega. Al disparo de dicho evento, se invocará este servicio que devolverá el token para el PAN. El valor del token sustituye al de la PAN y así a la aplicación backend le llega el token.

BACKENDAPP

Interfaz de servicios destinada a las aplicaciones backend y que ofrece servicios para:

- Eliminar un token. Cuando la aplicación debe borrar datos de tarjeta se lo notifica al servidor de tokens a través de esta operación.
- Obtener PAN enmascarado para un token. Se podrán obtener tres tipos de enmascarado, como se analiza en Formatos de máscara y la aplicación podría almacenar el PAN enmascarado.

PCIAPP

Interfaz de servicios destinada a las aplicaciones de la zona PCI, que son las únicas capacitadas para recuperar el PAN a partir del token. Este será el único servicio proporcionado por este interfaz.

ADMIN

Interfaz de usuario que permitirá al administrador realizar los casos de uso de mantenimiento que le son asignados:

- Gestión de accesos
- Gestión de claves de encriptación
- Gestión de tokens

MONITOR

Interfaz de página web única para ser invocada por un sistema de monitorización. Ofrecerá una página HTML con el estado del sistema. Si en esta página aparece una palabra clave se disparará una alarma en el control de sistemas.

2. ASPECTOS TEÓRICOS

2.1 NORMATIVA PCI - DSS

En los inicios de los años 2000 los casos de fraude con tarjetas de crédito aumentaron alarmantemente lo que suponía un gran problema para las grandes entidades financieras encargadas de la gestión de tarjetas. Por ello, en el año 2006, las principales empresas de tarjetas: VISA, American Express, JCB, Mastercard y Discover, crearon el estándar conocido como PCI-DSS o Payment Card Industry – Data Security Estándar.

La principal función de este estándar es definir los requisitos mínimos de seguridad que todas las empresas que transmitan, procesen o almacenen información de tarjetas de crédito deben cumplir. Los datos que se incluyen en una tarjeta de crédito o débito son:

- Número PAN, que es el número, habitualmente de 16 dígitos, que se encuentra en el frente de la tarjeta.
- El nombre del tarjetahabiente o titular de la tarjeta.
- La fecha de expiración de la tarjeta.
- El código de servicio, que es el código de 3 o 4 dígitos que se encuentra en la parte posterior de la tarjeta.

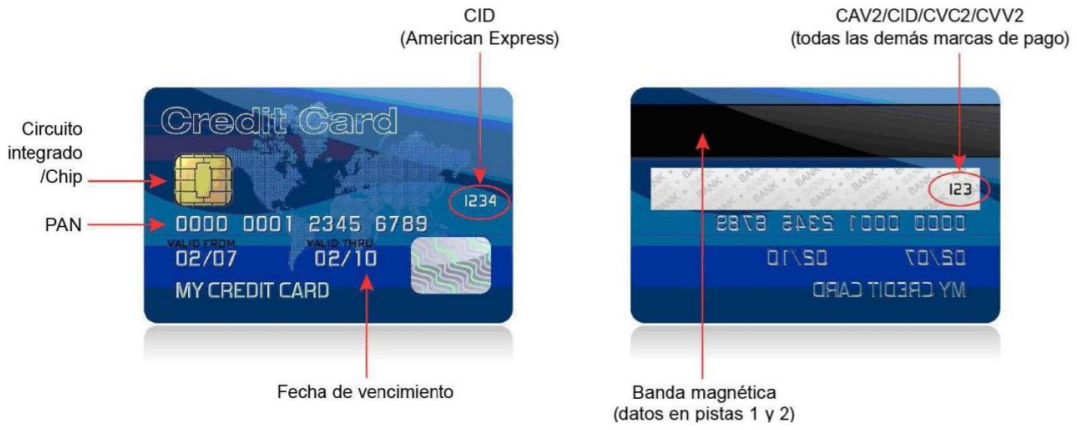

Figura 3. Tarjeta de crédito

Toda empresa o entidad que procese, almacene o transmita datos de tarjetas de crédito debe cumplir los requisitos establecidos por la normativa. A continuación se indica un resumen de estos requisitos, con intención de simplificar la especificación puesto que esta cuenta con más de 240 requisitos.

- Desarrollar y mantener una red segura.
  - Instalar y mantener una configuración de firewall para proteger los datos del titular de la tarjeta.
  - No usar valores predeterminados suministrador por el proveedor para las contraseñas del sistema y otros parámetros de seguridad.

- Proteger los datos del titular de la tarjeta.
  - Proteger los datos del titulas de la tarjeta que fueron almacenados.
  - Cifrar la transmisión de los datos del titulas de la tarjeta en las redes públicas abiertas.
- Mantener un programa de administración de vulnerabilidad.
  - Proteger todos los sistemas contra malware y actualizar los programas o software antivirus regularmente.
  - Desarrollar y mantener sistemas y aplicaciones seguros.
- Implementar medidas sólidas de control de acceso.
  - Restringir el acceso a los datos del titular de la tarjeta según la necesidad de saber que tenga la empresa.
  - Identificar y autenticar el acceso a los componentes del sistema.
  - Restringir el acceso físico a los datos del titular de la tarjeta.
- Supervisar y evaluar las redes con regularidad.
  - Rastrear y supervisar todos los accesos a los recursos de red y a los datos del titular de la tarjeta.
  - Probar con regularidad los sistemas y procesos de seguridad.
- Mantener una política de seguridad de información.
  - Mantener una política que aborde la seguridad de la información para todo el personal.

2.2 TOKENIZACIÓN

La normativa PCI indica que una de las posibles alternativas a la hora de implementar la protección del PAN es la de utilizar un sistema de tokenización, una solución sencilla pero muy efectiva.

Esta opción consiste en la sustitución del dato confidencial, en el caso de PCI el PAN, por otro dato que no necesita ser protegido. El proceso funciona de la siguiente forma:

- El sistema de tokenización recibe el dato confidencial.
- El dato confidencial es almacenado, protegido empleando algoritmos de cifrado, en un servidor de tokens.
- El sistema de tokenización genera un token único y crea una asociación entre este y el dato confidencial. La función de dicho token es la de ser utilizado como alias del dato que debe ser protegido.
- El token sustituye, en el resto de operaciones, al dato confidencial.

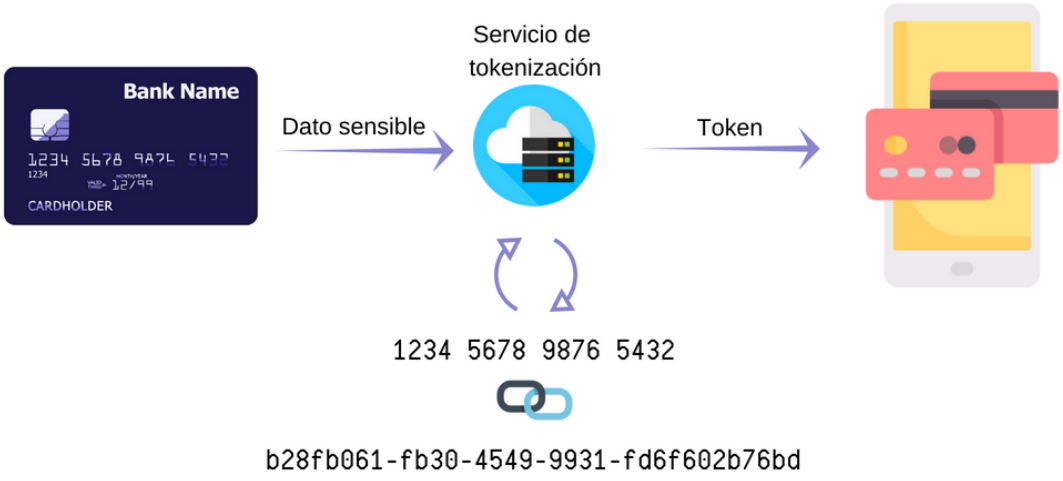

Figura 4. Servicio de tokenización

2.2.1 Tipos de tokens

Un punto clave a la hora de desarrollar un sistema de tokenización es seleccionar el tipo de token que se va a utilizar. En el siguiente diagrama se muestra la clasificación de los diferentes tokens:

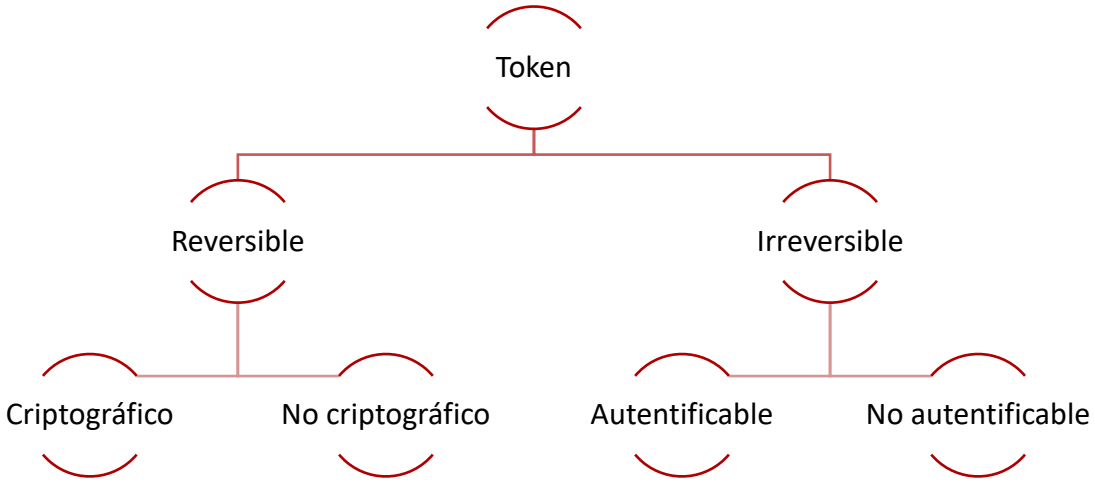

Figura 5. Clasificación de tokens

Los tokens reversibles son aquellos que permiten obtener el dato sensible original a través del valor del token y se distinguen:

- Criptográficos: se generan a partir del dato que se desea proteger utilizando algoritmos de cifrado. En este caso no se almacena ni el dato original ni el token en el servidor, únicamente la clave empleada el proceso de tokenización.
- No criptográficos: para cada dato confidencial se genera un valor aleatorio específico, ambos datos permanecen relacionados en una tabla que debe ser protegida correctamente.

En el caso de los tokens irreversibles no es posible obtener el dato original a partir del token y se subdividen en los siguientes:

- Autentificable: el token se genera aplicando una función matemática sobre el dato que se desea proteger. Si bien no se trata de un proceso reversible, a partir del dato original es posible validar el token asociado, aplicando nuevamente la misma función matemática sobre el dato.
- No autentificable: la forma de generación del token es idéntica al caso anterior, sin embargo, no es posible validar el token asociado al dato original, puesto que cada token generado es diferente aunque el dato de partida sea el mismo.

3. PLANIFICACIÓN Y PRESUPUESTO

3.1 PLANIFICACIÓN

En este apartado se desarrolla la descomposición jerárquica del trabajo que se lleva a cabo para poder ejecutar el proyecto, sirve para establecer el plazo de entrega de este, así como poder organizar el tiempo de trabajo.

3.1.1 Planificación preliminar

Al comienzo del proyecto se llevó a cabo una planificación temporal inicial, a muy grandes rasgos, con intención de tener unos objetivos temporales prestablecidos. Como se puede ver en la tabla que se muestra a continuación, únicamente se tuvieron en cuenta tareas de muy alto nivel, sin entrar en mayor detalle.

TAREA	DUR.	INICIO	FIN
Tutorías	1h/semana	5 de febrero de 2018	17 de junio de 2018
Estudio de normativas	14 días	5 de febrero de 2018	17 de febrero de 2018
Análisis del sistema	14 días	17 de febrero de 2018	28 de febrero de 2018
Diseño del sistema	14 días	28 de febrero de 2018	11 de marzo de 2018
Implementación	**55 días**	**11 de marzo de 2018**	**26 de abril de 2018**
Desarrollo de la funcionalidad	35 días	11 de marzo de 2018	9 de abril de 2018
Desarrollo de medidas de seguridad	20 días	9 de abril de 2018	26 de abril de 2018
Pruebas	21 días	26 de abril de 2018	13 de mayo de 2018
Documentación del proyecto	30 días	13 de mayo de 2018	7 de junio de 2018

Tabla 1. Planificación inicial

3.1.2 Planificación final

Tras realizar un pequeño estudio sobre todas las implicaciones del proyecto, se llevó a cabo una planificación más detallada, en la que se fijó el plan de trabajo a seguir a lo largo del desarrollo del proyecto.

Para realizar la planificación se tuvo en cuenta el horario de trabajo que el alumno seguiría, en este caso se prefijaron los siguientes horarios:

- De lunes a viernes de 16:00 a 21:00
- Sábados y domingos de 10:00 a 14:00 y de 16:00 a 22:00

A continuación se muestra, en forma de diagrama WBS, la distribución temporal de las tareas. En la primera descomposición se pueden observar las fases globales del proyecto, así como algunas tareas de transición.

Figura 6. Diagrama AWS general

INICIO DEL PROYECTO

La fase de inicio del proyecto sirve para definir, de forma clara e inequívoca, que se va a hacer. Para ello, en este caso, es necesario realizar un estudio sobre la normativa PCI, así como la situación actual, para poder delimitar claramente el alcance del proyecto y sus objetivos principales.

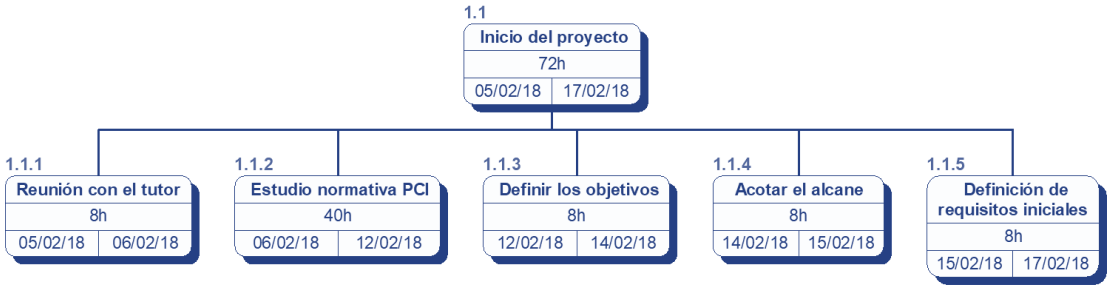

Figura 7. Diagrama WBS inicio del proyecto

ANÁLISIS DEL SISTEMA

Durante la fase de análisis se detalla, en profundidad, la funcionalidad del sistema, las restricciones de este, así como los criterios de aceptación.

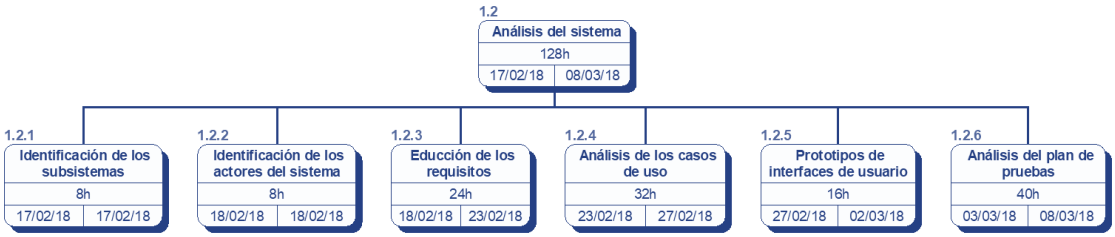

Figura 8. Diagrama WBS análisis del sistema

IMPLEMENTACIÓN

La fase de implementación se encuentra en su totalidad dedicada al desarrollo de los diferentes módulos o subsistemas que conforman la aplicación.

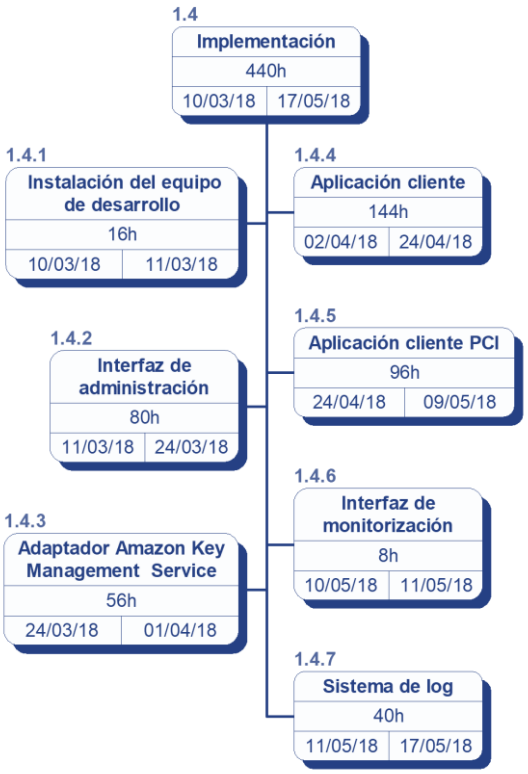

Figura 9. Diagrama WBS implementación

CIERRE DEL PROYECTO

Una vez concluidas las fases de implementación, pruebas y documentación, únicamente queda dar por concluido el proyecto bajo una pequeña fase de cierre que comprende la elaboración de los manuales del sistema, así como el correcto almacenamiento de todos los recursos generales durante la ejecución de este.

Figura 10. Diagrama WBS cierre del proyecto

DIAGRAMA DE GANTT

Por último, se muestra el diagrama de Gantt del proyecto, en el que se puede observar el tiempo de dedicación previsto para las diferentes tareas del proyecto, indicadas anteriormente.

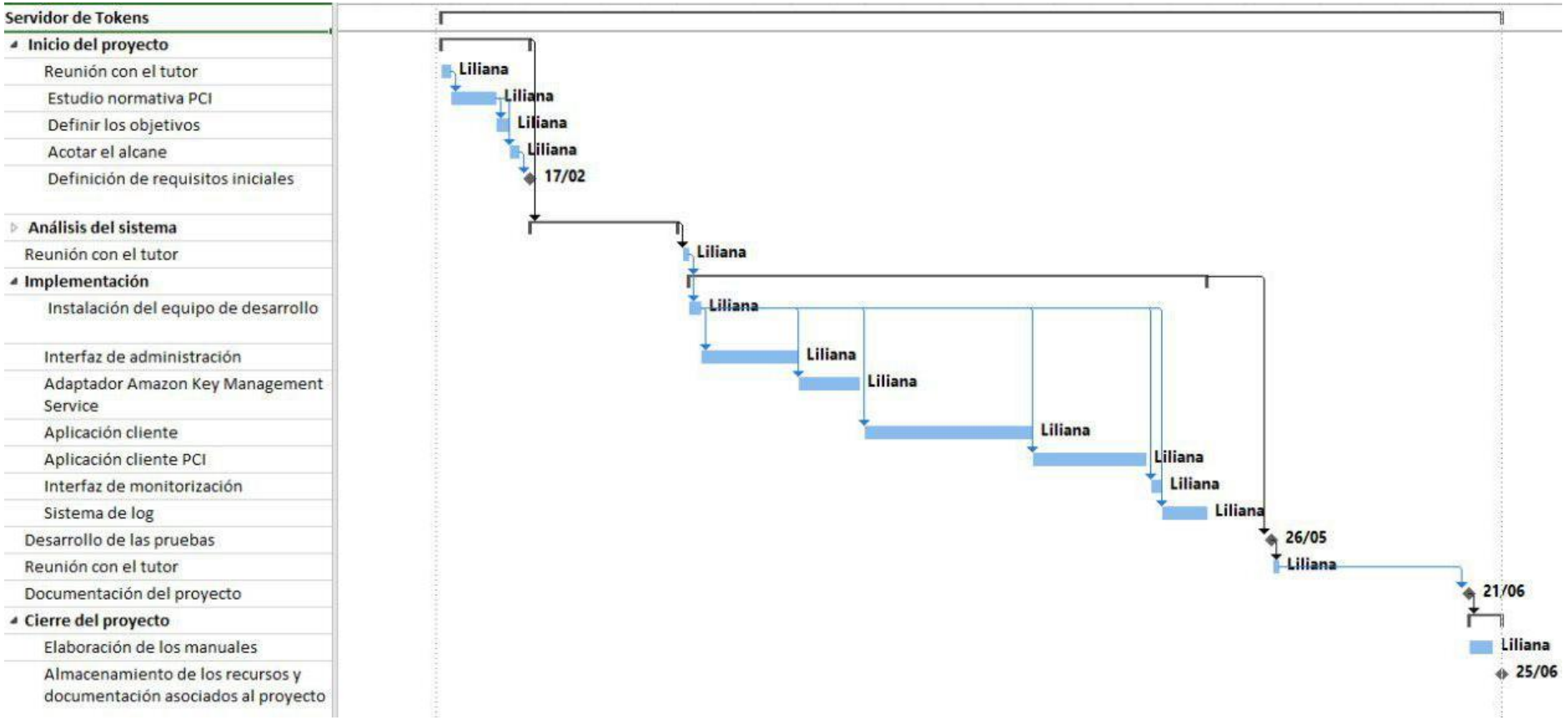

Figura 11. Diagrama de Gantt

3.2 PRESUPUESTO

A continuación se indica, de forma detallada, los costes derivados de la ejecución del presente proyecto,

3.2.1 Resumen del presupuesto

En primer lugar se muestra el proceso para obtener el presupuesto de ejecución del proyecto.

COSTES INDIRECTOS

Para la elaboración del proyecto únicamente será necesario contar con un ordenador completo con conexión a internet, por lo tanto, se calcula, en función de la duración de este, el porcentaje de amortización del equipo de desarrollo, para poder obtener su coste dentro del proyecto.

Teniendo en cuentan que un equipo habitual suele tener una vida útil de unos 5 años, se calcula el porcentaje de amortización de dicho equipo:

$$Porcentaje\ de\ amortización = \frac{113\ días\ de\ desarrollo}{1825\ días\ de\ vida\ útil} \approx 6{,}2\%$$

A partir del factor de amortización, se calcula el coste total que supondrá el equipo de desarrollo para el proyecto.

$$Coste\ equipo = \ 1.200€ \times \frac{6{,}2}{100} = 74{,}40\ €$$

Otro de los costes que hay que tener en cuenta a la hora de calcular los costes de ejecución del proyecto, es el coste derivado del uso de Amazon AWS, para el almacenamiento de las claves de encriptación. De este servicio se derivan dos costes:

- Cada clave de encriptación almacenada en el sistema supone un coste de 1$/mes
- Las primeras 20.000 solicitudes de encriptación no suponen ningún coste, a partir de ahí cada 10.000 peticiones supondrán un coste de 0.03$

En este caso únicamente es necesario tener en cuenta el coste relacionado con el almacenamiento de las claves de encriptación, puesto que no se van a superar las 20.000 peticiones de cifrado que se ofrecen de manera gratuita.

Teniendo en cuenta que el sistema necesita gestionar hasta cuatro claves de encriptación simultáneamente, se calcula el coste total del servicio.

$$Coste\ AWS = 4\ claves\ \times \frac{1\$ \sim 0{,}857€}{clave/mes} \times 4\ meses = \ 13{,}71\ €$$

A partir de todo lo anterior, a continuación se muestra un resumen de los costes indirectos del proyecto.

Concepto	**Coste**
Equipo de desarrollo	74,40 €
Amazon AWS	13,71 €
TOTAL	**88,11 €**

Tabla 2. Costes indirectos

PRESUPUESTO

Una vez calculados los costes indirectos del proyecto y teniendo en cuenta la planificación expuesta en el apartado anterior, se procede a realizar el presupuesto de ejecución. Para ello se tiene en cuenta la parte reservada para cubrir posibles riesgos que pudiesen afectar al proyecto así como el beneficio que se espera obtener.

Concepto	Horas	Precio/hora	Total
Inicio del proyecto	72	6,70 €	482,40 €
Análisis del sistema	136	6,70 €	911,20 €
Implementación	440	6,70 €	2948,00 €
Desarrollo de las pruebas	64	6,70 €	428,80 €
Documentación	160	6,70 €	1072,00 €
Cierre del proyecto	32	6,70 €	214,40 €
		PRESUPUESTO DE EJECUCIÓN MATERIAL	**6056,80 €**
		Costes indirectos	88,11 €
		Plan de contingencias (4%)	242,27 €
		Beneficio (12%)	726,81 €
		TOTAL	**7113,99 €**

Tabla 3. Presupuesto

3.2.2 Presupuesto para el cliente

El objetivo principal del presupuesto que se entrega al cliente es clarificar las diferentes fases que conforman el desarrollo del proyecto, así como el coste de ejecución asociado a cada una de ellas.

Concepto	Total
Inicio del proyecto	569,12 €
Análisis del sistema	1067,10 €
Implementación	3464,51 €
Desarrollo de las pruebas	505,09 €
Documentación	1259,18 €
Cierre del proyecto	248,99 €
SUBTOTAL	**7113,99 €**
IVA (21%)	1493,94 €
TOTAL	**8607,93 €**

Tabla 4. Presupuesto del cliente

4. ANÁLISIS

4.1 DEFINICIÓN DEL SISTEMA

4.1.1 Determinación del alcance del sistema

El proyecto incluye la creación de un sistema con diferentes interfaces, cada uno de ellos ofrecerá una funcionalidad diferente, y la mínima necesaria, a cada uno de los sistemas externos involucrados, además, para cada uno de ellos se establecerán los controles de acceso adecuados y precisos.

Por una parte, las aplicaciones existentes en la empresa, que actualmente recogen información sobre datos de pago, harán uso de una interfaz destinada a recibir el PAN y generar un token que se proporcionará a la aplicación, de forma que sea este dato el que almacene. Dicho dato será proporcionado por el usuario a través de un formulario sobre el cual se interceptará el evento de entrega para invocar al servidor de tokens y sustituir el valor del PAN por el del token proporcionado. De esta forma las aplicaciones no llegarán a conocer, en ningún momento, el valor del PAN.

Además, el sistema proporcionará una interfaz para que, únicamente las aplicaciones que así lo necesiten, puedan acceder al valor del PAN. Para ello las aplicaciones solicitantes deberán proporcionar su identidad y el valor del token, de esta forma, una vez autenticada la identidad de la petición, el sistema proporcionará el valor del PAN, para que la aplicación cliente pueda operar con él. Cabe destacar que las aplicaciones que obtienen dicho valor, únicamente deberán hacer uso de él y, en ningún caso, almacenar el dato que les ha sido provisto.

