Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA (SG)

Grado en Ingeniería Informática de Servicios y Aplicaciones