Existen también aplicaciones que no necesitan acceder al valor del PAN pero sí que necesiten realizar actividades sobre este valor, para ellas el sistema contará con una interfaz que permita principalmente dos operaciones:

- Eliminación de un token, con esta operación las aplicaciones podrán solicitar al servidor de tokens que elimine toda la información asociada a un token concreto.
- Obtención de PAN enmascarado. En algunas ocasiones las aplicaciones necesitan tener acceso a una fracción del PAN, es decir, al valor se aplica, desde el servidor de tokens, una máscara, antes de ser devuelto a la aplicación cliente.

Por otra parte, el sistema contará con un interfaz de usuario que permitirá al administrador del sistema realizar los casos de uso de mantenimiento que le son asignados, entre los que se encuentra principalmente la gestión de permisos. Es decir, el administrador podrá, a través de este interfaz, controlas que aplicaciones tienen acceso al servidor de tokens y que operaciones son las que le están permitidas.

4.2 REQUISITOS DEL SISTEMA

4.2.1 Requisitos funcionales

A continuación se detallan los requisitos funcionales del proyecto, esto es la descripción de las diferentes actividades que el sistema debe realizar. Se presentan agrupados atendiendo al tipo de usuario que utiliza el sistema.

ADMINISTRADOR

RF1. El sistema permitirá, a los usuarios administradores, acceder a la aplicación a través de autenticación usuario/contraseña.

RF2. La aplicación debe permitir la gestión de las aplicaciones que cuentan con acceso al sistema.

- RF2.1 Se permitirá registrar aplicaciones en el sistema, para ello será necesario facilitar un identificador o nombre.
  - RF2.1.1 No se permitirá registrar dos aplicaciones con el mismo identificador.
- RF2.2 Las aplicaciones podrán estar o no habilitadas, el sistema permitirá modificar dicho valor.
- RF2.3 Se podrán eliminar aplicaciones del sistema, lo que conllevará que esta no pueda solicitar más operaciones al servidor de tokens.
  - RF2.3.1 Esta operación requerirá confirmación por parte del usuario.
  - RF2.3.2 Únicamente será posible en caso de que la aplicación no cuente con tokens activos asociados.
- RF2.4 Será posible acceder a un listado completo de las aplicaciones registradas en el sistema, el cual será ordenable y filtrable.

RF3. El sistema proporcionará la opción de gestionar compañías.

- RF3.1 Existirá la posibilidad de registrar nuevas compañías en el sistema, para ello el usuario deberá proporcionar un nombre o identificador.
  - RF3.1.1 No se permitirá la duplicidad de nombres, es decir, el sistema debe impedir la posibilidad de registrar una compañía cuando ya exista otra previa con el mismo identificador.
- RF3.2 Será posible modificar si una compañía se encuentra o no habilitada en el sistema.
- RF3.3 El sistema contará con la opción de eliminar una compañía.
  - RF3.3.1 La eliminación de una compañía será confirmada por el administrador.
  - RF3.3.2 El sistema validará que la compañía no cuente con tokens activos asignados, solo en este caso será posible eliminar esta.
- RF3.4 Se permitirá el acceso, al usuario administrador, al listado de compañías, sobre el cual se podrán realizar búsquedas y ordenaciones.

RF4. El sistema permitirá que el administrador pueda llevar a cabo la gestión de claves registradas en el sistema.

RF4.1 Se mostrará un listado de las claves que actualmente coexisten en el sistema, en el que se mostrarán los siguientes datos sobre cada una de ellas: alias y estado.

RF4.2 El administrador, en caso de que así lo consideré, podrá seleccionar una para que sea renovada. Ver Rotación de claves.

RF5. El sistema permitirá controlar que entidades tienen acceso al servidor de tokens.

RF5.1 Será posible indicar que compañía, bajo que aplicación, tiene acceso a qué operaciones del sistema.

RF5.2 Se permitirá acceder a un listado de las entidades que tienen acceso al servidor y bajo qué interfaz, en este listado, además, se mostrará la credencial asignada a esa relación.

RF5.3 Un permiso de acceso podrá ser revocado, por el administrador, siempre que este lo desee.

RF5.3.1 Se permitirá deshabilitar un permiso, modificando el estado de este.

RF5.3.2 Será posible eliminar un permiso.

RF6. El sistema permitirá gestionar los tokens almacenados.

RF6.1 Será posible acceder a un listado completo de los tokens que se encuentran en el sistema. En el listado se mostrarán los siguientes datos sobre cada uno de los tokens: aplicación, cliente, valor, fecha de caducidad, fecha de creación y fecha de último uso.

RF6.1.1 Todos los campos permitirán su ordenación

RF6.1.2 Se podrá filtrar el listado por cualquier combinación de los siguientes valores: aplicación, compañía, rango de fechas caducidad, rango de fechas de creación y rango de fechas de último uso.

RF6.2 Existirá una opción para acceder a los tokens menos recientemente usados. El usuario seleccionará una fecha y el sistema mostrará todos aquellos tokens que no hayan sido usados posteriormente a dicha fecha.

RF6.3 El sistema permitirá eliminar tokens, desde cualquiera de los listados anteriores.

RF6.3.1 Será posible seleccionar un único token o varios para su eliminación.

RF6.4 El sistema proporcionará una opción para eliminar todos los tokens caducados, es decir, aquellos cuya fecha de caducidad sea posterior a la actual.

APLICACIONES EXTERNAS

RF7. El sistema permitirá la creación de nuevos tokens.

RF7.1 Para poder generar un nuevo token será necesario proporcionar la siguiente información: ticket generado por el cliente, contraseña generar por el servidor de tokens, credencial que autentifica la petición y el PAN que se desea convertir a token.

RF7.1.1 El ticket estará formado por la credencial que autentifica la petición, la fecha y hora de creación del ticket y un identificador de petición generado desde la aplicación solicitante.

RF7.2 El sistema verificará que la credencial cuenta con los permisos para poder realizar esta operación.

RF7.2.1 La credencial debe estar registrada en el sistema y habilitada.

RF7.2.2 La credencial está asociada a una aplicación y una compañía, ambas deben encontrarse habilitadas.

RF7.2.3 La credencial está asociada a una interfaz, dicha interfaz debe estar asociada a la operación de creación de tokens.

RF7.3 En caso de que la credencial cumpla con todos los requisitos, el sistema almacenará, de forma segura, el PAN proporcionado en la petición y generará un nuevo token, que se proporcionará al peticionario.

RF7.4 El sistema permitirá la creación de tokens temporales. Estos tokens contarán con una fecha de caducidad.

RF7.4.1 La fecha de caducidad de los tokens temporales viene determinada por el periodo de vida de estos, dato que será configurable en el sistema.

RF8. El sistema permitirá obtener el valor enmascarado de un PAN a través de un token.

RF8.1 Para tener acceso a este servicio, el peticionario debe proporciona la credencial que le autentifica, el token y el formato de salida de la máscara.

RF8.2 El sistema verificará que la credencial cuenta con los permisos para poder realizar esta operación.

RF8.2.1 La credencial debe estar registrada en el sistema y habilitada.

RF8.2.2 La credencial está asociada a una aplicación y una compañía, ambas deben encontrarse habilitadas.

RF8.2.3 La credencial está asociada a una interfaz, dicha interfaz debe estar asociada a la operación de obtención de PAN enmascarado.

RF8.3 Se permitirá obtener el enmascarado corto de un PAN, bajo este formado el sistema proporcionará únicamente los últimos dígitos permitidos, establecidos en función de la longitud del PAN. Ver Formatos de máscara.

RF8.4 Se permitirá obtener el enmascarado medio de un PAN, esto es, el sistema devolverá los seis primeros dígitos del PAN.

RF8.5 Se permitirá obtener el enmascarado largo de un PAN, en este caso el sistema proporcionará tanto los primeros seis dígitos como los n últimos permitidos.

RF9. Será posible eliminar un token, para ello será necesario proporcionar la credencial de acceso y el token que se desea eliminar.

RF9.1 El sistema verificará que la credencial cuenta con los permisos para poder realizar esta operación.

RF9.1.1 La credencial debe estar registrada en el sistema y habilitada.

RF9.1.2 La credencial está asociada a una aplicación y una compañía, ambas deben encontrarse habilitadas.

RF9.1.3 La credencial está asociada a una interfaz, dicha interfaz debe estar asociada a la operación de eliminación de tokens.

RF10. El sistema permitirá obtener el valor del PAN asociado a un token. Esta operación requerirá que se proporcione la credencial de autenticación y el token asociado al PAN que se desea obtener.

RF10.1 El sistema verificará que la credencial cuenta con los permisos para poder realizar esta operación.

RF10.1.1 La credencial debe estar registrada en el sistema y habilitada.

RF10.1.2 La credencial está asociada a una aplicación y una compañía, ambas deben encontrarse habilitadas.

RF10.1.3 La credencial está asociada a una interfaz, dicha interfaz debe estar asociada a la operación de obtención de PAN.

OTROS

RF11. El sistema permitirá renovar la clave de encriptación activa.

RF11.1 El proceso debe ejecutarse de manera periódica y automática.

RF11.1.1 El periodo de ejecución de este proceso será configurable.

RF11.2 El proceso de renovación generará y registrará una nueva clave de encriptación en el sistema.

RF11.2.1 Las peticiones de creación de un nuevo PAN se llevarán a cabo utilizando esta nueva clave.

RF11.3 El proceso eliminará la clave que haya llegado al final de su criptoperiodo.

RF11.3.1 Todos los tokens que fuesen encriptados con la clave a eliminar deben ser reencriptados con la nueva clave generada.

RF12. El sistema debe eliminar, de forma automática, los tokens temporales periódicamente.

RF12.1 Únicamente se eliminarán aquellos tokens caducados, es decir, los que hayan alcanzado su fecha de caducidad en el momento de realizar el borrado.

RF12.2 El periodo de ejecución podrá ser establecido por configuración.

RF13. El sistema eliminará, periódica y automáticamente, las contraseñas de acceso cuyo tiempo de vigencia haya concluido.

RF13.1 El periodo de ejecución podrá ser establecido por configuración.

RF14. El sistema contará con un proceso que se ejecute automáticamente y elimine los tickets una vez haya concluido su periodo de validez.

RF14.1 El proceso se ejecutará periódicamente, dicho periodo podrá ser establecido por configuración.

4.2.2 Requisitos no funcionales

Los requisitos no funcionales nacen de las necesidades del usuario, se trata de requisitos que no están relacionados directamente con la funcionalidad del sistema sino que definen cualidades de este.

RNF1. Las sesiones sin actividad deben caducar transcurridos quince minutos.

RNF2. La aplicación debe guardar log de las operaciones que se realizan sobre el servidor de tokens.

RNF2.1 Todos los registros de log incluirán la fecha y hora en la que se registró el evento.

RNF2.2 Se incluirá un registro de log por cada inicio y cierre de sesión de un usuario administrador.

RNF2.3 Se incluirá un registro de log en cada alta, baja y modificación de una aplicación o compañía en el sistema.

RNF2.4 Se incluirá un registro de log en cada arranque y parada de la aplicación.

RNF2.5 Se incluirá un registro de log en cada ejecución del proceso de renovación de clave de encriptación.

RNF2.6 Se incluirá un registro de log en cada ejecución del proceso de limpieza de tokens temporales.

RNF2.7 Se incluirá un registro de log en todas aquellas operaciones que involucren un PAN: creación de tokens y borrado de tokens, obtención de PAN enmascarado, obtención de PAN a través de un token, etc.

RNF2.8 Los log deberán almacenarse en un fichero local y, además, serán enviados a un servidor de tipo Syslog.

RNF3. Debe existir un mecanismo para distinguir un token de un PAN normal.

4.3 CICLO DE VIDA DE UN TOKEN

El sistema permitirá la creación de tokens temporales o permanentes, esto vendrá determinado por el valor del campo en el que se almacena la fecha de validez del token. En caso de que el campo esté vacío se entenderá que se trata de un token permanente y, en el caso contrario, de un token temporal.

- Los tokens permanentes solo serán borrados por expresa indicación de la aplicación backend que lo crea o por acción del administrador del sistema.
- Los tokens temporales, por el contrario, serán eliminados automáticamente por el sistema durante la ejecución de un proceso periódico de eliminación de tokens extinguidos que se ejecutará con frecuencia de minutos. Este valor de refresco es especificado por configuración.

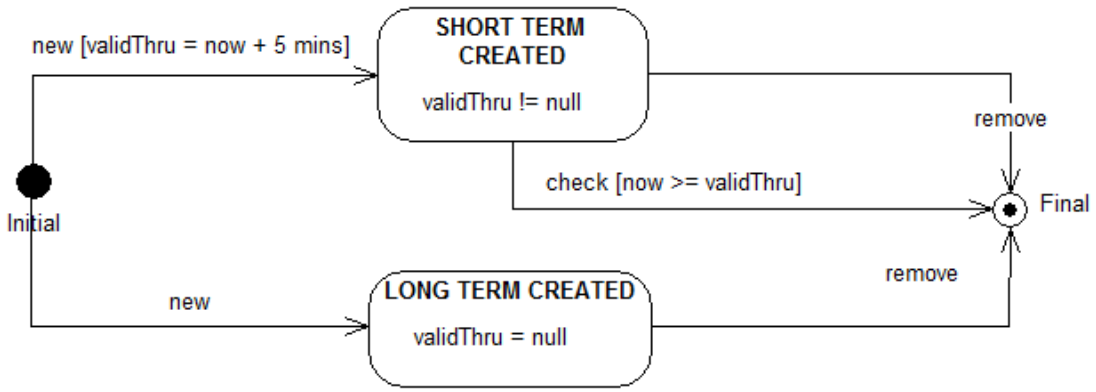

Figura 12. Ciclo de vida de un token

4.4 ROTACIÓN DE CLAVES

La normativa PCI (req. 3.6.4) indica que es necesario cambiar las claves de encriptación cuando llegan al final de su *criptoperiodo*. Asimismo, indica que se sigan las mejores prácticas recomendadas por la industria y cita la normativa NIST 800-57 como ejemplo.

Se propone una política de rotación de claves que se ejecute de forma automática en un proceso lanzado periódicamente por un timer o de forma puntual y específica por el administrador.

El concepto fundamente es el *criptoperiodo:* periodo de tiempo en el que la clave puede ser usada para encriptar y/o desencriptar. Se subdivide a su vez en dos periodos:

- Periodo de encriptación, en la literatura en inglés OUP (Originator Usage Period), es decir, rango de fechas en el que la clave puede ser usada para encriptación.
- Periodo de desencriptación, en inglés RUP (Recipient Usage Period), el tiempo durante el cual la calve se puede usar para desencriptar. Es igual o más largo que el periodo anterior y también configurable en el sistema.

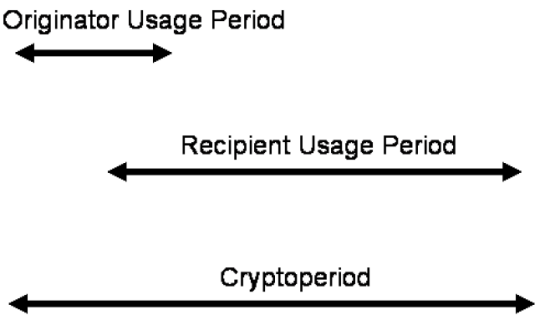

Figura 13. Criptoperiodo general

Jugando con los solapamientos de OUP y de RUP se pueden conseguir políticas distintas. A los efectos del sistema propuesto se entienden los criptoperiodos como se muestra en la figura siguiente.

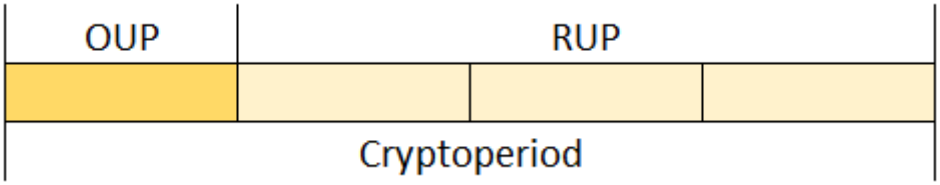

Figura 14. Criptoperiodo en el servidor de tokens

En función del volumen de información protegida, el cambio de clave puede llegar a ser problemático ya que obliga a procesarlo todo de nuevo. Para reducir el esfuerzo computacional necesario, las recomendaciones indican indicar usar varias claves con periodos de desencriptación mayor que el de encriptación.

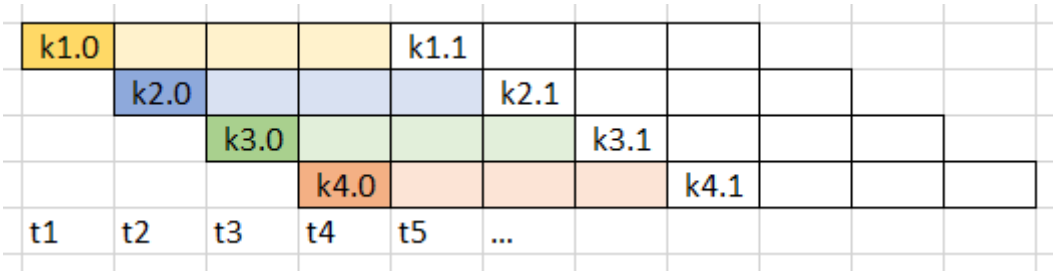

Figura 15. Criptoperiodo de 4 meses con un 1 mes de encriptación

Al ser superior el periodo de desencriptación al de encriptación, esto hace que en el sistema coexistan varias claves a la vez, aunque solo una es usada para la encriptación en cada momento. Para el proceso de desencriptación el sistema debe tener en cuenta con cual clave fue encriptado. La figura anterior muestra la evolución a lo largo del tiempo del uso de las claves. Así, en t1 la clave activa es k1.0, en t2 será la k2.0, etc.

Según va finalizando el criptoperiodo completo de cada clave es necesario generar una nueva y reencriptar con ella los tokens que aún permanecen en el sistema con la que se extingue, esto ocurre en t5 en la figura anterior.

Con este esquema solo es necesario reencriptar una fracción de todas las PAN en el sistema. Además, ofrece la ventaja de que, en caso de compromiso de una clave, el daño queda contenido a los PAN afectados por esa clave.

Es condición que la duración total del criptoperiodo sea múltiplo de la duración del periodo de encriptación. En el ejemplo de la figura 4 a 1. Además, en esas circunstancias, el número de claves mantenidas en el sistema es el cociente de la duración del periodo total entre el periodo de encriptación. En el ejemplo 4/1 = 4.

CICLO DE VIDA DE LAS CLAVES DE ENCRIPTACIÓN

El ciclo de vida que se propone para las claves del sistema es una adaptación del esquema general que aparece en NIST 800-57p1, apartado 7 Key States and Transitions.

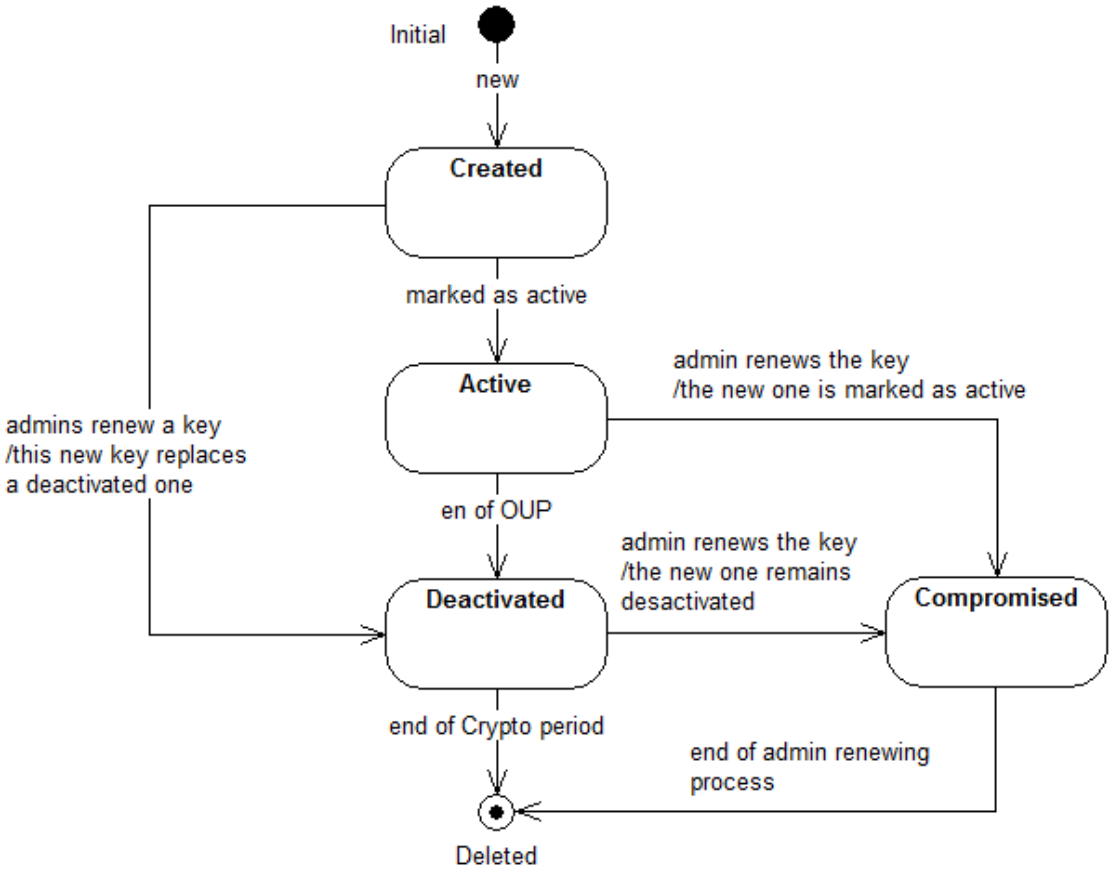

Figura 16. Ciclo de vida de la clave de encriptación

Las claves empleadas para la protección del PAN serán usadas por algoritmos de encriptación simétrica y estarán custodiadas en un almacén de claves.

El diagrama propuesto representa los estados conceptuales en los que podrá estar una clave contemplada desde el punto de vida del servidor de tokens. Los posibles estados se definen así:

- Created. La clave está añadida en el almacén de claves, pero sin ningún uso, presente ni pasado, dese el servidor de tokens. En este estado aún no ha entrado en su criptoperiodo.
- Active. La clave ha entrado en su criptoperiodo y ya está siendo usada para encriptar y desencriptar PAN. Solo puede haber una clave en el sistema en este estado.
- Deactivated. La clave ha terminado su periodo de encriptación y sigue siendo usada para desencriptar los PAN que fueron almacenados mientras estaba en estado active. Puede haber varias claves en este estado.

- Compromised. Se sospecha o verifica que la clave puede haber sido comprometida. Este es un estado temporal en que se activará manualmente el proceso de renovación de claves. Durante este se generará una nueva clave que se usará para reencriptar todos los PAN protegidos por la anterior clave comprometida. La nueva generada sustituirá en el sistema a la anterior.
- Deleted. Una vez la clave ha terminado su criptoperiodo y el proceso de renovación de clave ha sido ejecutado la clave se elimina del sistema.

Las transiciones contempladas son:

- Initial → Created, cuando se crea una nueva calve y es añadida al almacén de claves.
- Created → Active, cuando la clave es usada por primera vez para encriptar un PAN.
- Activa → Deactivated, cuando la clave ya no es usada para encriptar ya que otra ha pasado al estado active. Solo se emplea para desencriptar lo que ha sido encriptado mientras estaba activa.
- Deactivated → Deleted, tras la ejecución del proceso automático de renovación de clave, si esta clave ha llegado al final de su criptoperiodo, lo que implica que ya no hay PAN registrados que estén encriptados con ella, es eliminada del almacén de claves.
- Active → Compromised, cuando se activa manualmente, por parte del administrador, el proceso de renovación de clave.
- Deactivated → Compromised, se da la transición en las mismas circunstancias anteriores.
- Created → Deactivated, la clave es creada para sustituir a otra que previamente estaba en estado deactivated por el proceso manual de regeneración de calve.
- Compromised → Deleted, al finalizar el proceso manual de renovación de clave, esta se borra del almacén de claves.

PROCESO DE RENOVACIÓN DE CLAVE

Se distinguen dos procesos:

- El programado y automático, que se ejecutará cada cierto tiempo.
- El manual, lanzado por el administrador, quizá ante sospecha de compromiso.

El programado y automático es ejecutado cada vez que un timer del sistema lo invoque. Su misión es sustituir la clave activa. Durante su ejecución se efectuarán los siguientes pasos:

- Se generará una nueva clave, se almacenará en el almacén de claves y se establecerá como la activa en el sistema. A partir de este momento, todas las nuevas peticiones de tokens se realizarán con ésta.
- Se buscarán todos los tokens que permanecen en el sistema con la clave deactivated que ha llegado al final de su criptoperiodo. Todos y cada uno de ellos se reencriptarán con la nueva clave activa.
- La clave que ha llegado el final de su criptoperiodo se elimina del almacén de claves.

El proceso manual invocado por el administrador permitirá seleccionar la clave que se desea renovar. Para ella:

- Se generará una nueva clave con los mismos periodos y estado que la clave que se desea sustituir. Si es la activa, la nueva tendrá el mismo criptoperiodo que la anterior y será la nueva activa. Si es una desactivada, la nueva estará también desactivada y mantendrá el cryptoperiodo.
- Se reencriptarán todos los tokens guardados bajo la clave a sustituir.
- Se eliminará del almacén de claves la que acaba de ser sustituida.

4.5 SEGURIDAD

La especificación PCI para servidor de tokens indica como práctica recomendada usar control de autorización basado en roles y cita como ejemplo la especificación ANSI INCITS 359. La solución que se propone aquí es una simplificación del caso general propuesto en el documento citado.

Se propone un modelo conceptual de seguridad consistente en asignar permisos para ejecutar determinadas operaciones a la combinación de aplicación y compañía cuando acceden por una interfaz concreta. Es decir, permite especificar qué operaciones se pueden ejecutar desde cada interfaz y por qué compañía-aplicación.

Cada interfaz tiene un rol implícito que define el conjunto de operaciones que permite, y la credencial conecta cada usuario con su rol. Cada interfaz tiene predefinidas una serie de operaciones que permitirá realizar. Así la interfaz:

- UIFormApp permite realizar las operaciones:
  - Creación de token temporal.
  - Creación de token permanente.
- BackendApp permite:
  - Borrar token poseído.
  - Obtención de PAN enmascarado.
- PCIApp permite:
  - Obtención de PAN dese token.
- Monitor:
  - Consulta de estado

El administrador podrá manipular los estados de las entidades mencionadas. Así, cambiando el estado de la entidad precisa, podrá regular el acceso en tiempo de explotación con la granularidad adecuada.

4.6 FORMATOS DE MÁSCARA

Existen recomendaciones de formatos de máscara que deben ser observadas.

Acceptable PAN Truncation Formats					
PAN Length	*American Express*	*Discover*	*Mastercard*	*JCB*	*Visa*
> 16 digit PAN with 6 digit BIN	N/A	N/A	At least 6 digits removed. Maximum digits which may be retained: 17-digit PAN: "First 6, any other 5" 18-digit PAN: "First 6, any other 6" 19-digit PAN: "First 6, any other 7"	N/A	At least 6 digits removed. Maximum digits which may be retained: 17-digit PAN: "First 6, any other 5" 18-digit PAN: "First 6, any other 6" 19-digit PAN: "First 6, any other 7"
16 digit PAN with 8 digit BIN	N/A	At least 6 digits removed. Maximum digits which may be retained: "First 6, any other 4"	N/A	N/A	At least 6 digits removed. Maximum digits which may be retained: "First 6, any other 4"
16 digit PAN with 6 digit BIN	N/A	At least 6 digits removed. Maximum digits which may be retained: "First 6, any other 4"			
15 digit PAN	At least 5 digits removed. Maximum digits which may be retained: "First 6, last 4""	N/A	At least 5 digits removed. Maximum digits which may be retained: "First 6, any other 4".	N/A	N/A
< 15 digit PAN	N/A	Maximum digits which may be retained: "First 6, any other 4"		N/A	N/A

Figura 17. Máscaras permitidas

La tabla anterior puede ser resumida en:

- Si el PAN tiene más de 16 dígitos: se pueden mostrar los 6 primeros y los (lengt(PAN) – 12) últimos dígitos.
- Si el PAN tiene 16 o menos dígitos, se pueden mostrar los 6 primeros y los 4 últimos.

Como simplificación se propone soportar tres formatos de máscara:

- Corto: devuelve los últimos dígitos permitidos.
- Medio: devuelve los seis primeros dígitos.

Largo: devuelve los seis primeros dígitos y los últimos permitidos.

4.7 IDENTIFICACIÓN DE LOS SUBSISTEMAS

El desarrollo del servidor de tokens conlleva el desarrollo de dos subsistemas, así como la modificación de un tercero.

ENCRIPTADOR

Este subsistema tiene como función principal la gestión del proceso de encriptación que el servidor debe seguir para almacenar de forma segura los PAN. Esto incluye dos procesos principales:

- Gestión de las claves de encriptación utilizadas.
- Proceso de encriptación y desencriptación de los PAN.

Este componente debe ser capaz de gestionar todos los aspectos de cifrado, de forma que le sean totalmente transparente al servidor de tokens.

SERVIDOR DE TOKENS

Se trata del sistema principal y será quien se encargue del control de acceso, así como de la creación y gestión de los tokens que sustituyen a los PAN.

APLICACIONES CLIENTE

Las aplicaciones que actualmente trabajan con información de pago relativa a tarjetas de crédito deberán ser modificadas para comenzar a utilizar el servidor de tokens y dejar de almacenar los PAN. Este subsistema por tanto, forma parte del contexto del proyecto, aunque no directamente del desarrollo de este.

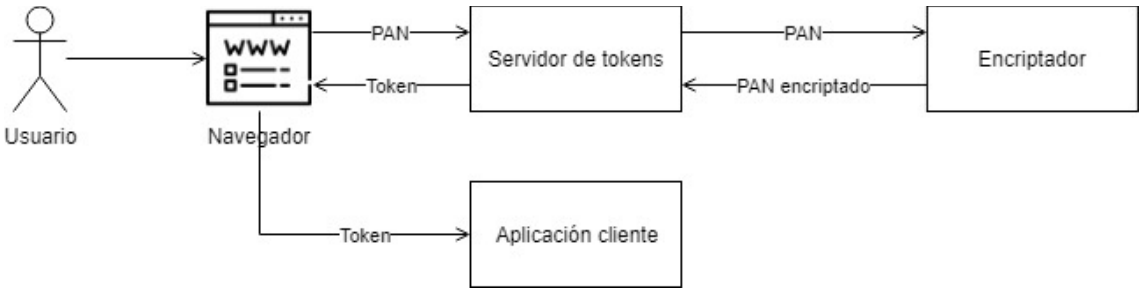

Figura 18. Subsistemas.

4.8 MODELO DE DOMINIO

4.8.1 Diagrama de modelo de dominio

A continuación se muestra el análisis inicial del modelo de dominio.

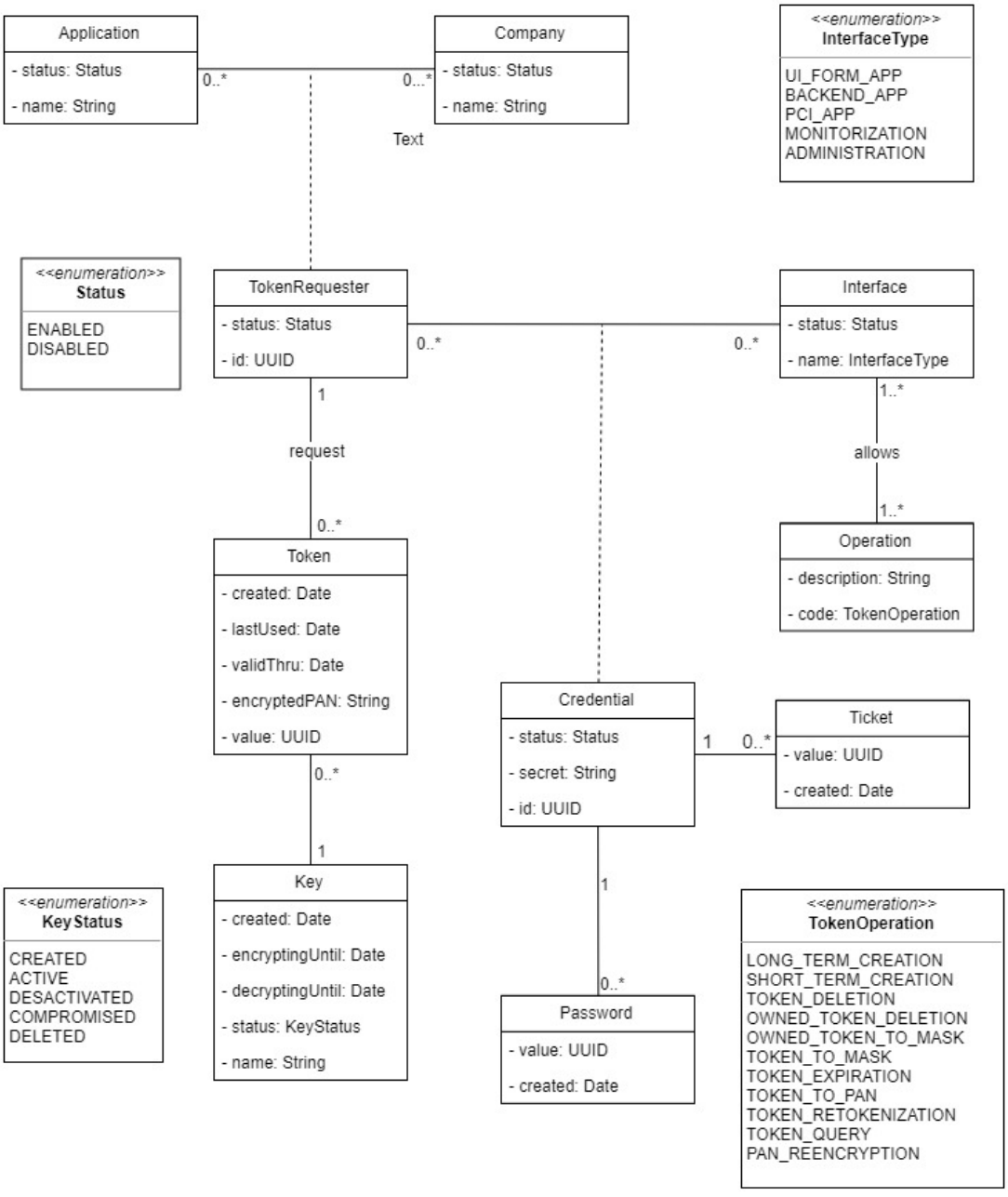

Figura 19. Modelo de dominio

4.8.2 Descripción de las entidades

El propósito de la descripción que se muestra a continuación, es intentar clarificar el significado de cada uno de los elementos descritos anteriormente.

APPLICATION

Representa a cada una de las instancias operativas de las aplicaciones que Iris tenga en producción en cada momento. Todas las aplicaciones que accedan al token server estarán representadas por una entidad de esta clase.

De cada una se registrará:

- Un nombre distintivo, que actuará como identidad de esta entidad, y
- Estado, que indicará si la entidad está habilitada o deshabilitada. En caso de estar deshabilitada se detendrá toda actividad pedido por esa aplicación.

COMPANY

Representa cada uno de los posibles clientes que operan en alguna de las aplicaciones.
De cada una se registrará:

- Un nombre distintivo que actuara como identidad de esta entidad.
- Su estado, que indicará si la entidad está habilitada o deshabilitada. En caso de estar deshabilitada se detendrá toda actividad pedida por esta compañía.

TOKEN

Representa una pareja token/PAN, de forma que el primero sustituirá al segundo en las aplicaciones.

No hay restricción en el número de veces que un mismo PAN pueda estar representado por tokens. Es decir, a un PAN se le asignan tantos tokens diferentes como veces se registre en el sistema, lo que debe ser único es la combinación token-PAN. Esto permite resolver escenarios en los que la misma tarjeta esté compartida por varios empleados o departamentos de la misma compañía.

De un token se almacenará:

- Su UUID, es un string que recibirán las aplicaciones backend. Este campo hará de identidad de esta entidad.
- La fecha de creación, es asignada automáticamente por el sistema y permitirá hacer filtrados.
- La fecha de último uso, actualizada cada vez que sea necesario desencriptar el PAN.
- La fecha de validez. Indicará el momento en el que el token deja de ser válido, a partir del cual puede ser eliminado del sistema.
- El PAN encriptado de acuerdo a los requisitos no funcionales de fortaleza criptográfica.

TOKENREQUESTER

Representa la reificación de la relación entre compañía y aplicación. Algo así como “esta compañía operando en esta aplicación”. La aparición de esta entidad artificial permite discernir la propiedad de los tokens que se crean y sirve también como unidad de control de permisos para algunas de las operaciones que se apliquen a los tokens creados.

De cada una se registrará:

- Un UUID que actuará como identidad de esta entidad.
- Su estado, que indicará si la entidad está habilitada o deshabilitada. En caso de estar deshabilitada se detendrá toda actividad pedida por la compañía para la aplicación.

KEY

Esta entidad representa en el sistema el contenido del almacén de claves. La especificación de campos es como sigue:

- Nombre, que guarda un alias bajo el cual la clave está guardada en el almacén.
- La fecha de creación, que representa la fecha de creación de la clave y el inicio del criptoperiodo.
- La fecha de finalización del periodo de encriptación, que representa el instante en el tiempo a partir del cual la clave ya no puede ser marcada como desactivada.
- La fecha de fin del criptoperiodo, empleado por el proceso de rotación de claves para decidir si se borra o no la clave.
- Estado, que indica el estado de la clave.

INTERFACE

Representa cada uno de los accesos que permite la aplicación.

- La interfaz estará identificada por su tipo, el cual será un enumerado cuyos posibles valores son fijos.
- El campo estado permite bloquear el acceso a través de esta interfaz.

CREDENTIAL

Representa el hecho de que un TokenRequester opera en una interfaz.

- Su campo estado permite bloquear el acceso de un TokenRequester a una interfaz concreta.
- Secreto, es un campo que almacena la clave de encriptación empleada para el intercambio de credenciales entre las aplicaciones y el servidor de tokens.

OPERATION

Representa cada una de las operaciones que se pueden ejecutar en el sistema desde cualquier interfaz que no sea las de administración. Son valores fijos determinados por el diseño y, por tanto, no se necesita mantenimiento. Las operaciones controladas son:

- Creación de token temporal.
- Creación de token permanente.
- Borrado de token poseído.
- Obtención de PAN enmascarado.
- Obtención de PAN desde token.
- Consulta de estado

PASSWORD

Representa la clave que el servidor de tokens proporcionada a cada aplicación cliente y que le habilita para solicitar la creación de un token durante un periodo de tiempo determinado.
De cada una se registrará:

- Valor que identifica la entidad y que es proporcionado a la aplicación backend.
- Fecha de creación de la entidad, que se utiliza para calcular el periodo de validez.

TICKET

Representa el identificador de petición que las aplicaciones backend proporcionan al servidor de tokens para evitar la duplicidad de peticiones.
De cada una se registrará:

- Valor que identifica la entidad y que es proporcionado desde la aplicación backend.
- Fecha de creación de la entidad, que se utiliza para calcular su periodo de validez.

4.9 ANÁLISIS DE CASOS DE USO Y ESCENARIOS

4.9.1 Identificación de actores del sistema

A continuación se definen los diferentes roles con los que es posible acceder al servidor de tokens, estos roles están asociados a diferentes derechos de uso, proporcionan accesos diferentes dentro de la aplicación.

ADMINISTRADOR

El usuario administrador accederá al servidor de token a través de la interfaz de usuario que esté proporciona y, haciendo uso de esta, podrá llevar a cabo las tareas de gestión y administración que le son asignadas.

UIFORMAPP

Bajo este rol se encuentran las aplicaciones que cuentan con un formulario para que el usuario proporcione los datos de pago, entre los que se encuentra el PAN. Al envío de este formulario las aplicaciones solicitarán al token server que genere un nuevo token a partir del PAN, para sustituir el valor por el del token.

Quienes acceden al sistema bajo este rol únicamente tienen acceso al sistema para crear tokens que sean o no temporales.

BACKENDAPP

Las aplicaciones que utilizan el sistema para sustituir los PAN con los que trabajan por tokens, utilizan este rol para solicitarle al sistema que realice ciertas operaciones sobre el PAN sin llegar a tener, en ningún momento, acceso a este dato. Este actor únicamente cuenta con permisos para eliminar un token del servidor, por motivos de saneamiento normalmente, y para obtener el valor del PAN enmascarado en diferentes formatos para, por ejemplo, mostrar este dato a sus usuarios.

PCIAPP

Este es el único rol del sistema que permite acceder al número PAN almacenado en el servidor de tokens, por lo tanto, únicamente podrán acceder desde este tipo de usuario aquellos que accedan desde una zona de red que cumpla con los requisitos de seguridad establecidos por PCI.

KEYROTATION TIMER

Se trata de un actor que cumple la función de realizar, periódicamente, la rotación de la clave de encriptación activa en el sistema.

CLEANING TIMER

Este actor se encarga de eliminar del sistema aquellas entidades cuyo ciclo de vida ha finalizado. Se encarga de una triple función: limpieza de tokens temporales, contraseñas de acceso y tickets.

MONITOR

Su función es la de invocar regularmente al sistema, ejecutando una acción concreta de este, para comprobar que funciona correctamente.

4.9.2 Diagrama de casos de uso

A continuación se analizan las interacciones que los diferentes actores descritos anteriormente tienen con la aplicación, de forma que se clarifique el uso de cada uno hace del sistema.

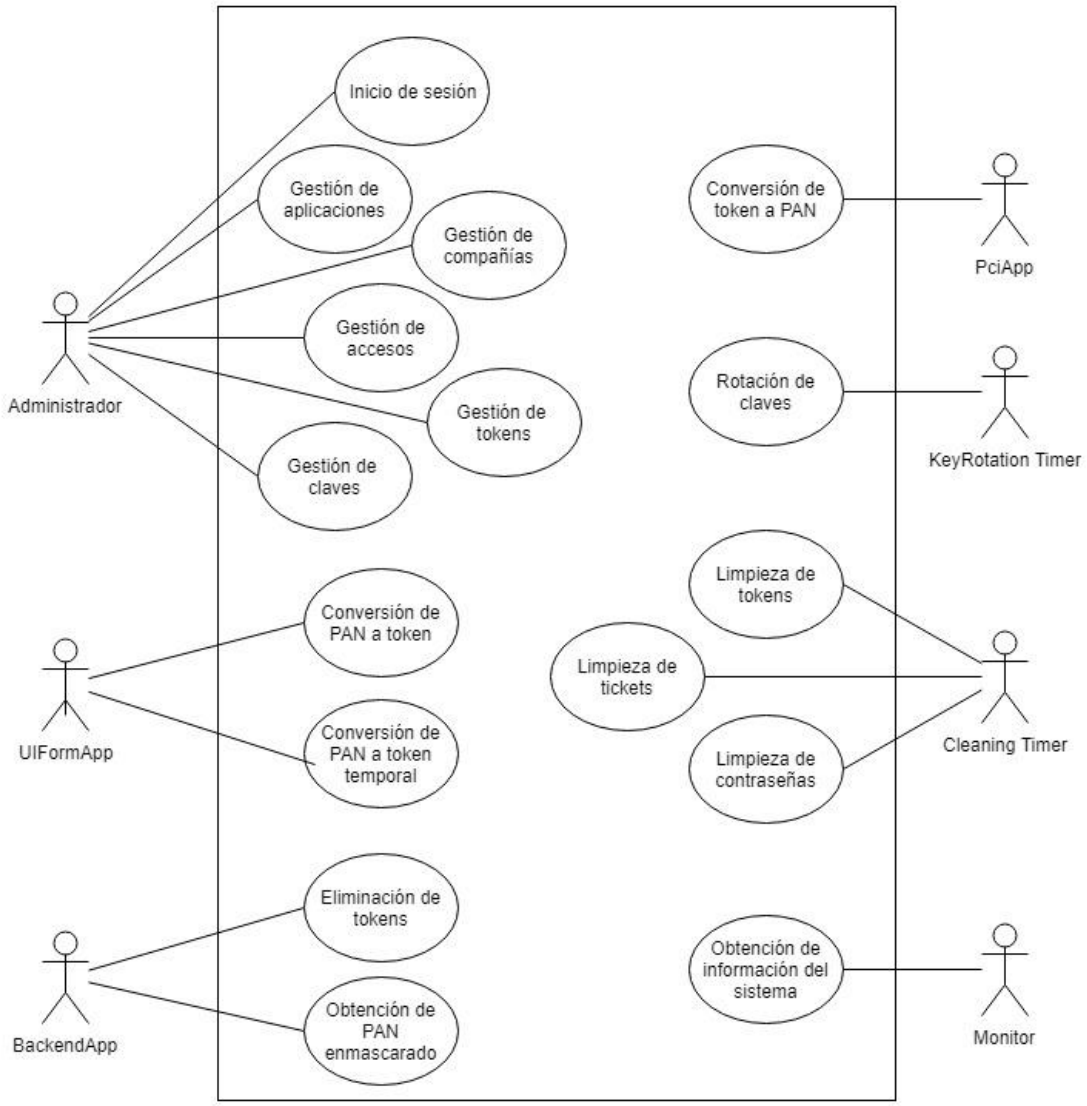

Figura 20. Diagrama de casos de uso

4.9.3 Especificación de casos de uso

A continuación se realiza una descripción pormenorizada de los escenarios derivados de cada uno de los casos de uso de principales que se incluyen en el anterior diagrama.

ACCESO A LA INTERFAZ DE ADMINISTRACIÓN

ID 1- INICIO DE SESIÓN

Precondiciones	-
Poscondiciones	El usuario administrador habrá accedido a la aplicación y tendrá acceso a las operaciones de administración.
Actores	Administrador
Descripción	El usuario administrador accederá a la interfaz web que da acceso a las tareas de administración, donde se mostrará un formulario para proporcionar el usuario y contraseña que dan acceso a la aplicación. Una vez los datos han sido validados, el sistema le presenta las opciones de administración disponibles.
Excepciones	En caso de que el usuario proporcione unos datos que no se corresponde con ninguno de los usuarios con permiso para acceder al sistema, este mostrará un mensaje de error e impedirá que el usuario pueda acceder al resto del sistema.

Tabla 5. Caso de uso de inicio de sesión

GESTIÓN DE APLICACIONES

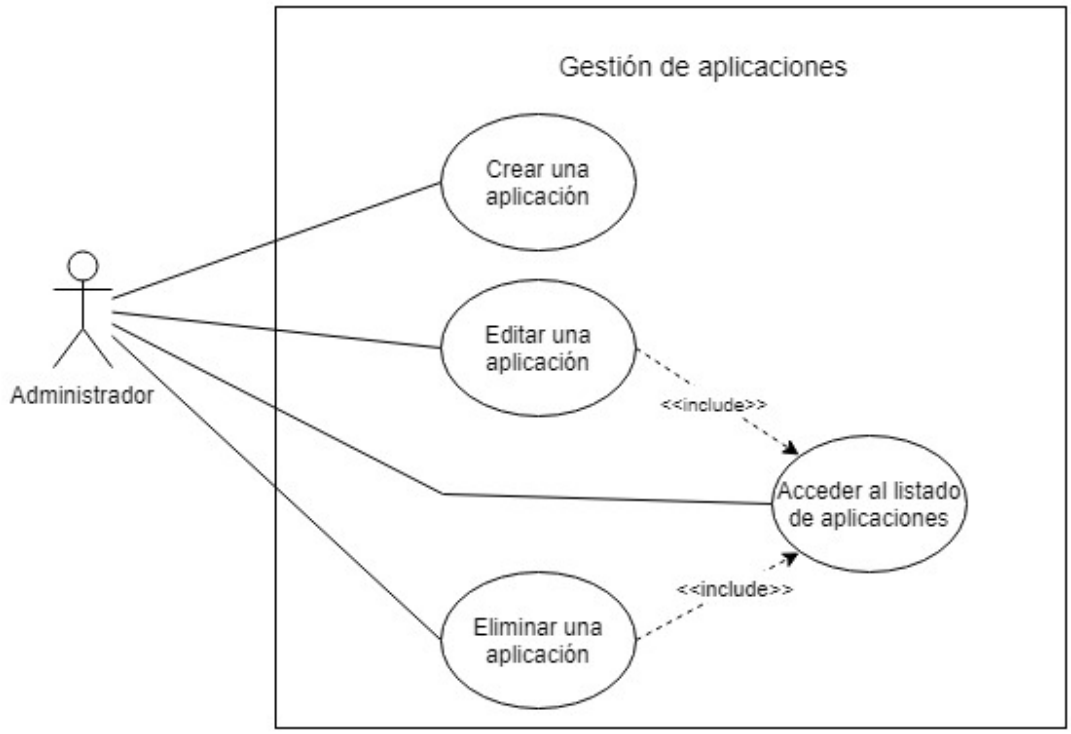

Figura 21. Caso de uso gestión de aplicaciones

ID2.1 – CREAR UNA APLICACIÓN

Precondiciones	El usuario administrador debe haber iniciado sesión satisfactoriamente en el sistema.
Poscondiciones	Una nueva aplicación se encuentra registrada en el sistema.
Actores	Administrador
Descripción	El usuario administrador proporcionará al sistema el nombre con el cual se identificará a la aplicación, así como si esta se encuentra o no habilitada.
Variaciones (escenarios secundarios)	En caso de que se proporcione un nombre de aplicación que ya se encuentra registrado en el sistema, esta no podrá ser registrada y se mostrará un error al usuario.

Tabla 6. Caso de uso de creación de una aplicación

ID2.2 – ACCEDER AL LISTADO DE APLICACIONES

Precondiciones	El usuario administrador debe haber iniciado sesión satisfactoriamente en el sistema.
Poscondiciones	El usuario puede acceder al listado completo de aplicaciones registradas en el sistema.
Actores	Administrador
Descripción	En caso de que el usuario seleccione esta opción en el menú, se mostrará un listado completo de las aplicaciones registradas en el sistema. Dicho listado mostrará tanto el nombre de la aplicación como su estado. Además, se permitirá que el usuario pueda filtrar las aplicaciones a partir de cualquier de los datos que aparecen sobre estas.
Excepciones	En caso de que no exista ninguna aplicación registrada en el sistema, este mostrará un mensaje informativo que así lo indique.

Tabla 7. Caso de uso de acceso al listado de aplicaciones

ID2.3 – EDITAR UNA APLICACIÓN

Precondiciones	El usuario administrador debe haber iniciado sesión satisfactoriamente en el sistema.
Poscondiciones	El estado de una aplicación ha sido modificado.
Actores	Administrador
Descripción	El usuario administrador seleccionará, entre la lista completa, aquella aplicación que desea modificar. Una vez identificada, el usuario seleccionará el nuevo estado de la aplicación y este será modificado mostrando un mensaje que así lo confirme.

Tabla 8. Caso de uso de edición de una aplicación

ID2.4 – ELIMINAR UNA APLICACIÓN

Precondiciones	El usuario administrador debe haber iniciado sesión satisfactoriamente en el sistema.
Poscondiciones	Una aplicación habrá sido eliminada del sistema y, por lo tanto, también habrán sido eliminados todos sus accesos al servidor de tokens.
Actores	Administrador

Descripción	El usuario seleccionará, entre la lista completa, aquella aplicación que desea eliminar. Una vez identificada, el sistema solicitará que el usuario confirme la acción y, en caso afirmativo, esta será eliminada del sistema.
Variaciones (escenarios secundarios)	En caso de que la aplicación cuente con tokens activos asociados, esta no podrá ser eliminada del sistema, que debe informar de dicho suceso a través de un mensaje de error.

Tabla 9. Caso de uso de eliminación de una aplicación

GESTIÓN DE COMPAÑÍAS

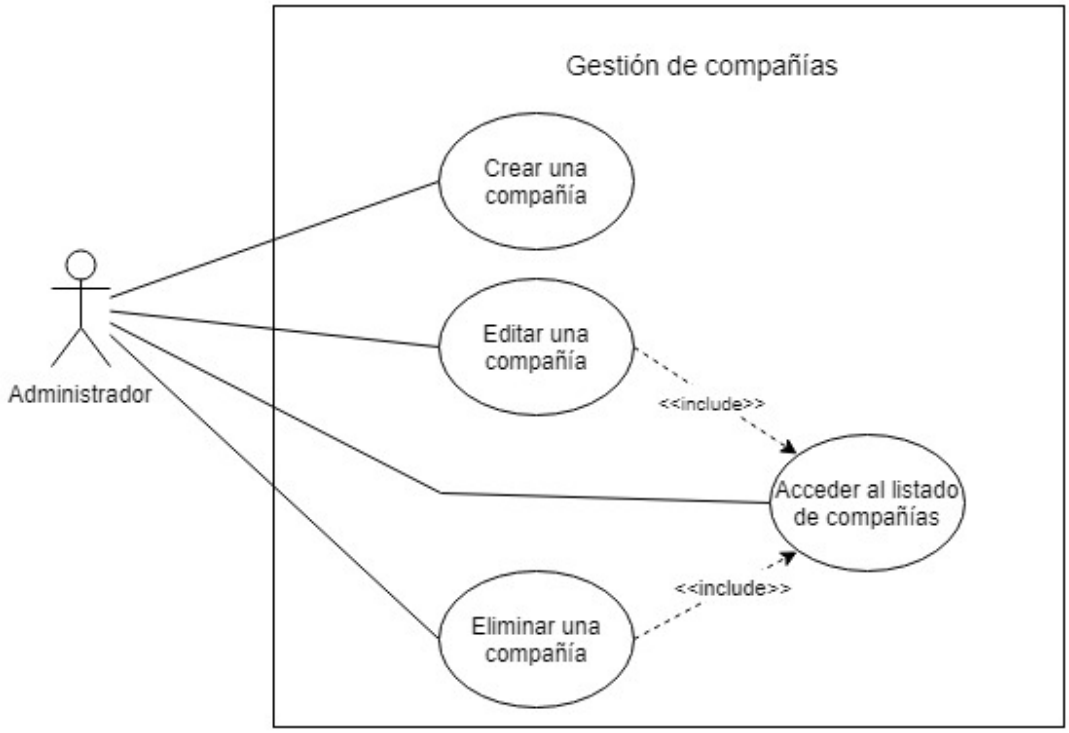

Figura 22. Caso de uso gestión de compañías

ID3.1 – CREAR UNA COMPAÑÍA

Precondiciones	El usuario administrador debe haber iniciado sesión satisfactoriamente en el sistema.
Poscondiciones	Una nueva compañía se encuentra registrada en el sistema.
Actores	Administrador
Descripción	El usuario administrador proporcionará al sistema el nombre con el cual se identificará a la compañía, así como si esta se encuentra o no habilitada.
Variaciones (escenarios secundarios)	En caso de que se proporcione un nombre de compañía que ya se encuentra registrado en el sistema, esta no podrá ser registrada y se mostrará un error al usuario.

Tabla 10. Caso de uso de creación de una compañía

ID3.2 – ACCEDER AL LISTADO DE COMPAÑÍAS

Precondiciones	El usuario administrador debe haber iniciado sesión satisfactoriamente en el sistema.
Poscondiciones	El usuario puede acceder al listado completo de compañías registradas en el sistema.
Actores	Administrador
Descripción	En caso de que el usuario seleccione esta opción en el menú, se mostrará un listado completo de las compañías registradas en el sistema. Dicho listado mostrará tanto el nombre de la compañía como su estado. Además, se permitirá que el usuario pueda filtrar las compañías a partir de cualquier de los datos que aparecen sobre estas.
Excepciones	En caso de que no se haya creado aún ninguna compañía, el sistema mostrará un mensaje informativo que así lo indique.

Tabla 11. Caso de uso de acceso al listado de compañías

ID3.3 – EDITAR UNA COMPAÑÍA

Precondiciones	El usuario administrador debe haber iniciado sesión satisfactoriamente en el sistema.
Poscondiciones	El estado de una compañía ha sido modificado.
Actores	Administrador
Descripción	El usuario administrador seleccionará, entre la lista completa, aquella compañía que desea modificar. Una vez identificada, el usuario seleccionará el nuevo estado de la compañía y este será modificado mostrando un mensaje que así lo confirme.

Tabla 12. Caso de uso de edición de una compañía

ID3.4 – ELIMINAR UNA COMPAÑÍA

Precondiciones	El usuario administrador debe haber iniciado sesión satisfactoriamente en el sistema.
Poscondiciones	Una compañía habrá sido eliminada del sistema y, por lo tanto, también habrán sido eliminados todos sus accesos al servidor de tokens.
Actores	Administrador
Descripción	El usuario seleccionará, entre la lista completa, aquella compañía que desea eliminar. Una vez identificada, el sistema solicitará que el usuario confirme la acción y, en caso afirmativo, esta será eliminada del sistema.
Variaciones (escenarios secundarios)	En caso de que la compañía cuente con tokens activos asociados, esta no podrá ser eliminada del sistema, que debe informar de dicho suceso a través de un mensaje de error.

Tabla 13. Caso de uso de eliminación de una compañía

GESTIÓN DE ACCESOS

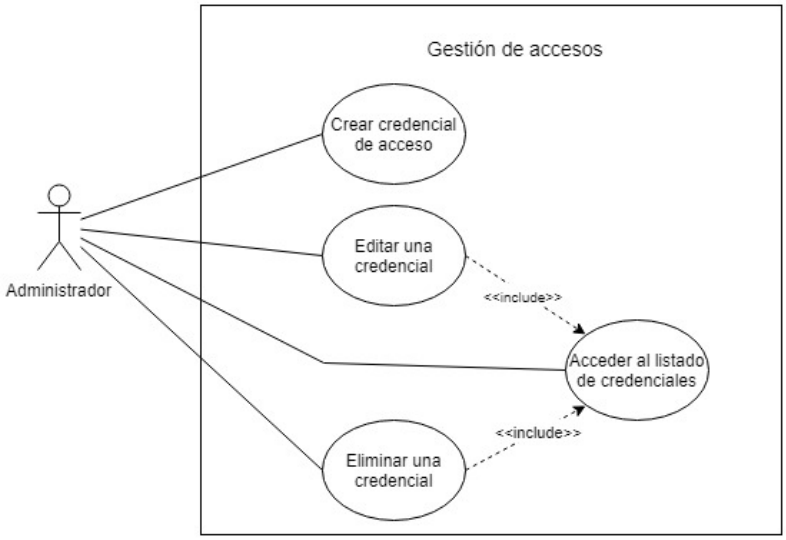

Figura 23. Caso de uso gestión de accesos

ID4.1 – CREAR UNA CREDENCIAL DE ACCESO

Precondiciones	El usuario administrador debe haber iniciado sesión satisfactoriamente en el sistema.
Poscondiciones	Se ha creado una o varias credenciales de acceso al sistema nuevas.
Actores	Administrador
Descripción	El usuario administrador seleccionará, de entre la lista de aplicaciones disponibles, una de ellas. A continuación, hará lo mismo con la compañía y, por último, seleccionará una o varias interfaces a través de las cuales quiere proporcionar acceso para la aplicación y compañía seleccionada. Una vez que el usuario confirme la acción, el sistema generará un nueva credencial para cada interfaz seleccionada por el usuario y se mostrarán los datos asociados a dichas credenciales, tanto el identificador como el secreto de cifrado.

Tabla 14. Caso de uso de creación de una credencial

ID4.2 – ACCEDER AL LISTADO DE CREDENCIALES

Precondiciones	El usuario administrador debe haber iniciado sesión satisfactoriamente en el sistema.
Poscondiciones	El usuario puede acceder al listado completo de credenciales registradas en el sistema.
Actores	Administrador
Descripción	En caso de que el usuario seleccione esta opción en el menú, se mostrará un listado completo de las credenciales registradas en el sistema. Dicho listado mostrará la aplicación, la compañía y la interfaz de acceso a la que está asociada a la credencial. Además, se indicará el estado de la credencial, el identificador y el secreto, de forma que el administrador pueda consultar estos datos siempre que necesite comunicárselos al cliente en cuestión.
Excepciones	En caso de que el sistema no cuente con ninguna credencial de acceso, se mostrará un mensaje informativo que así lo indique.

Tabla 15. Caso de uso de acceso al listado de credenciales

ID4.3 – EDITAR UNA CREDENCIAL

Precondiciones	El usuario administrador debe haber iniciado sesión satisfactoriamente en el sistema.
Poscondiciones	El estado de una credencial ha sido modificado.
Actores	Administrador
Descripción	El usuario administrador seleccionará, entre la lista completa, aquella credencial que desea modificar. Una vez identificada, el usuario seleccionará el nuevo estado de la credencial y este será modificado mostrando un mensaje que así lo confirme.

Tabla 16. Caso de uso de edición de una credencial

ID4.4 – ELIMINAR UNA CREDENCIAL

Precondiciones	El usuario administrador debe haber iniciado sesión satisfactoriamente en el sistema.
Poscondiciones	Una compañía habrá sido eliminada del sistema y, por lo tanto, también habrán sido eliminados todos sus accesos al servidor de tokens.
Actores	Administrador
Descripción	El usuario seleccionará, entre la lista completa, aquella credencial que desea eliminar. Una vez identificada, el sistema solicitará que el usuario confirme la acción y, en caso afirmativo, esta será eliminada del sistema.

Tabla 17. Caso de uso de eliminación de una credencial

GESTIÓN DE CLAVES

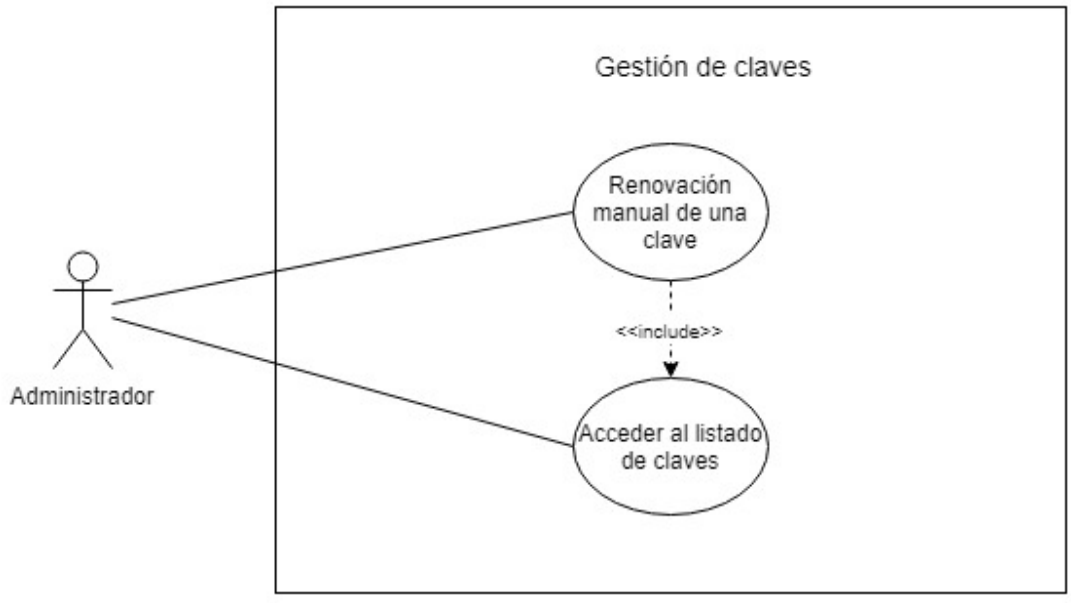

Figura 24. Caso de uso gestión de claves

ID5.1 – Acceder al listado de claves

Precondiciones	El usuario administrador debe haber iniciado sesión satisfactoriamente en el sistema.
Poscondiciones	El usuario puede acceder al listado completo de claves de encriptación registradas en el sistema.
Actores	Administrador
Descripción	En caso de que el usuario seleccione esta opción en el menú, se mostrará un listado completo de las claves de encriptación registradas en el sistema. Dicho listado mostrará el alias asociado a la clave y el estado de esta.

Tabla 18. Caso de uso de acceso al listado de claves

ID5.2 – Renovación manual de una clave

Precondiciones	El usuario administrador debe haber iniciado sesión satisfactoriamente en el sistema y accedido a la opción de gestión de claves en el menú principal.
Poscondiciones	El estado de una credencial ha sido modificado.
Actores	Administrador
Descripción	El usuario seleccionará, de entre el listado de claves, aquella para la cual desea realizar el proceso manual de renovación, esta opción deberá ser confirmada por el usuario. Tras la confirmación, el sistema generará una nueva clave de encriptación a la que se asignarán los mismos periodos de encriptación y el mismo estado que la clave seleccionada por el usuario. Esta nueva clave será utilizada para reencriptar todos los tokens asociados a la clave seleccionada para su renovación, una vez terminado el proceso de reencriptación, la clave será eliminada del sistema y no volverá a ser utilizada.

Tabla 19. Caso de uso de renovación manual de una clave

GESTIÓN DE TOKENS

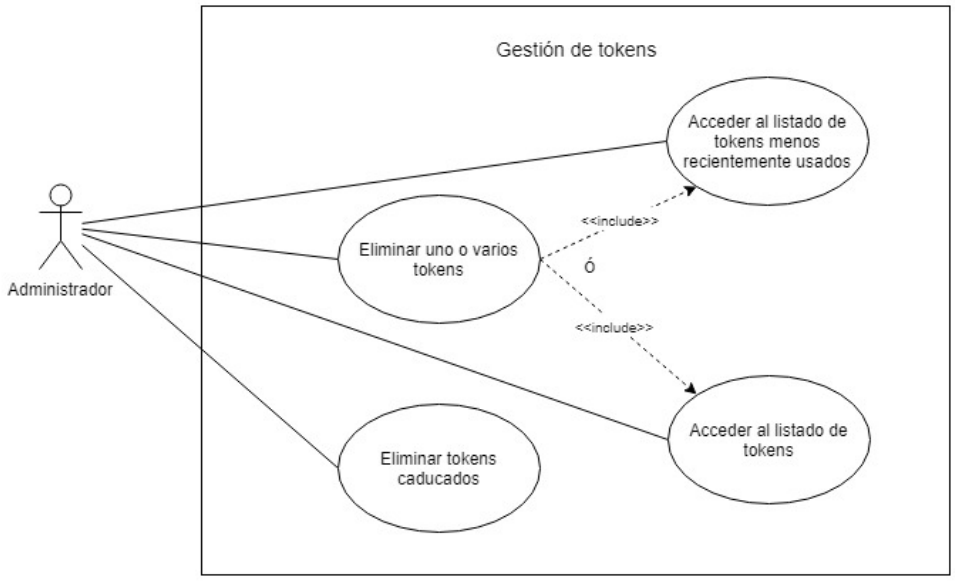

Figura 25. Caso de uso gestión de tokens

ID6.1 – Acceder al listado de tokens

Precondiciones	El usuario administrador debe haber iniciado sesión satisfactoriamente en el sistema.
Poscondiciones	El usuario tiene acceso a un listado completo de los tokens registrados en el sistema.
Actores	Administrador
Descripción	El usuario selecciona en el menú la opción de gestión de tokens, tras esto el sistema muestra un formulario con los siguientes filtros: - Aplicación. - Cliente. - Rango de fecha de caducidad. - Rango de fecha de creación. - Rango de fecha de último uso. El administrador indicará uno o varios valores y tras seleccionar la opción buscar, el sistema mostrará un listado con aquellos tokens que cumplen con los criterios indicados.
Variaciones (escenarios secundarios)	El usuario administrador podría no seleccionar ningún criterio de búsqueda y seleccionar directamente la opción buscar, en este caso se mostrarán todos los tokens registrados en el sistema.
Excepciones	En caso de que el sistema no tengo ningún token, se mostrará un mensaje informativo que así lo indique.

Tabla 20. Caso de uso de acceso al listado de tokens

ID6.2 – Acceder al listado de tokens menos recientemente usados

Precondiciones	El usuario administrador debe haber iniciado sesión satisfactoriamente en el sistema.
Poscondiciones	El usuario tiene acceso a los tokens que no han sido utilizados desde la fecha indicada por el usuario.
Actores	Administrador
Descripción	El usuario indicará la fecha límite para la cual desea realizar la búsqueda, y a continuación, el sistema mostrará todos aquellos tokens que no hayan sido utilizados después de la fecha indicada por el usuario.

Tabla 21. Caso de uso de acceso al listado de tokens menos recientemente usados

ID6.3 – Eliminar uno o varios tokens

Precondiciones	El usuario administrador debe haber iniciado sesión satisfactoriamente en el sistema.
Poscondiciones	Uno o varios tokens han sido eliminados del sistema.
Actores	Administrador
Descripción	El usuario selecciona, en la lista que se proporciona, los tokens que desea eliminar. Podrá seleccionar uno o varios y, a continuación, hará uso de la opción eliminar, tras lo cual estos serán eliminados del sistema y se mostrará un mensaje informativo que así lo confirme.

Tabla 22. Caso de uso de eliminación de tokens

ID6.4 – ELIMINAR TOKENS CADUCADOS

Precondiciones	El usuario administrador debe haber iniciado sesión satisfactoriamente en el sistema.
Poscondiciones	Todos los tokens cuya fecha de caducidad haya sido alcanzada habrán sido eliminados del sistema.
Actores	Administrador
Descripción	El usuario administrador seleccionará la opción de eliminación de tokens temporales, tras lo cual el sistema eliminará todos aquellos tokens cuya fecha de caducidad sea anterior a la fecha de ejecución del proceso. Tras esto el sistema mostrará un mensaje de confirmación.

Tabla 23. Caso de uso de eliminación de tokens caducados

CREACIÓN DE TOKENS

ID7 – CREAR UN TOKEN

Precondiciones	La aplicación y la compañía deben haber sido registradas por el administrador en el sistema, además se debe haber permitido el acceso a través de la interfaz UIFormApp.
Poscondiciones	Un nuevo token está registrado en el sistema.
Actores	UIFormApp
Descripción	Cuando una aplicación utiliza un formulario en el que solicita el PAN y demás datos pertinentes de la tarjeta (fecha de caducidad, titular, CVV, etc.) realiza una petición al servidor de tokens para que le proporcione los datos de acceso necesarios para solicitar, al envío del formulario, la solicitud de conversión de PAN a token. Esta primera petición contendrá la siguiente información: - La credencial que habilita para el uso del servidor de tokens. - Una cadena encriptada con el secreto proporcionado por el servidor de tokens. Esta cadenada estará formada por el timestamp, la credencial y el identificador del ticket. Cuando el servidor de tokens recibe la petición, realiza las validaciones necesarias para asegurar que se trata de una solicitud correcta y generará una contraseña para que la aplicación pueda solicitar la conversión del PAN. Antes del envío de los datos de la tarjeta a la aplicación, se ejecutara una petición de creación de token al servidor La petición deberá contener la siguiente información: - El PAN que se desea convertir. - La credencial que habilita para el uso de esta operación. - La contraseña proporcionada por el servidor de tokens. En caso de que la solicitud sea correcta, el servidor responderá con el token que debe emplear la aplicación.
Variaciones (escenarios secundarios)	El servidor de token también permite la creación de tokens temporales, el escenario transcurrirá exactamente igual que en el caso principal, la única diferencia reside en que el sistema creará el token con una fecha de caducidad establecida.
Excepciones	La creación del token no será llevada a cabo en los siguientes casos: - Credencial inválida - Ticket inválido o caducado. - Contraseña inválida o caducada.

Tabla 24. Caso de uso de creación de token

ELIMINACIÓN DE TOKEN

ID8 – ELIMINAR UN TOKEN

Precondiciones	La aplicación y la compañía deben haber sido registradas por el administrador en el sistema, además se debe haber permitido el acceso a la operación de eliminación de token.
Poscondiciones	Un token ha sido eliminado del sistema
Actores	BackendApp
Descripción	Cuando una aplicación backend borra los datos de pago que tiene almacenados de una tarjeta debería notificar el borrado al servidor de tokens para mantener saneada la información. La petición deberá contener: - La credencial que habilita para el uso de esta operación. - El token que se desea eliminar. - La identidad del usuario que ha iniciado sesión en la aplicación backend.
Excepciones	La eliminación del token no será llevada a cabo en los siguientes casos: - Credencial inválida. - Token inexistente.

Tabla 25. Caso de uso de eliminación de token

OBTENER PAN ENMASCARADO

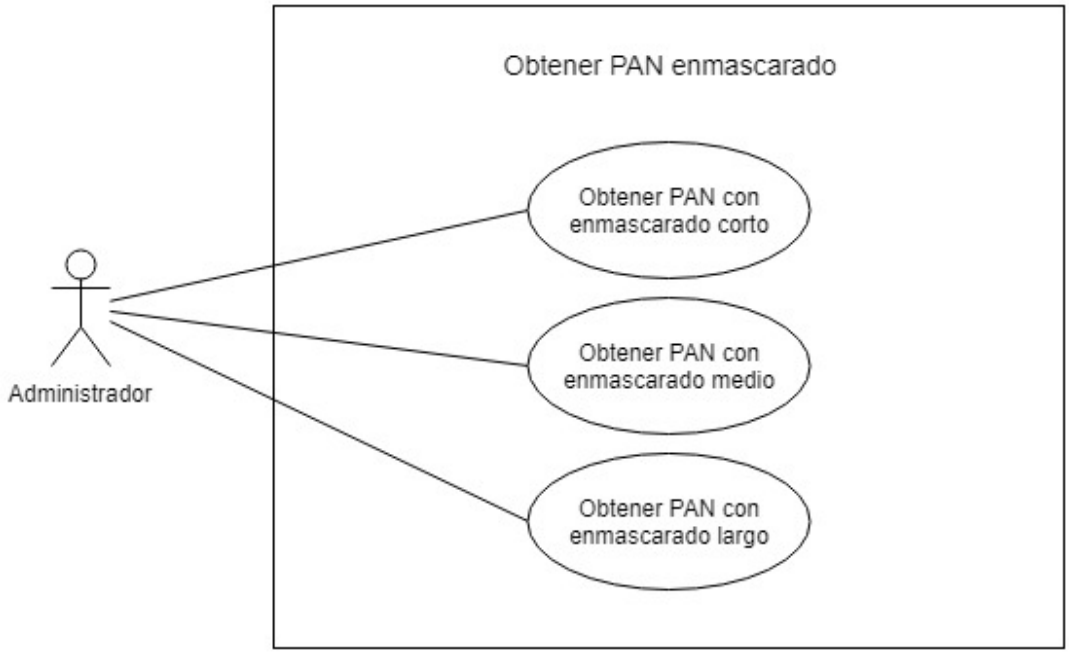

Figura 26. Caso de uso obtener PAN enmascarado

ID9.1 – OBTENER PAN CON ENMASCARADO CORTO

Precondiciones	La aplicación y la compañía deben haber sido registradas por el administrador en el sistema, además se debe haber permitido el acceso a la operación de obtención de PAN enmascarado.
Poscondiciones	La aplicación solicitante obtiene el valor del PAN con enmascarado corto.
Actores	BackendApp
Descripción	Una aplicación backend solicita el PAN enmascarado correspondiente a un token. La petición deberá contener la siguiente información: - Credencial que habilita para el uso de esta operación. - El valor del token. - La identidad del usuario de la aplicación backend que ha abierto la sesión bajo la cual se hace la llamada. Si todo es correcto, el servidor de tokens le proporcionará a la aplicación el valor del PAN utilizando la máscara corta, es decir, únicamente se mostrarán los últimos dígitos permitidos.
Excepciones	La obtención de PAN enmascarado no será llevada a cabo en los siguientes casos: - Credencial inválida. - Token inexistente.

Tabla 26. Caso de uso de obtención de PAN con enmascarado corto

ID9.2 – OBTENER PAN CON ENMASCARADO MEDIO

Precondiciones	La aplicación y la compañía deben haber sido registradas por el administrador en el sistema, además se debe haber permitido el acceso a la operación de obtención de PAN enmascarado.
Poscondiciones	La aplicación solicitante obtiene el valor del PAN con enmascarado medio.
Actores	BackendApp
Descripción	Una aplicación backend solicita el PAN enmascarado correspondiente a un token. La petición deberá contener la siguiente información: - Credencial que habilita para el uso de esta operación. - El valor del token. - La identidad del usuario de la aplicación backend que ha abierto la sesión bajo la cual se hace la llamada. Si todo es correcto, el servidor de tokens le proporcionará a la aplicación el valor del PAN utilizando la máscara media, es decir, se mostrarán únicamente los seis primeros dígitos del PAN.
Excepciones	La obtención de PAN enmascarado no será llevada a cabo en los siguientes casos: - Credencial inválida. - Token inexistente.

Tabla 27. Caso de uso de obtención de PAN con enmascarado medio

ID9.3 – Obtener PAN con enmascarado largo

Precondiciones	La aplicación y la compañía deben haber sido registradas por el administrador en el sistema, además se debe haber permitido el acceso a la operación de obtención de PAN enmascarado.
Poscondiciones	La aplicación solicitante obtiene el valor del PAN con enmascarado largo.
Actores	BackendApp
Descripción	Una aplicación backend solicita el PAN enmascarado correspondiente a un token. La petición deberá contener la siguiente información: - Credencial que habilita para el uso de esta operación. - El valor del token. - La identidad del usuario de la aplicación backend que ha abierto la sesión bajo la cual se hace la llamada. Si todo es correcto, el servidor de tokens le proporcionará a la aplicación el valor del PAN utilizando la máscara larga, es decir, se mostrarán los seis primeros dígitos del PAN y los n últimos permitidos en función de la longitud del PAN.
Excepciones	La obtención de PAN enmascarado no será llevada a cabo en los siguientes casos: - Credencial inválida. - Token inexistente.

Tabla 28. Caso de uso de obtención de PAN con enmascarado largo

OBTENCIÓN DE PAN

ID10 – Obtención de PAN a partir de token

Precondiciones	La aplicación y la compañía deben haber sido registradas por el administrador en el sistema, además se debe haber permitido el acceso a la operación de obtención del PAN
Poscondiciones	La aplicación solicitante obtiene el valor del PAN.
Actores	PciApp
Descripción	Se trata de la única operación que se realiza desde las aplicaciones situadas en la zona PCI y que, por tanto, pueden solicitar el PAN asociado a un token. La petición deberá contener los siguientes datos: - La credencial que habilita para el uso de esta operación. - El valor del token. - La identidad del usuario que ha iniciado sesión en la aplicación backend.
Excepciones	La obtención de PAN no será llevada a cabo en los siguientes casos: - Credencial inválida. - Token inexistente.

Tabla 29. Caso de uso de obtención de PAN

ROTACIÓN DE CLAVE

ID11 – ROTACIÓN DE CLAVE

Precondiciones	La aplicación y la compañía deben haber sido registradas por el administrador en el sistema, además se debe haber permitido el acceso a la operación de obtención del PAN
Poscondiciones	La aplicación solicitante obtiene el valor del PAN.
Actores	KeyRotation Timer
Descripción	Proceso periódico y automático que se encarga de sustituir la clave de encriptación periódicamente. Durante su ejecución se efectuarán los siguientes pasos: - Se generará una nueva clave y se establecerá como la activa en el sistema. A partir de este momento, todas las nuevas peticiones de creación de tokens se realizarán con esta. - Se buscarán todos los tokens que permanecen en el sistema con la clave que ha llegado al final de su criptoperiodo. Todos y cada uno de ellos se reencriptarán con la nueva clave. - La clave que ha llegado al final de su criptoperiodo se elimina del sistema.

Tabla 30. Caso de uso de rotación de clave

LIMPIEZAS

ID12.1 – LIMPIEZA DE TOKENS TEMPORALES

Precondiciones	-
Poscondiciones	Los tokens caducados habrán sido eliminados del sistema.
Actores	Cleaning Timer
Descripción	El disparo de este timer eliminará todos los tokens temporales cuya fecha de expiración sea anterior al momento en el que se ejecuta el barrido. El periodo de disparo será configurable en el sistema.

Tabla 31. Caso de uso de limpieza de tokens temporales

ID12.2 – LIMPIEZA DE TICKETS

Precondiciones	-
Poscondiciones	Los tickets caducados habrán sido eliminados del sistema.
Actores	Cleaning Timer
Descripción	El disparo de este timer eliminará todos los tickers cuya fecha de expiración sea anterior al momento en el que se ejecuta el barrido. El periodo de disparo será configurable en el sistema.

Tabla 32. Caso de uso de limpieza de tickets

ID12.3 – LIMPIEZA DE CONTRASEÑAS

Precondiciones	-
Poscondiciones	Las contraseñas caducadas habrán sido eliminadas del sistema.
Actores	Cleaning Timer
Descripción	El disparo de este timer eliminará todas las contraseñas cuya fecha de expiración sea anterior al momento en el que se ejecuta el barrido. El periodo de disparo será configurable en el sistema.

Tabla 33. Caso de uso de limpieza de contraseñas

4.10 ANÁLISIS DE INTERFACES DE USUARIO

A continuación se presentan una serie de esquemas o prototipos de pantallas, de forma que se identifican las ubicaciones de los diferentes elementos que componen las interfaces.

4.10.1 Descripción de la interfaz

INICIO DE SESIÓN

Será la pantalla que se muestre cuando un usuario administrador acceda al servidor de tokens, en ella deberá proporcionar el usuario y la contraseña para tener acceso al resto de la aplicación.

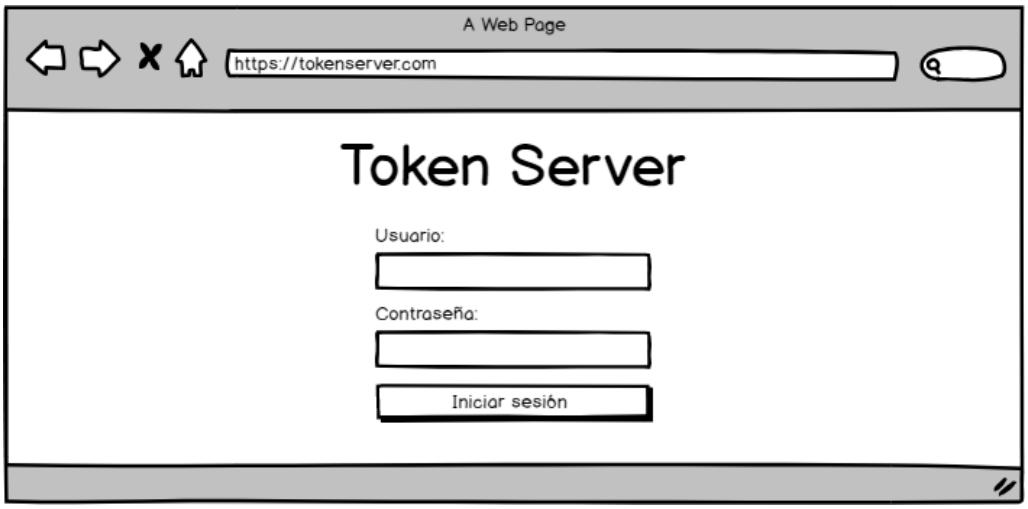

Figura 27. Prototipo interfaz inicio de sesión

CONTROL DE ACCESO

Esta pantalla será la que el administrador utilizará para gestionar el control de acceso al servidor de tokens. Se trata de una vista tabular que cuenta con las siguientes subvistas:

GESTIÓN DE APLICACIONES

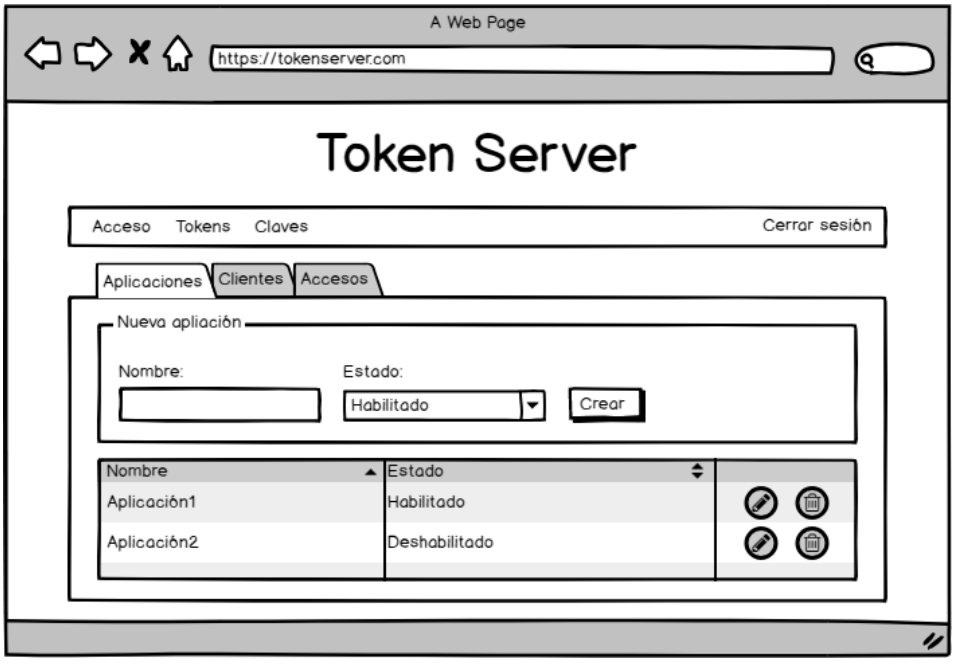

Figura 28. Prototipo interfaz de gestión de aplicaciones

GESTIÓN DE COMPAÑÍAS

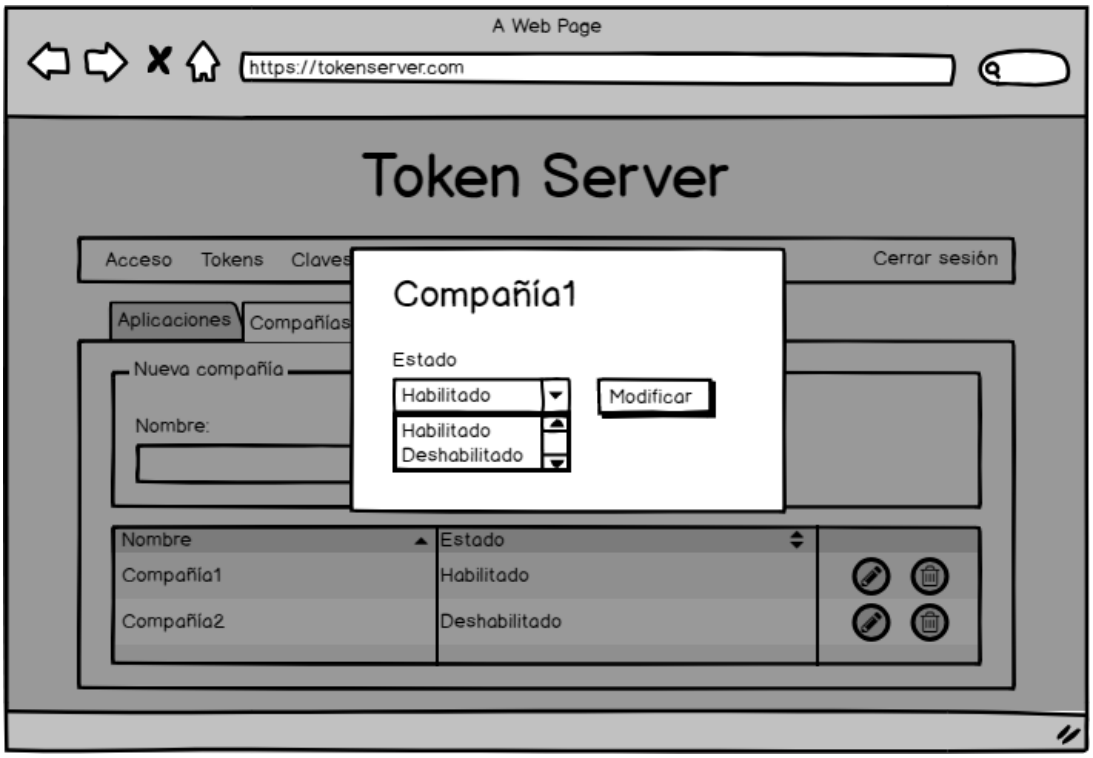

Figura 29. Prototipo interfaz de gestión de compañías

GESTIÓN DE CREDENCIALES

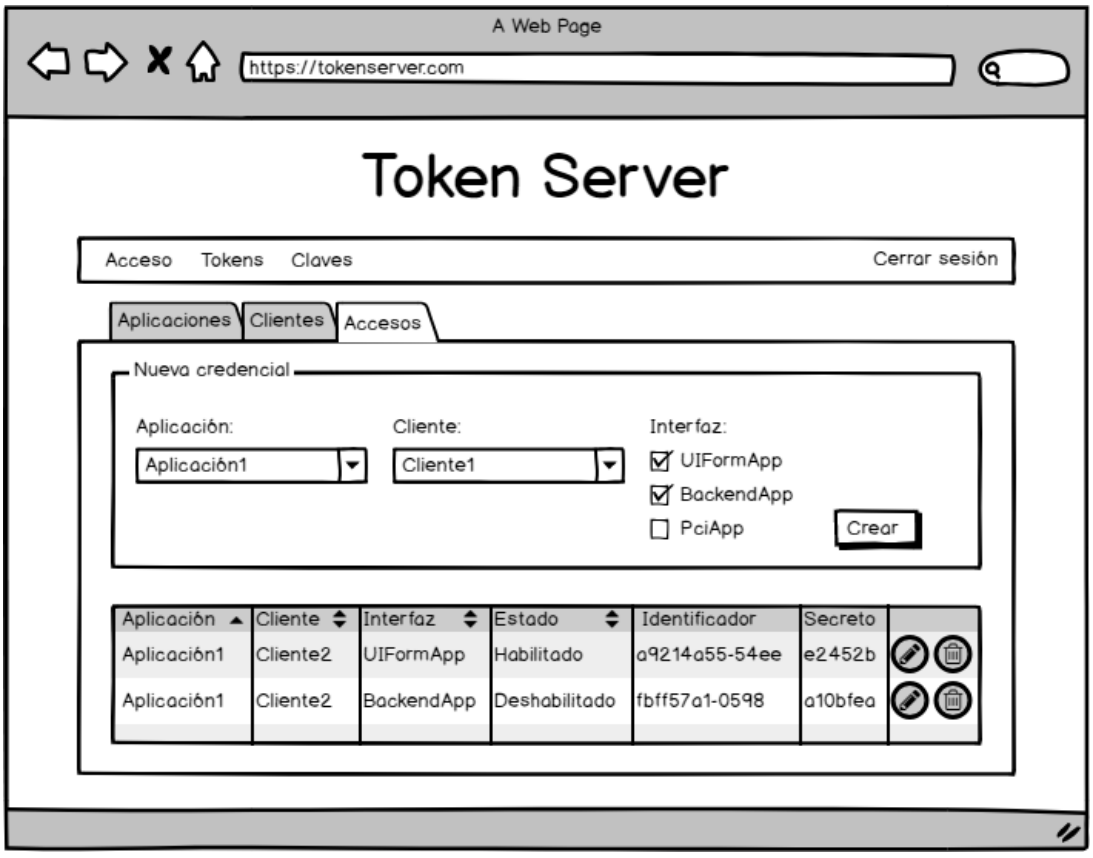

Figura 30. Prototipo interfaz de gestión de accesos

GESTIÓN DE TOKENS

La segunda opción del menú de la aplicación permite acceder a la pantalla de gestión de tokens. En la parte superior el administrador podrá llevar a cabo un filtrado para encontrar, con mayor facilidad, los tokens que desea identificar.

Además, también en la parte superior, es incluirá una zona para que el usuario pueda buscar los tokens menos recientemente usados a partir de una fecha dada.

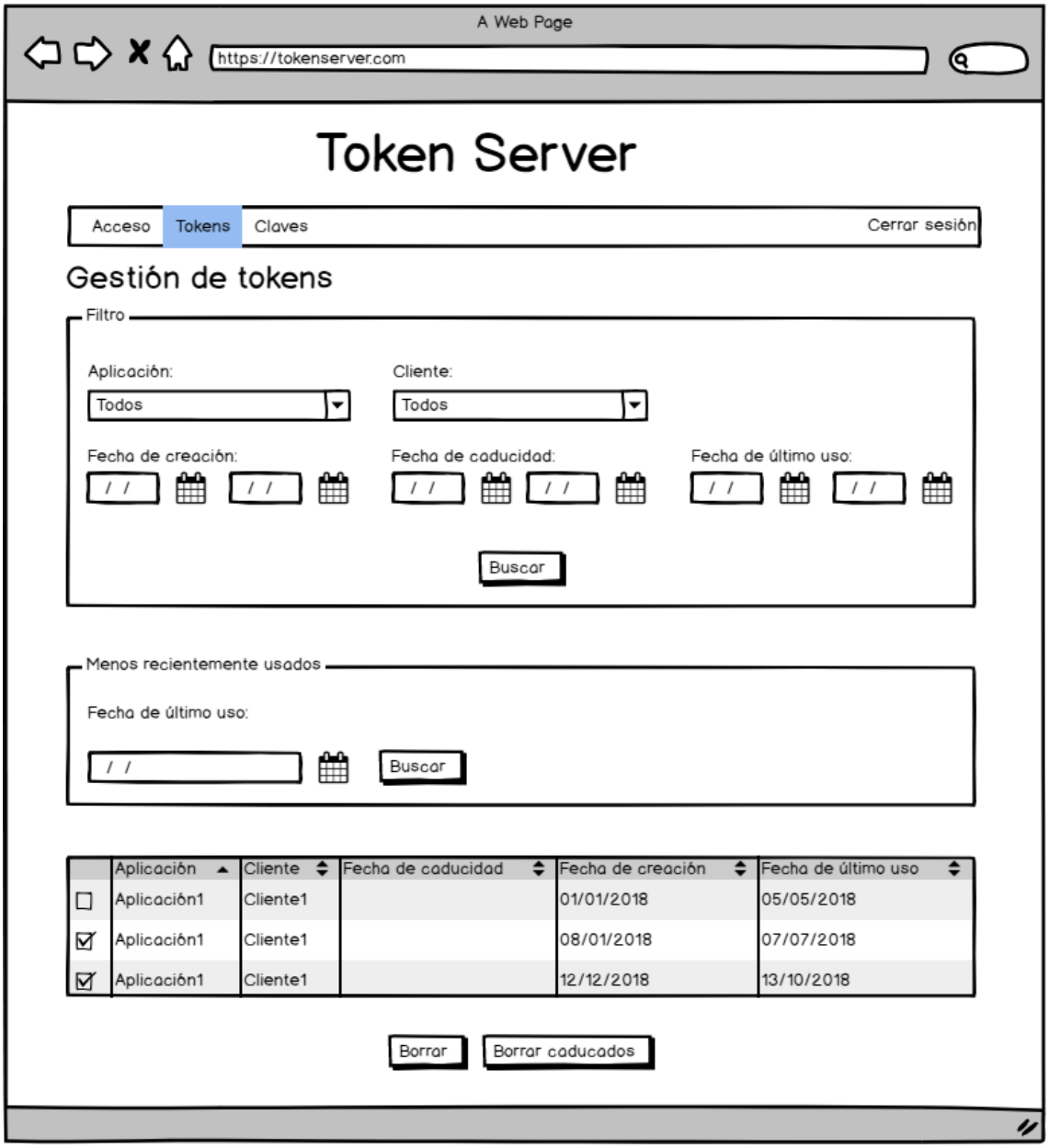

Figura 31. Prototipo de interfaz de gestión de tokens.

Por último, la aplicación mostrará un listado con la información asociada a cada uno de los tokens que han coincidido con la búsqueda realizada por el administrador.

GESTIÓN DE CLAVES

La interfaz de gestión de claves de encriptación estará formada únicamente por el listado de estas. Sobre cada una se muestra el alias que la identifica así como el estado de esta. Además, desde aquí será posible ejecutar el proceso de renovación de clave manual. Ver 4.4Rotación de claves.

Figura 32. Prototipo de interfaz de gestión de claves

4.10.2Relación interfaz / casos de uso

En la siguiente tabla queda reflejada la relación entre los casos de uso descritos en Especificación de casos de uso y los prototipos de interfaz de usuario que se han presentado anteriormente.

	INICIO DE SESIÓN	GESTIÓN DE APLICACIONES	GESTIÓN DE COMPAÑÍAS	GESTIÓN DE CREDENCIALES	GESTIÓN DE TOKENS	GESTIÓN DE CLAVES
ID1	•					
ID2.1		•				
ID2.2		•				
ID2.3		•				
ID2.4		•				
ID3.1			•			
ID3.2			•			
ID3.3			•			
ID3.4			•			
ID4.1				•		
ID4.2				•		
ID4.3				•		
ID4.4				•		
ID5.1						•
ID5.2						•
ID6.1					•	
ID6.2					•	
ID6.3					•	
ID6.4					•	

Tabla 34. Relación interfaz de usuario / caso de uso

El resto de casos de uso descriptos en Especificación de casos de uso no cuentan con una interfaz de usuario.

4.11 ESPECIFICACIÓN DEL PLAN DE PRUEBAS

A continuación se detallan los diferentes casos de prueba planteados en función de los casos de uso descritos en apartados anteriores.

ACCESO A LA INTERFAZ DE ADMINISTRACIÓN

CASO DE PRUEBA: CP1.1.1	
Entrada	**Resultado Esperado**
Acceso a la aplicación con un usuario y contraseña asociados a un usuario administrador.	El usuario logra acceder a la interfaz de administración y se le permite acceder a catálogo de operaciones disponibles para este actor.
CASO DE PRUEBA: CP1.1.2	
Entrada	**Resultado Esperado**
Acceso a la aplicación con un identificador de usuario que no se encuentra registrado en el sistema	El usuario no logra acceder a la aplicación y se muestra un mensaje de error.
CASO DE PRUEBA: CP1.1.2	
Entrada	**Resultado Esperado**
Acceso a la aplicación con una contraseña inválida.	El usuario no logra acceder a la aplicación y se muestra un mensaje de error.

Tabla 35. Casos de prueba de acceso a la interfaz de administración

GESTIÓN DE APLICACIONES

CASO DE PRUEBA: CP2.1.1	
Entrada	**Resultado Esperado**
En la creación de una aplicación, se proporciona un nombre que no se encuentra registrado en el sistema.	La aplicación se crea correctamente.
CASO DE PRUEBA: CP2.1.2	
Entrada	**Resultado Esperado**
En la creación de una aplicación, no se proporciona un nombre.	La aplicación no se crea y se muestra un mensaje de error.
CASO DE PRUEBA: CP2.1.3	
Entrada	**Resultado Esperado**
En la creación de una aplicación, se proporciona un nombre que ya se encuentra asociado a otra aplicación registrada previamente.	La aplicación no se crea y se muestra un mensaje de error.
CASO DE PRUEBA: CP2.2.1	
Entrada	**Resultado Esperado**
Se selecciona la opción de gestión de aplicaciones en el menú.	El sistema muestra un listado de las aplicaciones registradas, en el que se puede ver su nombre y estado.

CASO DE PRUEBA: CP2.2.2	
Entrada	**Resultado Esperado**
En el listado de aplicaciones, se introduce un valor en el filtro que no se corresponde con ninguna aplicación	El sistema muestra el listado vacío y muestra un mensaje de información.
CASO DE PRUEBA: CP2.2.3	
Entrada	**Resultado Esperado**
En el listado de aplicaciones, se introduce un valor en el filtro que se corresponde con varias aplicaciones.	El sistema muestra el listado con el subconjunto de aplicaciones que cumplen con el criterio proporcionado por el usuario.
CASO DE PRUEBA: CP2.3.1	
Entrada	**Resultado Esperado**
Se modifica el estado de una aplicación.	El estado es modificado correctamente y así se puede observar en el listado de aplicaciones
CASO DE PRUEBA: CP2.4.1	
Entrada	**Resultado Esperado**
Se selecciona la opción de borrado sobre una de las aplicaciones a las que se tiene acceso a través del listado y se confirma la acción. La aplicación seleccionada no cuenta con ningún token activo.	La aplicación es eliminada del sistema y se muestra un mensaje de confirmación.
CASO DE PRUEBA: CP2.4.2	
Entrada	**Resultado Esperado**
Se selecciona la opción de borrado sobre una de las aplicaciones a las que se tiene acceso a través del listado y se confirma la acción. La aplicación seleccionada cuenta con varios tokens activos.	La aplicación no es eliminada del sistema y se muestra un mensaje de error.

Tabla 36. Casos de prueba de gestión de aplicaciones

GESTIÓN DE COMPAÑÍAS

CASO DE PRUEBA: CP3.1.1	
Entrada	**Resultado Esperado**
En la creación de una compañía, se proporciona un nombre que no se encuentra registrado en el sistema.	La compañía se crea correctamente.
CASO DE PRUEBA: CP3.1.2	
Entrada	**Resultado Esperado**
En la creación de una compañía, no se proporciona un nombre.	La compañía no se crea y se muestra un mensaje de error.
CASO DE PRUEBA: CP3.1.3	
Entrada	**Resultado Esperado**
En la creación de una compañía, se proporciona un nombre que ya se encuentra asociado a otra compañía registrada previamente.	La compañía no se crea y se muestra un mensaje de error.

CASO DE PRUEBA: CP3.2.1	
Entrada	**Resultado Esperado**
Se selecciona la opción de gestión de compañías en el menú.	El sistema muestra un listado de las compañías registradas, en el que se puede ver su nombre y estado.
CASO DE PRUEBA: CP3.2.2	
Entrada	**Resultado Esperado**
En el listado de compañías, se introduce un valor en el filtro que no se corresponde con ninguna compañía.	El sistema muestra el listado vacío y muestra un mensaje de información.
CASO DE PRUEBA: CP3.2.3	
Entrada	**Resultado Esperado**
En el listado de compañías, se introduce un valor en el filtro que se corresponde con varias compañías.	El sistema muestra el listado con el subconjunto de compañías que cumplen con el criterio proporcionado por el usuario.
CASO DE PRUEBA: CP3.3.1	
Entrada	**Resultado Esperado**
Se modifica el estado de una compañía.	El estado es modificado correctamente y así se puede observar en el listado de compañías,
CASO DE PRUEBA: CP3.4.1	
Entrada	**Resultado Esperado**
Se selecciona la opción de borrado sobre una de las compañías a las que se tiene acceso a través del listado y se confirma la acción. La compañía seleccionada no cuenta con ningún token activo.	La compañía es eliminada del sistema y se muestra un mensaje de confirmación.
CASO DE PRUEBA: CP3.4.2	
Entrada	**Resultado Esperado**
Se selecciona la opción de borrado sobre una de las compañías a las que se tiene acceso a través del listado y se confirma la acción. La compañía seleccionada cuenta con varios tokens activos.	La compañía no es eliminada del sistema y se muestra un mensaje de error.

Tabla 37. Casos de prueba de gestión de compañías

GESTIÓN DE ACCESOS

CASO DE PRUEBA: CP4.1.1	
Entrada	**Resultado Esperado**
En la creación de un nuevo permiso de acceso, se selecciona una aplicación, un cliente y una única interfaz.	La credencial de acceso se ha creado correctamente y se ha generado un identificador de credencial y un secreto compartido.

CASO DE PRUEBA: CP4.1.2	
Entrada	**Resultado Esperado**
En la creación de un nuevo permiso de acceso, se selecciona una aplicación, un cliente y tres interfaces.	Se crear tres credenciales de acceso.
CASO DE PRUEBA: CP4.2.1	
Entrada	**Resultado Esperado**
Se selecciona la opción de gestión de credenciales en el menú.	El sistema muestra un listado de las credenciales registradas, en el que se puede ver los siguientes datos sobre cada credencial: aplicación, compañía, interfaz, estado, identificador y secreto.
CASO DE PRUEBA: CP4.2.2	
Entrada	**Resultado Esperado**
En el listado de credenciales, se introduce un valor en el filtro que no se corresponde con ninguna credencial.	El sistema muestra el listado vacío y muestra un mensaje de información.
CASO DE PRUEBA: CP4.2.3	
Entrada	**Resultado Esperado**
En el listado de credenciales, se introduce un valor en el filtro que se corresponde con varias credenciales de acceso.	El sistema muestra el listado con el subconjunto de credenciales que cumplen con el criterio proporcionado por el usuario.
CASO DE PRUEBA: CP4.3.1	
Entrada	**Resultado Esperado**
Se deshabilita una credencial de acceso, modificando el estado de esta.	El estado es modificado correctamente y así se puede observar en el listado de credenciales.
CASO DE PRUEBA: CP4.4.1	
Entrada	**Resultado Esperado**
Se selecciona la opción de borrado sobre una de las credenciales a las que se tiene acceso a través del listado y se confirma la acción.	La credencial es eliminada del sistema y se muestra un mensaje de confirmación.

Tabla 38. Casos de prueba de gestión de accesos

GESTIÓN DE CLAVES

CASO DE PRUEBA: CP5.1.1	
Entrada	**Resultado Esperado**
Se selecciona la opción de gestión de claves en el menú.	El sistema muestra un listado de las claves de encriptación registradas, en el que se puede ver los siguientes datos sobre cada clave: alias y estado.
CASO DE PRUEBA: CP5.1.2	
Entrada	**Resultado Esperado**
En el listado de claves, se introduce un valor en el filtro que no se corresponde con ninguna clave.	El sistema muestra el listado vacío y muestra un mensaje de información.

CASO DE PRUEBA: CP5.1.3	
Entrada	**Resultado Esperado**
En el listado de claves, se introduce un valor en el filtro que se corresponde con varias claves de encriptación.	El sistema muestra el listado con el subconjunto de claves que cumplen con el criterio proporcionado por el usuario.
CASO DE PRUEBA: CP5.2.1	
Entrada	**Resultado Esperado**
Se selecciona una clave de encriptación para su renovación manual.	La clave seleccionada es sustituida por una nueva, con el mismo estado que el anterior, lo cual queda reflejado en el listado.

Tabla 39. Casos de prueba de gestión de claves

GESTIÓN DE TOKENS

CASO DE PRUEBA: CP6.1.1	
Entrada	**Resultado Esperado**
Se accede a la opción de gestión de tokens desde el menú principal y, sin seleccionar ningún valor en los filtros propuestos, se realiza la búsqueda.	El sistema muestra todos los tokens disponibles.
CASO DE PRUEBA: CP6.1.2	
Entrada	**Resultado Esperado**
Se accede a la opción de gestión de tokens desde el menú principal, se seleccionan varios filtros de entre los propuestos y se realiza la búsqueda.	El sistema muestra un subconjunto de los tokens registrados en el sistema. Los tokens mostrados cumplen los requisitos de búsqueda introducidos por el usuario.
CASO DE PRUEBA: CP6.2.1	
Entrada	**Resultado Esperado**
Se selecciona la opción de búsqueda de tokens menos recientemente usados, sin indicar la fecha límite.	El sistema muestra un mensaje de error, puesto que la indicación de la fecha límite es obligatoria.
CASO DE PRUEBA: CP6.2.2	
Entrada	**Resultado Esperado**
Se selecciona la opción de búsqueda de tokens menos recientemente usados, indicando una fecha límite.	El sistema muestra un listado con los tokens cuya fecha de último uso es posterior a la indicada por el usuario.
CASO DE PRUEBA: CP6.3.1	
Entrada	**Resultado Esperado**
Se selecciona un token en el listado y se elimina.	El token desaparece de la lista y se muestra un mensaje de confirmación.
CASO DE PRUEBA: CP6.3.1	
Entrada	**Resultado Esperado**
Se seleccionan varios tokens en el listado y se eliminan.	Los tokens desaparecen de la lista y se muestra un mensaje de confirmación.

CASO DE PRUEBA: CP6.4.1	
Entrada	**Resultado Esperado**
Se selecciona la opción de eliminar los tokens caducados.	Los tokens cuya fecha de caducidad haya sido alcanza son eliminados del sistema y se muestra un mensaje de confirmación.

Tabla 40. Casos de prueba de gestión de tokens

CREACIÓN DE TOKENS

CASO DE PRUEBA: CP7.1.1	
Entrada	**Resultado Esperado**
Solicitar una creación de token con una credencial que no se encuentra registrada en el sistema.	El token no se crea.
CASO DE PRUEBA: CP7.1.2	
Entrada	**Resultado Esperado**
Solicitar una creación de token con una credencial que se encuentra deshabilidatada en el sistema.	El token no se crea.
CASO DE PRUEBA: CP7.1.3	
Entrada	**Resultado Esperado**
Solicitar una creación de token utilizando un secreto que no se encuentra asociado a la credencial.	El token no se crea.
CASO DE PRUEBA: CP7.1.4	
Entrada	**Resultado Esperado**
Solicitar una creación de token utilizando un ticket que ya ha sido utilizado previamente.	El token no se crea.
CASO DE PRUEBA: CP7.1.5	
Entrada	**Resultado Esperado**
Solicitar una creación de token utilizando una credencial que no cuenta con permiso para realizar la operación.	El token no se crea.
CASO DE PRUEBA: CP7.1.6	
Entrada	**Resultado Esperado**
Solicitar una creación de token utilizando una contraseña inválida.	El token no se crea.
CASO DE PRUEBA: CP7.1.7	
Entrada	**Resultado Esperado**
Solicitar una creación de token utilizando una contraseña caducada.	El token no se crea.

CASO DE PRUEBA: CP7.1.8	
Entrada	**Resultado Esperado**
Solicitar una creación de token utilizando una contraseña válida, en vigencia, pero que no está asociada a la credencial con la que se autentifica la aplicación.	El token no se crea.
CASO DE PRUEBA: CP7.1.9	
Entrada	**Resultado Esperado**
Solicitar una creación de token utilizando una credencial válida, encriptada con el secreto asociado a dicha credencial. El ticket que se envía no ha sido utilizado previamente y se utiliza la contraseña proporcionada por el servidor de token durante su periodo de vigencia.	El token se crea y es devuelto al navegador.

Tabla 41. Casos de prueba de creación de tokens

ELIMINACIÓN DE TOKEN

CASO DE PRUEBA: CP8.1.1	
Entrada	**Resultado Esperado**
Solicitar la eliminación de un token con una credencial que no se encuentra registrada en el sistema.	El token no se elimina.
CASO DE PRUEBA: CP8.1.2	
Entrada	**Resultado Esperado**
Solicitar la eliminación de un token con una credencial que se encuentra deshabilitada en el sistema.	El token no se elimina.
CASO DE PRUEBA: CP8.1.3	
Entrada	**Resultado Esperado**
Solicitar la eliminación de un token utilizando una credencial que no cuenta con permiso para realizar la operación.	El token no se elimina.
CASO DE PRUEBA: CP8.1.4	
Entrada	**Resultado Esperado**
Solicitar la eliminación de un token que no existe.	El token no se elimina.
CASO DE PRUEBA: CP8.1.5	
Entrada	**Resultado Esperado**
Solicitar la eliminación de un token utilizando una credencial válida, con permiso para realizar esta operación y con un token correcto.	El token se elimina.

Tabla 42. Casos de prueba de eliminación de token

OBTENCIÓN DE PAN ENMASCARADO

CASO DE PRUEBA: CP9.1.1	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN con enmascaramiento corto utilizando una credencial que no existe en el sistema.	El PAN enmascarado no se devuelve.
CASO DE PRUEBA: CP9.1.2	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN con enmascaramiento corto utilizando una credencial que se encuentra deshabilitada.	El PAN enmascarado no se devuelve.
CASO DE PRUEBA: CP9.1.3	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN con enmascaramiento corto utilizando una credencial que no cuenta con permiso para realizar la operación.	El PAN enmascarado no se devuelve.
CASO DE PRUEBA: CP9.1.4	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN con enmascaramiento corto para un token que no existe.	El PAN enmascarado no se devuelve.
CASO DE PRUEBA: CP9.1.5	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN con enmascaramiento corto con una credencial válida para un token existente en el sistema.	El sistema proporciona una cadena de texto en la que únicamente se muestran los n últimos dígitos permitidos en función de la longitud del PAN solicitado.
CASO DE PRUEBA: CP9.2.1	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN con enmascaramiento medio utilizando una credencial que no existe en el sistema.	El PAN enmascarado no se devuelve.
CASO DE PRUEBA: CP9.2.2	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN con enmascaramiento medio utilizando una credencial que se encuentra deshabilitada.	El PAN enmascarado no se devuelve.
CASO DE PRUEBA: CP9.2.3	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN con enmascaramiento medio utilizando una credencial que no cuenta con permiso para realizar la operación.	El PAN enmascarado no se devuelve.

CASO DE PRUEBA: CP9.2.4	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN con enmascaramiento medio para un token que no existe.	El PAN enmascarado no se devuelve.
CASO DE PRUEBA: CP9.2.5	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN con enmascaramiento medio con una credencial válida para un token existente en el sistema.	El sistema proporciona una cadena de texto en la que únicamente se muestran los seis primeros dígitos del PAN solicitado.
CASO DE PRUEBA: CP9.3.1	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN con enmascaramiento largo utilizando una credencial que no existe en el sistema.	El PAN enmascarado no se devuelve.
CASO DE PRUEBA: CP9.3.2	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN con enmascaramiento largo utilizando una credencial que se encuentra deshabilitada.	El PAN enmascarado no se devuelve.
CASO DE PRUEBA: CP9.3.3	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN con enmascaramiento largo utilizando una credencial que no cuenta con permiso para realizar la operación.	El PAN enmascarado no se devuelve.
CASO DE PRUEBA: CP9.3.4	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN con enmascaramiento largo para un token que no existe.	El PAN enmascarado no se devuelve.
CASO DE PRUEBA: CP9.3.5	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN con enmascaramiento largo con una credencial válida para un token existente en el sistema.	El sistema proporciona una cadena de texto en la que únicamente se muestran los seis primeros dígitos y los n últimos en función de la longitud del PAN solicitado.

Tabla 43. Casos de prueba de obtención de PAN enmascarado

OBTENCIÓN DE PAN

CASO DE PRUEBA: CP10.1.1	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN utilizando una credencial que no existe en el sistema.	El PAN no se devuelve.

CASO DE PRUEBA: CP10.1.2	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN utilizando una credencial que se encuentra deshabilitada.	El PAN no se devuelve.
CASO DE PRUEBA: CP10.1.3	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN utilizando una credencial que no cuenta con permiso para realizar la operación.	El PAN no se devuelve.
CASO DE PRUEBA: CP10.1.4	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN para un token que no existe.	El PAN no se devuelve.
CASO DE PRUEBA: CP10.1.5	
Entrada	**Resultado Esperado**
Solicitar la obtención de PAN una credencial válida para un token existente en el sistema.	El sistema devolverá a la aplicación solicitante el PAN asociado al token proporcionado.

Tabla 44. Casos de prueba de obtención de PAN

ROTACIÓN DE CLAVE

CASO DE PRUEBA: CP11.1.1	
Entrada	**Resultado Esperado**
Se programa el timer para que se ejecute a una hora prefijada.	Se comprueba que existe una nueva clave en el almacén, que está marcada como activa. Además, deben existir tokens cuya clave de encriptación es la nueva.

Tabla 45. Casos de uso de rotación de clave

LIMPIEZAS

CASO DE PRUEBA: CP12.1.1	
Entrada	**Resultado Esperado**
Se realiza una consulta para comprobar el número de tokens caducados registrados en el sistema. Se ejecuta el timer de borrado.	Se realiza la nueva consulta y el resultado es 0.
CASO DE PRUEBA: CP12.2.1	
Entrada	**Resultado Esperado**
Se realiza una consulta para comprobar el número de tickets caducados registrados en el sistema. Se ejecuta el timer de borrado.	Se realiza la nueva consulta y el resultado es 0.
CASO DE PRUEBA: CP12.3.1	
Entrada	**Resultado Esperado**
Se realiza una consulta para comprobar el número de contraseñas caducadas registradas en el sistema. Se ejecuta el timer de borrado.	Se realiza la nueva consulta y el resultado es 0.

Tabla 46. Casos de prueba de timers de limpieza

5. DISEÑO

5.1 ARQUITECTURA DEL SISTEMA

El principal objetivo de este apartado es realizar una descripción de los diferentes elementos que componen la arquitectura del sistema, para ello se realizará un análisis de esta, incluyendo la descripción del despliegue de dicha arquitectura.

El modelo de arquitectura seguido a la hora de desarrollar el servidor de tokens ha sido Clean Architecture. Se trata de una arquitectura popularizada por Robert Cecil Martin, más conocido como "Uncle Bob" y se basa en la estructuración del código en capas contiguas. De esta forma una capa únicamente tiene comunicación con aquellas que le son inmediatas.

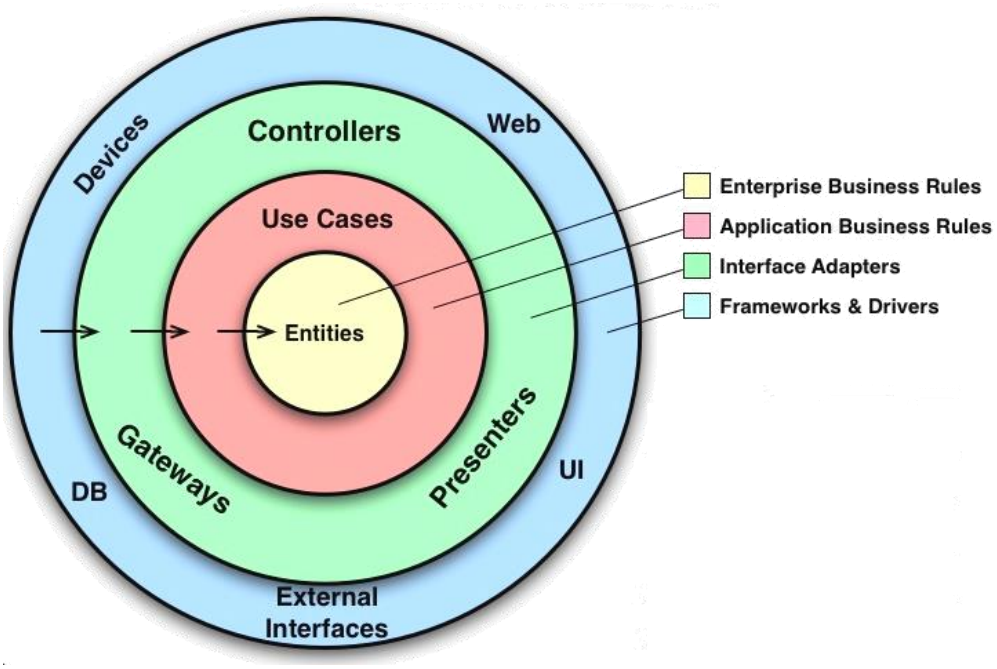

Figura 33. Clean architecture

Los niveles que componen esta arquitectura son:

- UI, que representa la interfaz de usuario.
- Presenters, donde se sitúan las clases que atienden los eventos generados por la interfaz de usuario y que, por tanto, también se encargan de llevar a cabo el formateo de la información que aparece en la interfaz.
- Use cases, aquí se encuentran las clases dedicadas a las reglas de negocio de la aplicación.
- Entities, que incluye las clases que representan el modelo de datos de la aplicación.

La regla principal de esta arquitectura es la regla de la dependencia. Esta regla indica que las dependencias a nivel de código fuente solo pueden apuntar hacia el interior del círculo. De esta forma, un círculo interior no puede saber nada sobre el círculo que le rodea, esto es, el código desarrollado en un círculo exterior no puede ser utilizado por el interno.

5.1.1 Patrones de diseño

A continuación se detallan los patrones de diseño utilizados para estructurar la arquitectura del sistema.

MODELO VISTA PRESENTADOR

En la capa de presentación es donde se gestiona todo o que tiene que ver con la forma en que la información es mostrada al usuario, es decir, sobre cómo funcionan las vistas. Con intención de conseguir un mejor manejo de esta capa sea optado por utilizar un patrón Modelo Vista Presentador, MVP, el cual es un derivado del clásico patrón Modelo Vista Controlador pero, en este caso, únicamente se encarga de implementar la lógica que ocurre en la capa de presentación.

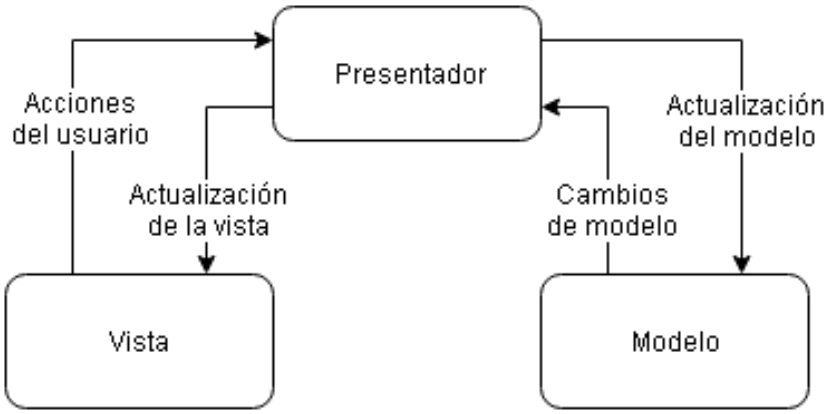

Figura 34. Modelo Vista Presentador

- La vista únicamente tiene métodos para pasarle la información ya formateada a las páginas de la interfaz de usuario. Además, contará con los métodos necesarios para obtener los datos proporcionados por el usuario.
- El presentador tiene acceso tanto al modelo como a la vista, puesto que hace de intermediario entre ellos. Es quien se encarga de realizar la lógica de presentación, tiene acceso a los datos que le proporciona la vista e interactúa con el modelo para llevar a cabo las modificaciones que el usuario ha seleccionado.
- El modelo se encarga de implementar el caso de uso que el usuario ha solicitado llevar a cabo.

REPOSITORIO

La capa de datos se encarga de obtener todos los datos que la aplicación necesita para su correcto funcionamiento, en este caso obtenidos de una base de datos. Sin embargo, este patrón permite realizar una abstracción del origen de los datos, de modo que no importa de donde vengan estos sino que van a estar presentes en la aplicación y se podrán utilizar para llevar a cabo las operaciones, independientemente de su lugar de origen.

5.1.2 Diagrama de despliegue

A continuación se presenta del diagrama de despliegue de la aplicación.

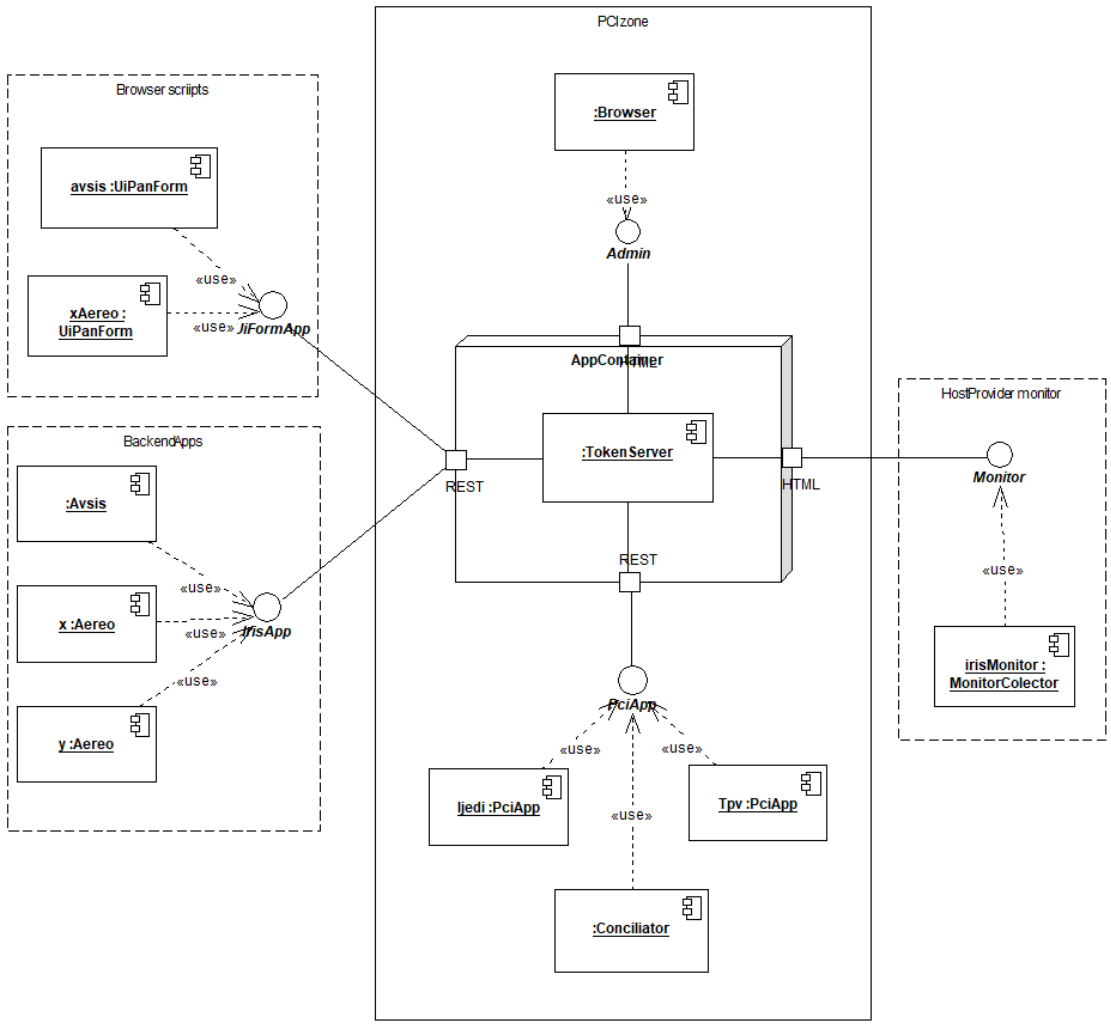

Figura 35. Diagrama de despliegue

El servidor de tokens estará dispuesto en la zona de red PCI, separado de otras zonas de la red. Cada interfaz ofrece servicios a las máquinas ubicadas en cada una de ellas:

- Zona PCI: alberga las aplicaciones que usan la interfaz PCIApp, es decir, aquellas que necesitan recuperar el PAN. Estará configurada conforme a los requisitos de red y seguridad exigidos por PCI-SS. Nótese que la interfaz de usuario para administración está ubicada en esta zona, no será por tanto accesible desde el exterior y quedará blindado su acceso por los requisitos de la red PCI.
- Zona BackendApps: alberga a los sistemas que ejecutan aplicaciones que gestionan datos de pago de tarjetas de crédito. Su configuración de seguridad puede ser similar a la exigida por PCI, aunque no están afectadas por la especificación al no manejar el PAN.
- Zona Browser Scripts: es una zona ficticia que representa a los scripts que se ejecutarán en los navegadores para hacer la petición de transformación de PAN a token. Los scripts que se ejecuten procederán de la zona PCI.
- Zona HostProviderMonitor: representa la zona de red que solo permite el acceso al sistema de monitorización y a esa interfaz.

5.2 DISEÑO DE CLASES

A continuación se detalla cual ha sido el diseño del encriptador, puesto que se trata de una parte importante del proyecto que se encarga de las tareas de encriptación y de la gestión y almacenamiento seguro de las claves de encriptación utilizadas.

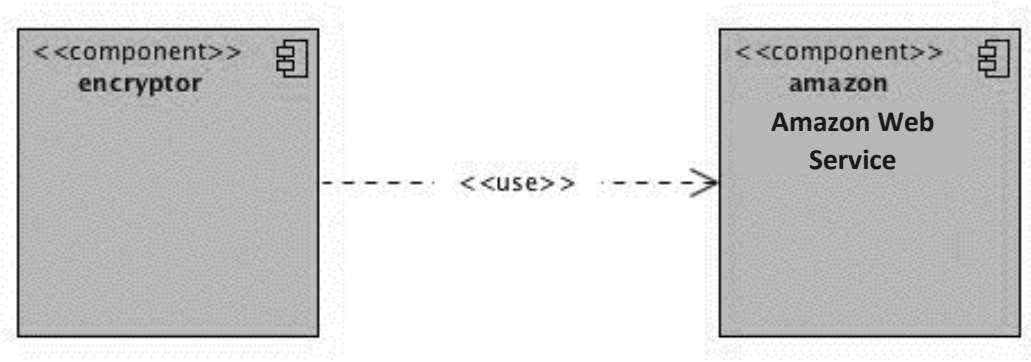

Figura 36. Diagrama encryptor

En el diagrama que se muestra a continuación se puede observar como el principal objetivo a la hora de realizar la implementación de este sistema era que se convirtiese en una abstracción completa de los servicios ofrecidos desde Amazon Web Service, AWS. La intención es que fuese lo más fácil de utilizar posteriormente desde el servidor de tokenización y que permitiese poder utilizarse sin conocer detalles sobre cómo funciona el sistema de encriptación.

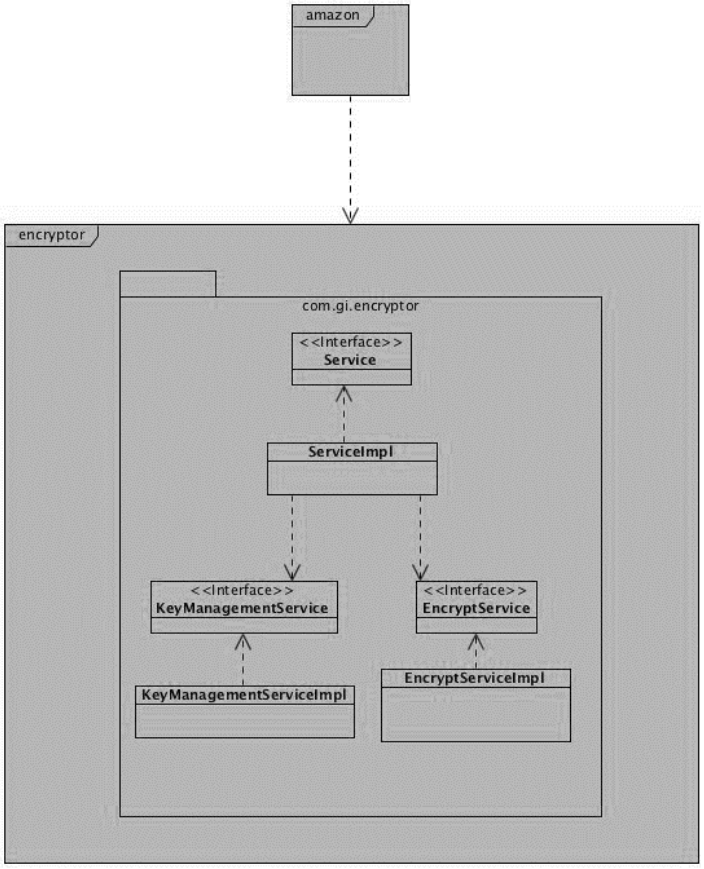

Figura 37. Diagrama de clases encryptor

5.3 DIAGRAMAS DE SECUENCIA

A continuación se muestra el diagrama que especifica la petición que una aplicación externa realiza al servidor de tokens para convertir un PAN a token.

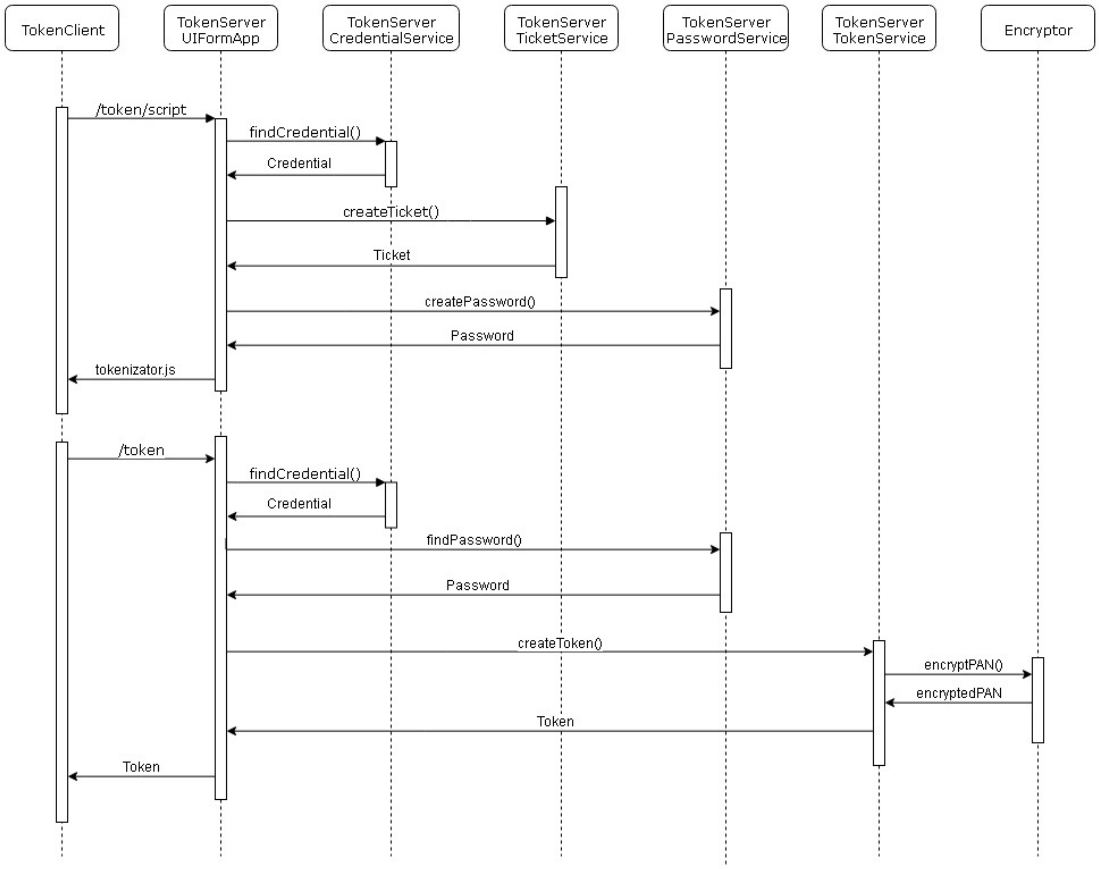

Figura 38. Diagrama de secuencia de creación de token

Como se puede observar, en este caso de uso intervienen los tres subsistemas identificados en

Identificación de los subsistemas. Además, el proceso se subdivide en dos operaciones:

OBTENCIÓN DEL JAVASCRIPT

La aplicación backend, a la hora de cargar su formulario de solicitud de los datos de la tarjeta, deberá solicitar al servidor de tokens que le proporcione el fichero JavaScript que se encargará de llevar a cabo la conversión de PAN a token cuando se envíe el formulario.

Para que esto sea posible, la aplicación deberá enviar al navegador del cliente lo que se ha denominado un ticket. El ticket tendrá la siguiente estructura:

$$UUID + "." + once$$

Siendo UUID el identificador de la credencial que le ha sido asignada por el servidor de tokens cuando el usuario administrador registró dicho permiso de acceso. Y *once* el resultado de encriptar, con el secreto que también le ha sido proporcionado por el servidor de tokens, tanto la credencial de acceso como el timestamp y un identificador de petición con el fin de evitar repeticiones.

$$once = enc\,(\,timestamp + "." + \text{UUID} + "." \text{ identificador}\,, secreto)$$

Este ticket, será enviado al servidor de tokens a través de la cabecera *Authentication* en la operación que solicita el código JavaScript cuando se carga el formulario en el navegador del cliente.

De esta forma, cuando el servidor de tokens recibe la llamada a través de la interfaz REST, utiliza el ticket para validar la petición:

- La primera parte del ticket, es decir, el identificador de la credencial sin encriptar, será utilizada para comprobar que se trata de una credencial registrada en el sistema, que se encuentra habilitada y que permite el acceso a la operación solicitada.
- Una vez identificada la credencial, el servidor de tokens es capaz de obtener el valor del secreto que comparten ambas aplicaciones. Este dato es utilizado para desencriptar la parte *once* del ticket y obtener así los tres valores que contiene: la fecha de creación, la identificación de la credencial y la identificación del ticket.
- Con el valor del identificador del ticket obtenido, el servidor de tokens comprueba que ese mismo identificador no ha sido utilizado previamente en el periodo de validez de este. En caso de que no esté registrado, se almacena para evitar la repetición peticiones y, en caso de que ya haya sido utilizado y aún no haya caducado, el servidor de tokens no proporcionará el fichero JavaScript a la aplicación.
- A continuación el servidor de tokens comprueba que las dos credenciales proporcionadas en el ticket coinciden. Si no es así la petición no será atendida.
- Por último, se utiliza el valor del timestamp para comprobar que la antigüedad de la petición no sobrepasa el límite establecido por configuración en el servidor de tokens.

Si todas las validaciones realizadas a partir del ticket son correctas, el servidor de tokens atiende la petición correctamente y devuelve a la aplicación el JavaScript que será utilizado a la hora de enviar el formulario con los datos de pago.

Antes de esto el sistema se encarga de generar una clave que se inserta en el código JavaScript y una contraseña que es devuelta en forma de cabecera en la respuesta enviada por el servidor de tokens.

SOLICITUD DE CONVERSIÓN DE PAN A TOKEN

El evento de envío del formulario será interceptado por el JavaScript proporcionado, de forma que antes de que los datos viajen a la aplicación, se invocará al servidor de tokens para que este se encargue de sustituir el valor del PAN.

El JavaScript se encarga de recoger el valor del PAN y realizar una doble encriptación:

- En primer lugar utilizará la clave proporcionada en el JavaScript como clave de encriptación.
- A continuación, el resultado obtenido en el paso anterior, se encripta utilizando la contraseña proporcionada en la respuesta.

Por lo tanto, en la petición de conversión solicitada al servidor de tokens, se envía:

- El identificador de la credencial.
- Una cabecera que contiene el PAN encriptado.
- Una cabecera que contiene la contraseña proporcionada por el servidor de tokens.

Si la petición es enviada correctamente, el servidor realizará las siguientes validaciones:

- La credencial proporcionada existe, está habilitada y permite el acceso a la operación de creación de tokens.
- La contraseña proporcionada existe, se corresponde con la credencial y no está caducada.

Llegados a este punto, el último paso que debe realizar el servidor de tokens, es desencriptar el valor del PAN para invocar al subsistema de encriptación y que este se encargue de proporcionarle el valor final del PAN encriptado.

El encriptador se conectará a Amazon Web Service Encryption para realizar el algoritmo de cifrado de acuerdo con los estándares definidos por PCI – DSS. Una vez obtenido el valor, el servidor crea un nuevo token y se lo proporciona al navegador en forma de campo oculto.

De esta forma la aplicación backend no llega a obtener en ningún momento el valor del PAN y así este permanece asegurado.

5.4 DISEÑO DE LA BASE DE DATOS

La principal función del presente apartado es describir el sistema de gestión de bases de datos utilizado para el correcto funcionamiento de la aplicación, así como la integración de este dentro del sistema, las tablas que se han tenido en cuenta y las relaciones entre estas.

5.4.1 Descripción del SGBD usado

El sistema de gestión de base de datos utilizado para el desarrollo de la base de datos del sistema ha sido HSQLDB o Hyperthreaded Structured Query Language Database. Se trata de un sistema libre, desarrollado en Java y basado en HypersonicSQL.

Las principales características que presenta dicho gestor son:

- Completo sistema gestor de bases de datos relacional.
- Tiempo de arranque mínimo y gran velocidad en las operaciones SELECT, INSERT, DELETE Y UPDATE.
- Sintaxis SQL estándar.
- Integridad referencial
- Procedimientos almacenados en Java.
- Triggers.
- Tablas en disco de hasta 8GB.

5.4.2 Integración del SGBD en el sistema

Habitualmente es necesario generar gran cantidad de código que permita al sistema interactuar con la base de datos, puesto que existen grandes discordancias entre los modelos relaciones y los objetos, como por ejemplo la granularidad de los objetos o la utilización de herencia, etc.

En este caso, el sistema de gestión de bases de datos utilizado se encuentra integrado a la perfección en el sistema, para ello se utiliza la API de persistencia de Java (JPA), que permite construir un sencillo puente entre el modelo de objetos Java y el modelo relacional, disminuyendo considerablemente la carga de trabajo necesaria para poder interactuar con la base de datos.

Existen varias implementaciones que permiten aprovechar las ventajas proporcionadas por JPA, en este caso se utiliza Eclipselink, que permite el mapeo de atributos entre la base de datos relacional y el modelo de objetos, se encarga de convertir los datos entre los tipos utilizados por Java y los que utiliza SQL.

Con intención de que la capa del modelo de dominio se encuentre lo más desacoplada posible y no cuente con ninguna dependencia al sistema de gestión de base de datos, las configuraciones se han llevado a cabo a través de ficheros XML.

5.4.3 Diagrama Entidad-Relación

En último lugar se presenta el diagrama Entidad-Relación que permite conocer, de forma sencilla, las tablas que conforman el modelo relacional, así como las relaciones existentes entre ellas.

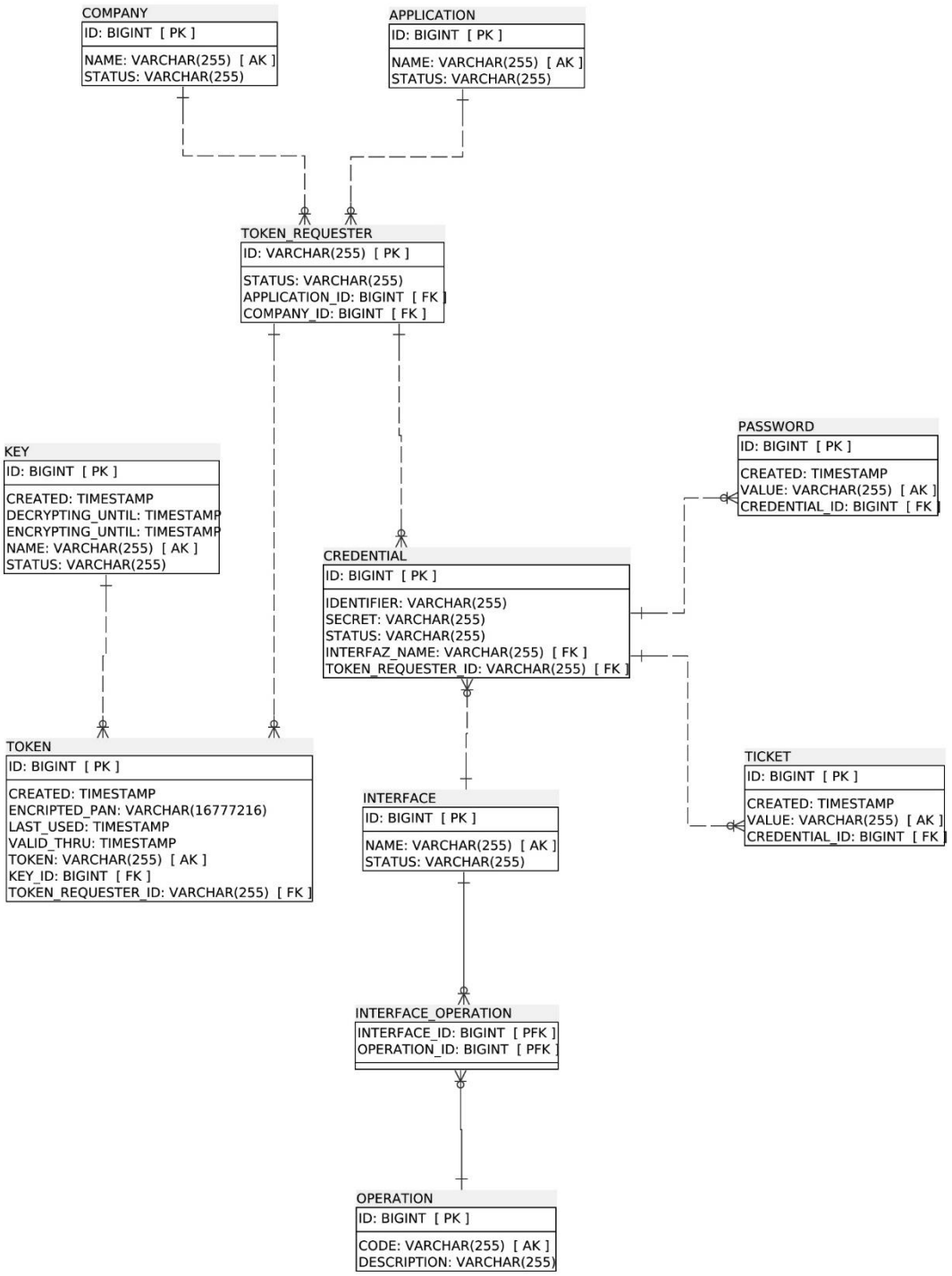

Figura 39. Diagrama entidad relación

5.5 DISEÑO DE LA INTERFAZ

A continuación se muestra la interfaz de usuario definitiva con la que cuenta la aplicación.

INICIO DE SESIÓN

Figura 40. Interfaz de inicio de sesión

GESTIÓN DE APLICACIONES

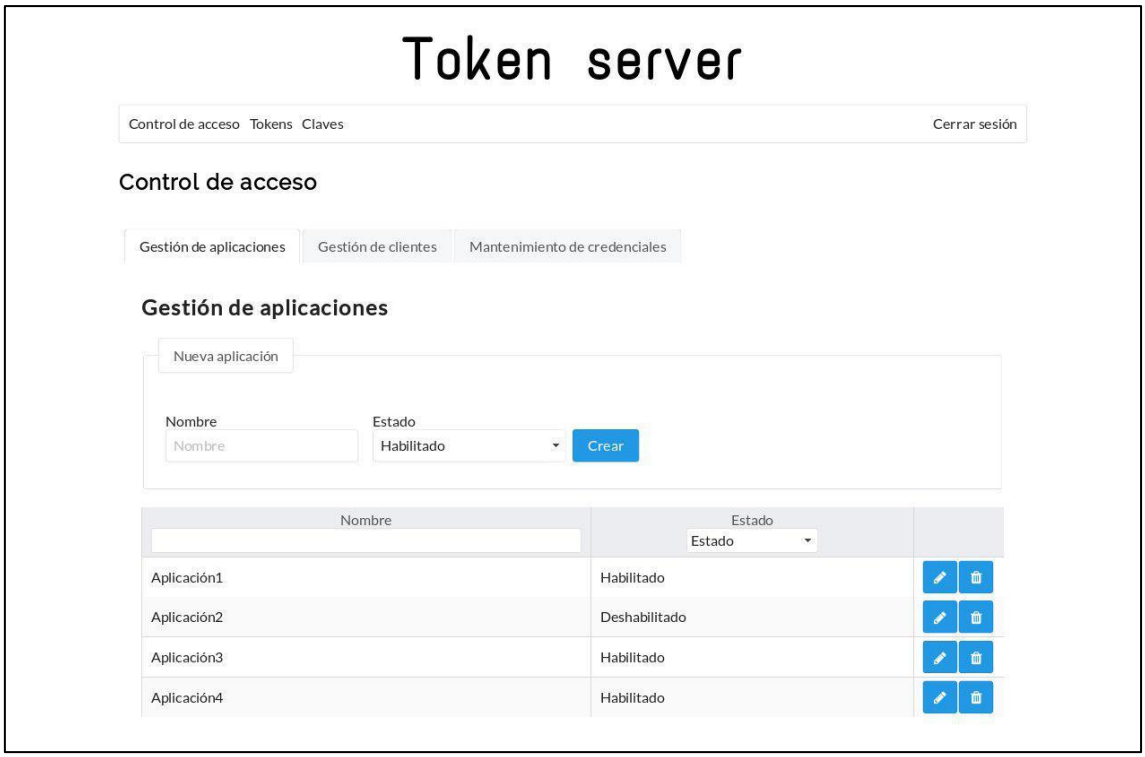

Figura 41. Interfaz de gestión de aplicaciones

GESTIÓN DE COMPAÑÍAS

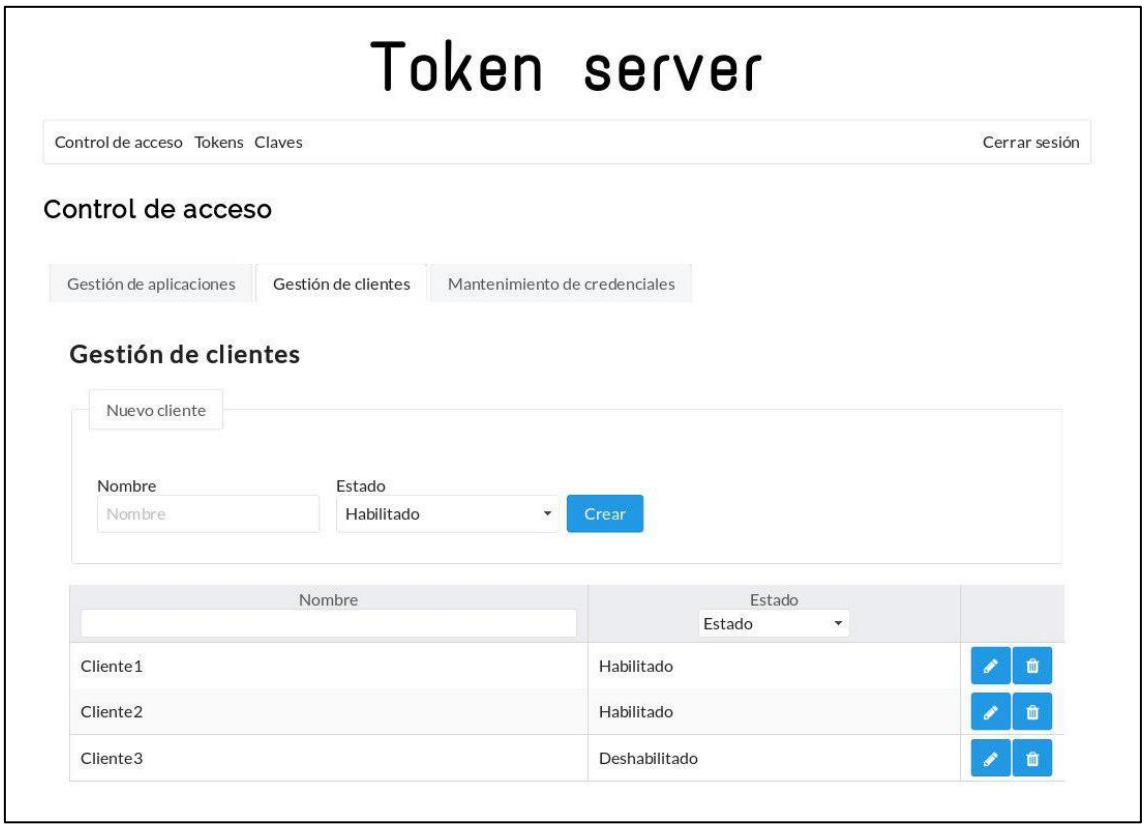

Figura 42. Gestión de clientes

GESTIÓN DE CREDENCIALES

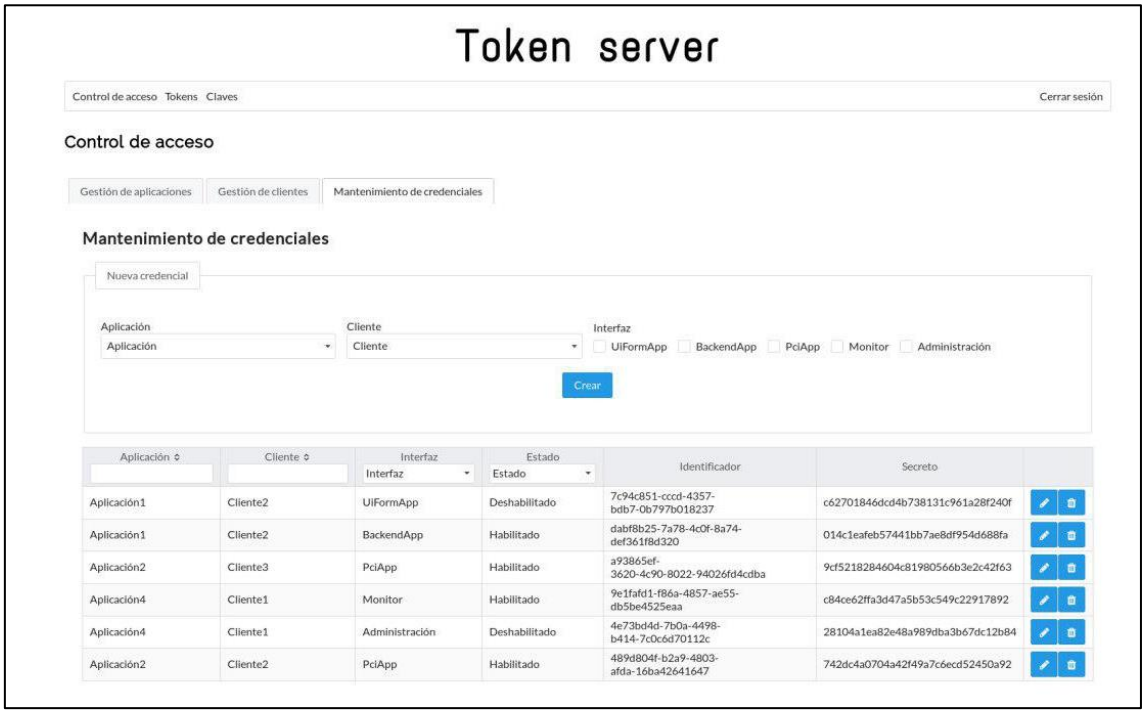

Figura 43. Gestión de credenciales

GESTIÓN DE TOKENS

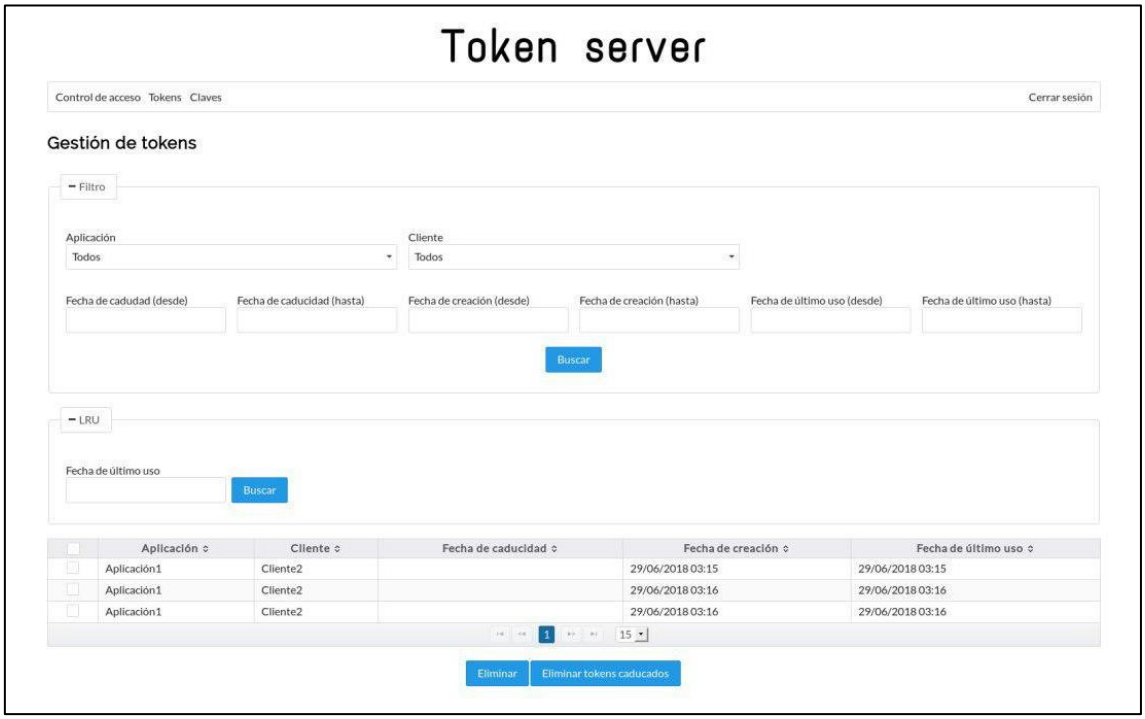

Figura 44. Interfaz de gestión de tokens

GESTIÓN DE CLAVES

Figura 45. Interfaz de gestión de claves

6. IMPLEMENTACIÓN DEL SISTEMA

6.1 ESTÁNDARES Y NORMAS SEGUIDOS

JAVA CODE CONVENTIONS

Las convenciones de codificación de Java establecen una serie de reglas que comprenden:

- Nombres de archivos
- Organización de directorios
- Identación
- Comentarios
- Declaraciones
- Sentencias
- Espacios en blanco
- Nomenclaturas
- Prácticas de programación

6.2 LENGUAJES DE PROGRAMACIÓN

JAVA

El lenguaje de programación utilizado para el desarrollo del proyecto ha sido Java, se trata del lenguaje de programación orientado a objetos más utilizado.

Ha sido el principal lenguaje de programación utilizado para la implementación de los tres sistemas desarrollados: la aplicación cliente, el servidor de tokens y el adaptador de encriptación.

Si bien se trata del principal lenguaje empleado, a continuación se describen brevemente otros lenguajes utilizados:

JPQL

Se trata de un lenguaje de consultas diseñado para combinar la simplicidad de la semántica y sintaxis de SQL con la expresividad de un lenguaje orientando a objetos. Permite al programador abstraerse de la base de datos utilizada, puesto que permite expresar las consultas en términos de entidades en lugar de sobre el modelo de base de datos.

XHTML

Se trata de un lenguaje de marcado que pretende reemplazar el clásico HTML, puesto que cada vez más herramientas son incompatibles con este sistema, la idea que proporciona XHTML es la de combinar la sintaxis HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos.

XHTML pretende proporcionar soporte para la creación de una web semántica, donde la información y la forma de presentarla se encuentren claramente diferenciadas.

JAVASCRIPT

Puesto que algunas de las acciones del sistema deben llevarse a cabo en el navegador del cliente, se ha utilizado JavaScript. Este lenguaje puede interactuar con el código HTML, permitiendo utilizan contenido dinámico.

Además, como lenguajes complementarios se han utilizado:

- SQL, como lenguaje de consultas a la base de datos.
- CSS, como lenguaje para definir la presentación de los documentos XHTML.

6.3 HERRAMIENTAS Y PROGRAMAS USADOS EN EL DESARROLLO

6.3.1 Herramientas

Para poder llevar a cabo el proyecto, se han utilizado las siguientes herramientas de apoyo.

DRAW.IO

Los diagramas UML han sido parte importante en el desarrollo del proyecto y, para su creación, se ha hecho uso de esta aplicación que resulta muy sencilla y fácil de aprender.

Se trata de una herramienta online de elaboración de diagramas, totalmente gratuita, que no necesita ningún tipo de instalación y muy completa, puesto que soporta los diagramas UML más utilizados en el desarrollo software habitualmente.

BALSAMIQ

Uno de los pasos importantes llevados a cabo durante la fase de análisis fue la realización de unos pequeños esbozos que mostrasen la distribución de las diferentes funcionalidades proporcionadas por la interfaz web de administración de la aplicación.

Para ello se ha utilizado esta herramienta web que facilita y agiliza la creación de dichos bocetos o prototipos de interfaz de usuario. Una de las principales ventajas que ofrece es que cuando con muchos objetos de interfaz de usuario prediseñados como barras de menú, botones, elementos de formulario, etc. Por lo que el proceso se simplifica bastante, además, los resultados obtenidos son muy satisfactorios.

6.3.2 Programas

NETBEANS

Se trata de un entorno de desarrollo libre y desarrollo principalmente en Java. Permite el desarrollo de aplicaciones a partir de un conjunto de componentes software denominados módulos. Esta herramienta de desarrollo permite la elaboración de todo tipo de aplicaciones Java.

NOTEPAD++

Notepad++ es un editor de texto y de código con soporte para múltiples lenguajes de programación, por lo tanto, se ha utilizado como herramienta de soporta para le edición de diferentes tipos de ficheros: sql, xhtml, css, etc.

MICROSOFT OFFICE

Paquete ofimático utilizado para construir los diferentes documentos que rodean al proyecto.

- Word, para la creación del presente proyecto.
- Excel, para la elaboración del presupuesto del proyecto.
- Project, para la creación de la planificación temporal del proyecto.
- PowerPoint, para la creación de la presentación oral del proyecto.

WBS SHEDULE PRO

WBS Shedule Pro es un software de gestión de proyecto de planificación basado en Windows, que combina una estructura de trabajo con gráficos WBS, diagramas de red, hojas de tareas, además de varias funciones adicionales para producir una herramienta de planificación y gestión de proyectos eficiente.

6.4 CREACIÓN DEL SISTEMA

6.4.1 Descripción detallada de las clases

A continuación se realiza una descripción de las principales clases que conforman el subsistema encryptor, con intención de clarificar las operaciones que este componente proporciona al servidor de tokens.

KEY MANAGEMENT SERVICE

La responsabilidad de esta clase es la de gestionar las claves de encriptación que se utilizan en el sistema y que se encuentran almacenadas en el almacén de claves proporcionado por AWS.

Métodos			
Acceso	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Público	Void	createKey	descripción: String alias: String
Público	Void	enabledKey	keyId: String
Público	Void	disabledKey	keyId: String
Público	Void	updateAlias	keyId: String alias: String
Privado	Cadena de texto	getKeyId	alias: String
Público	Void	deleteKey	alias: String

Tabla 47. Key Management Service

ENCRYPT SERVICE

Esta clase se encarga de llevar a cabo los procesos de criptografía.

Métodos			
Acceso	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Público	String	encrypt()	text : String key: String
Público	String	decrypt()	text: String key: String
Público	String	reencrypt()	text: String oldKey: String newKey: String

Tabla 48. Encrypt Service

7. DESARROLLO DE LAS PRUEBAS

A continuación se presentan los resultados obtenidos durante la fase de pruebas diseñadas durante el análisis del sistema.

CASO DE USO: ACCESO A LA INTERFAZ DE ADMINISTRACIÓN

Caso de prueba	Entrada	Resultado obtenido
CP1.1.1	Acceso a la aplicación con un usuario y contraseña asociados a un usuario administrador.	☑
CP1.1.2	Acceso a la aplicación con un identificador de usuario que no se encuentra registrado en el sistema.	☒ El sistema no permite el acceso, pero no muestra un mensaje de error si no que aparece una pantalla de error, puesto que el sistema intentaba acceder a una página de error no implementada.
Acciones	En caso de que haya un error al realizar el inicio de sesión se dirige al usuario a la misma página de inicio y se añade un mensaje en la parte superior de la pantalla.	
CP1.1.3	Acceso a la aplicación con una contraseña inválida.	☒ El sistema no permite el acceso, pero no muestra un mensaje de error si no que aparece una pantalla de error, puesto que el sistema intentaba acceder a una página de error no implementada.
Acciones	Las acciones llevadas a cabo en el caso de prueba anterior, solventan también este problema.	

Tabla 49. Resultados casos de prueba de acceso a la interfaz de administración

GESTIÓN DE APLICACIONES

Caso de prueba	Entrada	Resultado obtenido
CP2.1.1	En la creación de una aplicación, se proporciona un nombre que no se encuentra registrado en el sistema.	☑
CP2.1.2	En la creación de una aplicación, no se proporciona un nombre.	☒ El sistema crea la aplicación con el nombre vacío.
Acciones	Se hace que el campo nombre sea obligatorio y se realiza una validación antes de la creación de una nueva aplicación.	

Caso de prueba	Entrada	Resultado obtenido
CP2.1.3	En la creación de una aplicación, se proporciona un nombre que ya se encuentra asociado a otra aplicación registrada previamente.	☑
CP2.2.1	Se selecciona la opción de gestión de aplicaciones en el menú.	☑
CP2.2.2	En el listado de aplicaciones, se introduce un valor en el filtro que no se corresponde con ninguna aplicación	☑
CP2.2.3	En el listado de aplicaciones, se introduce un valor en el filtro que se corresponde con varias aplicaciones.	☑
CP2.3.1	Se modifica el estado de una aplicación.	☑
CP2.4.1	Se selecciona la opción de borrado sobre una de las aplicaciones a las que se tiene acceso a través del listado y se confirma la acción. La aplición seleccionada no cuenta con ningún token activo.	☑
CP2.4.2	Se selecciona la opción de borrado sobre una de las aplicaciones a las que se tiene acceso a través del listado y se confirma la acción. La aplición seleccionada cuenta con varios token activos.	☑

Tabla 50. Resultado casos de prueba de gestión de aplicaciones

GESTIÓN DE COMPAÑÍAS

Caso de prueba	Entrada	Resultado obtenido
CP3.1.1	En la creación de una compañía, se proporciona un nombre que no se encuentra registrado en el sistema.	☑
CP3.1.2	En la creación de una compañía, no se proporciona un nombre.	☒ El sistema crea la compañía con el nombre vacío.
Acciones	Se hace que el campo nombre sea obligatorio y se realiza una validación antes de la creación de una nueva compañía.	
CP3.1.3	En la creación de una compañía, se proporciona un nombre que ya se encuentra asociado a otra compañía registrada previamente.	☑

CP3.2.1	Se selecciona la opción de gestión de compañías en el menú.	☑
CP3.2.2	En el listado de compañías, se introduce un valor en el filtro que no se corresponde con ninguna compañía.	☑
CP3.2.3	En el listado de compañías, se introduce un valor en el filtro que se corresponde con varias compañías.	☑
CP3.3.1	Se modifica el estado de una compañía.	☑
CP3.4.1	Se selecciona la opción de borrado sobre una de las compañías a las que se tiene acceso a través del listado y se confirma la acción. La compañía seleccionada no cuenta con ningún token activo.	☑
CP3.4.2	Se selecciona la opción de borrado sobre una de las compañías a las que se tiene acceso a través del listado y se confirma la acción. La compañía seleccionada cuenta con varios token activos.	☑

Tabla 51. Resultado casos de prueba de gestión de compañías

GESTIÓN DE ACCESOS

Caso de prueba	Entrada	Resultado obtenido
CP4.1.1	En la creación de un nuevo permiso de acceso, se selecciona una aplicación, un cliente y una única interfaz.	☑
CP4.1.2	En la creación de un nuevo permiso de acceso, se selecciona una aplicación, un cliente y tres interfaces.	☑
CP4.2.1	Se selecciona la opción de gestión de credenciales en el menú.	☑
CP4.2.2	En el listado de credenciales, se introduce un valor en el filtro que no se corresponde con ninguna credencial.	☒ El mensaje informando sobre la ausencia de credenciales que cumplan los criterios no se muestra.
Acciones	El mensaje no se mostraba correctamente, puesto que no había sido incluido en el fichero de propiedades. Tras incluirlo, se muestra correctamente.	
CP4.2.3	En el listado de credenciales, se introduce un valor en el filtro que se corresponde con varias credenciales de acceso.	☑

Caso de prueba	Entrada	Resultado obtenido
CP4.3.1	Se deshabilita una credencial de acceso, modificando el estado de esta.	☑
CP4.4.1	Se selecciona la opción de borrado sobre una de las credenciales a las que se tiene acceso a través del listado y se confirma la acción.	☑

Tabla 52. Resultado casos de prueba de gestión de accesos

GESTIÓN DE CLAVES

Caso de prueba	Entrada	Resultado obtenido
CP5.1.1	Se selecciona la opción de gestión de claves en el menú.	☑
CP5.1.2	En el listado de claves, se introduce un valor en el filtro que no se corresponde con ninguna clave.	☑
CP5.1.3	En el listado de claves, se introduce un valor en el filtro que se corresponde con varias claves de encriptación.	☑
CP5.2.1	Se selecciona una clave de encriptación para su renovación manual.	☑

Tabla 53. Resultado casos de prueba de gestión de claves

GESTIÓN DE TOKENS

Caso de prueba	Entrada	Resultado obtenido
CP6.1.1	Se accede a la opción de gestión de tokens desde el menú principal y, sin seleccionar ningún valor en los filtros propuestos, se realiza la búsqueda.	☒ El sistema muestra un error debido a que no se ha indicado un campo obligatorio.
Acciones	El campo de fecha límite de la búsqueda de menos recientemente usados está marcado como obligatorio, sin embargo, en este caso de uso, ese valor no es obligatorio y estaba impediente la correcta ejecución del proceso. Se llevan a cabo las medidas de validación necesarias para que dicho capo requerido únicamente se tenga en cuenta en el caso de la búsqueda correspondiente.	
CP6.1.2	Se accede a la opción de gestión de tokens desde el menú principal, se seleccionan varios filtros de entre los propuestos y se realiza la búsqueda.	☒ El sistema muestra un error debido a que no se ha indicado un campo obligatorio.
Acciones	El problema y las medidas tomadas son las mismas que el caso de prueba anterior.	

Caso de prueba	Entrada	Resultado obtenido
CP6.2.1	Se selecciona la opción de búsqueda de tokens menos recientemente usados, sin indicar la fecha límite.	☑
CP6.2.2	Se selecciona la opción de búsqueda de tokens menos recientemente usados, indicando una fecha límite.	☑
CP6.3.1	Se selecciona un token en el listado y se elimina.	☑
CP6.3.2	Se seleccionan varios tokens en el listado y se eliminan.	☑
CP6.4.1	Se selecciona la opción de eliminar los tokens caducados.	☑

Tabla 54. Resultado casos de prueba de gestión de tokens

CREACIÓN DE TOKENS

Caso de prueba	Entrada	Resultado obtenido
CP7.1.1	Solicitar una creación de token con una credencial que no se encuentra registrada en el sistema.	☑
CP7.1.2	Solicitar una creación de token con una credencial que se encuentra deshabilitada en el sistema.	☑
CP7.1.3	Solicitar una creación de token utilizando un secreto que no se encuentra asociado a la credencial.	☑
CP7.1.4	Solicitar una creación de token utilizando un ticket que ya ha sido utilizado previamente.	☑
CP7.1.5	Solicitar una creación de token utilizando una credencial que no cuenta con permiso para realizar la operación.	☑
CP7.1.6	Solicitar una creación de token utilizando una contraseña inválida.	☑
CP7.1.7	Solicitar una creación de token utilizando una contraseña caducada.	☒ El token se ha creado correctamente.
Acciones	No se había tenido en cuenta la caducidad de las contraseñas, por lo tanto, se añade una nueva validación que compruebe si la fecha de caducidad de la contraseña no ha sido excedida.	

Caso de prueba	Entrada	Resultado obtenido
CP7.1.8	Solicitar una creación de token utilizando una contraseña válida, en vigencia, pero que no está asociada a la credencial con la qu se autentifica la aplicación.	☑
CP7.1.9	Solicitar una creación de token utilizando una credencial válida, encriptada con el secreto asociado a dicha credencial. El ticket que se envía no ha sido utilizado previamente y se utiliza la contraseña proporcionada por el servidor de token durante su periodo de vigencia.	☑

Tabla 55. Resultado casos de prueba de creación de tokens

ELIMINACIÓN DE TOKEN

Caso de prueba	Entrada	Resultado obtenido
CP8.1.1	Solicitar la eliminación de un token con una credencial que no se encuentra registrada en el sistema.	☑
CP8.1.2	Solicitar la eliminación de un token con una credencial que se encuentra deshabilidatada en el sistema.	☑
CP8.1.3	Solicitar la eliminación de un token utilizando una credencial que no cuenta con permiso para realizar la operación.	☑
CP8.1.4	Solicitar la eliminación de un token que no existe.	☑
CP8.1.5	Solicitar la eliminación de un token utilizando una credencial válida, con permiso para realizar esta operación y con un token correcto.	☑

Tabla 56. Resultado casos de prueba de eliminación de token

OBTENCIÓN DE PAN ENMASCARADO

Caso de prueba	Entrada	Resultado obtenido
CP9.1.1	Solicitar la obtención de PAN con enmascaramiento corto utilizando una credencial que no existe en el sistema.	☑
CP9.1.2	Solicitar la obtención de PAN con enmascaramiento corto utilizando una credencial que se encuentra deshabilitada.	☑

Caso de prueba	Entrada	Resultado obtenido
CP9.1.3	Solicitar la obtención de PAN con enmascaramiento corto utilizando una credencial que no cuenta con permiso para realizar la operación.	☑
CP9.1.4	Solicitar la obtención de PAN con enmascaramiento corto para un token que no existe.	☑
CP9.1.5	Solicitar la obtención de PAN con enmascaramiento corto con una credencial válida para un token existente en el sistema.	☑
CP9.2.1	Solicitar la obtención de PAN con enmascaramiento medio utilizando una credencial que no existe en el sistema.	☑
CP9.2.2	Solicitar la obtención de PAN con enmascaramiento medio utilizando una credencial que se encuentra deshabilitada.	☑
CP9.2.3	Solicitar la obtención de PAN con enmascaramiento medio utilizando una credencial que no cuenta con permiso para realizar la operación.	☑
CP9.2.4	Solicitar la obtención de PAN con enmascaramiento medio para un token que no existe.	☑
CP9.2.5	Solicitar la obtención de PAN con enmascaramiento medio con una credencial válida para un token existente en el sistema.	☒ Se muestran los últimos cuatro dígitos del PAN, en lugar de los seis primeros.
Acciones	Debido a un error de programación, el caso de enmascaramiento medio ha sido tratado como si fuese una solicitud de obtención de PAN con máscara corta. Se solventa el problema contemplando correctamente el caso del enmascaramiento medio.	
CP9.3.1	Solicitar la obtención de PAN con enmascaramiento largo utilizando una credencial que no existe en el sistema.	☑
CP9.3.2	Solicitar la obtención de PAN con enmascaramiento largo utilizando una credencial que se encuentra deshabilitada.	☑
CP9.3.3	Solicitar la obtención de PAN con enmascaramiento largo utilizando una credencial que no cuenta con permiso para realizar la operación.	☑
CP9.3.4	Solicitar la obtención de PAN con enmascaramiento largo para un token que no existe.	☑

Caso de prueba	Entrada	Resultado obtenido
CP9.3.5	Solicitar la obtención de PAN con enmascaramiento largo con una credencial válida para un token existente en el sistema.	☑

Tabla 57. Resultado casos de prueba de obtención de PAN enmascarado

OBTENCIÓN DE PAN

Caso de prueba	Entrada	Resultado obtenido
CP10.1.1	Solicitar la obtención de PAN utilizando una credencial que no existe en el sistema.	☑
CP10.1.2	Solicitar la obtención de PAN utilizando una credencial que se encuentra deshabilitada.	☑
CP10.1.3	Solicitar la obtención de PAN utilizando una credencial que no cuenta con permiso para realizar la operación.	☑
CP10.1.4	Solicitar la obtención de PAN para un token que no existe.	☑
CP10.1.5	Solicitar la obtención de PAN una credencial válida para un token existente.	☑

Tabla 58. Resultado casos de prueba de obtención de PAN

ROTACIÓN DE CLAVE

Caso de prueba	Entrada	Resultado obtenido
CP11.1.1	Se programa el timer para que se ejecute a una hora prefijada.	☑

Tabla 59. Resultado casos de prueba de rotación de clave

LIMPIEZAS

Caso de prueba	Entrada	Resultado obtenido
CP12.1.1	Se realiza una consulta para comprobar el número de tokens caducados registrados en el sistema. Se ejecuta el timer de borrado.	☑
CP12.2.1	Se realiza una consulta para comprobar el número de tickets caducados registrados en el sistema. Se ejecuta el timer de borrado.	☑
CP12.3.1	Se realiza una consulta para comprobar el número de contraseñas caducadas registradas en el sistema. Se ejecuta el timer de borrado.	☑

Tabla 60. Resultado casos de prueba de timers de limpieza

8. MANUALES DEL SISTEMA

8.1 MANUAL DE INTEGRACIÓN

El propósito de este manual es detallar cual es el proceso que deben seguir las aplicaciones afectadas para comenzar a utilizar el servidor de tokens desarrollado.

El primera paso que se deberá realizar es incluir el código JavaScript que se muestra en la siguiente figura en la vista que se encarga de mostrar el formulario donde el usuario debe proporcionar los datos de la tarjeta. Es necesario incluir este código de forma que se ejecute cuando se carga el formulario.

El parámetro URL se corresponde con el host en el que estará ubicado el servidor de tokens, por lo tanto será necesario ponerse en contacto con el administrador del sistema para que proporcione la URL adecuada.

```
<script>
    var password;
    $(function () {
        $.ajax({
            url: 'http://tokenserver/ws/uiformapp/rs/token/script',
            headers: {'Authorization': '#{ticket}'}
        })
                .done(function (response, textStatus, xhr) {
                    password = xhr.getResponseHeader("Authorization");
                    var s = document.createElement("script");
                    s.type = "text/javascript";
                    s.innerHTML = response;
                    $("body").append(s);
                });
    });
</script>
```

Figura 46. Código JavaScript aplicación backend

Como se puede observar en el anterior fragmento de código, se utiliza la cabecera *Authorization* para que la aplicación envíe al servidor de tokens el ticket con su información. La aplicación backend debe encontrar la forma de proporcionar esta información de la manera que le resulte más cómoda.

A continuación se muestra un ejemplo desarrollado para el presente proyecto. Cada aplicación en su caso, debería modificar los valores de credencial y secreto por aquellos que le hayan sido proporcionados por el administrador del servidor de tokens. Además, se recomienda no ubicar estos datos directamente en el código.

Para realizar el encriptado es necesario utilizar JSON Web Tokens, sin embargo, queda a elección de cada aplicación que librería se utiliza para generar el JWT.

```
public class TicketController {

    private final static String CREDENTIAL = "7c94c851-cccd-4357-bdb7-0b797b018237";
    private final static String SECRET = "c62701846dcd4b738131c961a28f240f";

    public String getTicket() {
        String token = "";
        String once = LocalDateTime.now().truncatedTo(ChronoUnit.SECONDS).toString()
                + "." + CREDENTIAL
                + "." + UUID.randomUUID().toString();
        try {
            Algorithm algorithm = Algorithm.HMAC256(SECRET);
            token = JWT.create()
                    .withClaim("ticket", once)
                    .sign(algorithm);
        } catch (IllegalArgumentException | UnsupportedEncodingException ex) {
            Logger.getLogger(TicketController.class.getName()).log(Level.SEVERE, null, ex);
        }
        return CREDENTIAL + "." + token;
    }

}
```

Figura 47. Generación del ticket

Otro detalle que las aplicaciones deben modificar se corresponde con datos del formulario que solicita los datos.

- El atributo *id* del formulario debe ser exactamente “payForm”, puesto que este será el elemento que el servidor de tokens buscará para poder incluir el valor del token.
- Se debe incluir la llamada a la función *takingOff(password)* que será la encarga de realizar la solicitud de conversión de PAN a token cuando se envíe el formulario por parte del usuario.

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:c="http://java.sun.com/jsp/jstl/core">

    <h:head>
        <title>Token client</title>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

    </h:head>

    <h:body>
        <c:set var="ticket" value="#{ticketController.getTicket()}" />
        <h:form id="payForm" onsubmit="return takingOff(password);">
            <label>Titular de la Tarjeta:</label>
            <input type="text" maxlength="50" required="required"/><br/>
            <label>Número de tarjeta:</label>
            <input type="text" id="paNumber" maxlength="16"
                   required="required" pattern="[0-9]{15,16}"/><br/>
            <label>CVC2:</label>
            <input type="text" maxlength="4" required="required" pattern="[0-9]{3,4}"/><br/>
            <label>Fecha de caducidad:</label>
            <input type="text" maxlength="2" placeholder="mm" required="required"
                   pattern="(0[1-9]|1[012])"/>
            <input type="text" maxlength="2" placeholder="aa" required="required"
                   pattern="[1-9]{2}" /><br/>
            <button type="submit">Pagar</button>
        </h:form>
        <script ...16 lines />
    </h:body>
</html>
```

En último lugar, el servidor de tokens proporcionará el token ubicándolo en un elemento hidden del formulario cuyo atributo *name* será “panToken”. Será por tanto el valor de este elemento el que la aplicación backend deberá recoger para acceder el token.

9. CONCLUSIONES Y AMPLIACIONES

9.1 CONCLUSIONES

El desarrollo de este proyecto me ha permitido obtener cierta retroalimentación, tanto técnica como personal.

En cuanto a la ejecución técnica, me ha servido para afianzar y encontrar un sentido más prácico a muchos de los contenidos del máster, así como para quitar temores a utilizar herramientas y tecnologías nuevas para mí.

Unos de los principales miedos que me surgieron cuando me propusieron la realización de este proyecto radicaba en que no se trataba de un tema que me llamase para nada mi atención. En cuanto a preferencias personales me decanto sin ninguna duda por el desarrollo frontend, el diseño de aplicaciones, etc. Algo que, desde luego, no tenía nada que ver con este proyecto, que incluía principalmente temas de seguridad, criptografía, redes, etc.

Sin embargo lo que en un principio parecía ser un problema, finalmente se ha convertido en un mecanismo para conocer más a fondo partes de la ingeniería informática que seguramente, sin el desarrollo de este proyecto, no habrían llegado a mi vida. Valoro como algo positivo el haberme involucrado en el desarrollo de un proyecto que me ha permitido acercarme a temas en los que no había pensado profundizar en ninguna ocasión.

En cuando a las conclusiones personales, he podido darme cuenta de la importancia que tiene organizarse de forma adecuada, realizar una planificación realista y lo más detallada posible. Así como igual de importante es hacer todo lo que esté en nuestras manos para poder cumplirla, la satisfacción que supone saber que estás realizando las tareas como tú habías planificado y que así no parezca que todo está siendo un desastre.

Uno de los aspectos más importantes que este proyecto me ha permitido descubrir, es aprender a relativizar la importancia de las cosas. En muchas ocasiones, a lo largo de estos meses de desarrollo, he invertido muchas horas en detalles que, al fin y al cabo, carecían de importancia y que, en muchos casos, ni si quiera serán valorados por los usuarios finales de la aplicación. Esto además ha conllevado una pérdida de tiempo de dedicación a otros aspectos que si resultaban importantes.

Por último, me gustaría destacar otra de las lecciones aprendidas durante el desarrollo de este proyecto y es la importancia de tener claro que quieres hacer, cuales son los objetivos que quieres fijarte en el proyecto. Al inicio del desarrollo invertí mucho tiempo vagando entre diferentes ideas, sin tener del todo claro que era exactamente lo que la aplicación tenía que hacer, cuál era el enfoque que le quería dar, desarrollando cosas que finalmente no llegué a completar por que más tarde decidí descartarlas, etc.

9.2 AMPLIACIONES

La principal ampliación que se presenta para el sistema consiste en el desarrollo del sistema de encriptación del PAN. Es decir, la idea a futuro, es prescindir de los servicios proporcionados por Amazon Web Service. Para ello será necesario desarrollar un almacén de claves de encriptación que cumpla con los requisitos de seguridad establecidos por la normativa PCI – DSS y, además, crear un sistema que se encargue de ejecutar las operaciones de encriptación y desencriptación del PAN de acuerdo a las directrices sobre criptografía que también se incluyen en la normativa.

10. BIBLIOGRAFÍA

Amazon. 2018. https://docs.aws.amazon.com/es_es/encryption-sdk/latest/developer-guide/java-example-code.html. [En línea] 06 de 2018.

—. 2018. https://docs.aws.amazon.com/es_es/kms/latest/developerguide/programming-keys.html. [En línea] 06 de 2018.

Auth0. https://jwt.io/. [En línea]

Council, PCI Security Standards. 2011. *PCI DSS Tokenization Guidelines.* 2011.

—. 2015. *Tokenization Product Security Guidelines.* 2015.

Council, PCI Security Standars. 2016. *Norma de seguridad de datos: Requisitos y procedimientos de evaluación de seguridad.* 2016.

HSQLDB. 2018. http://hsqldb.org/doc/guide/index.html. [En línea] 06 de 2018.

https://www.pcihispano.com/. [En línea]

Martin, Robert C. 2017. *Clean Architecture: A Craftsman's Guide to Software Structure and Design.* s.l. : Prentice Hall, 2017.

Peyrott, Sebastián. *JWT Handbook.* s.l. : Auth0.

The Clean Architecture. **Martin, Robert. 2012.** 2012.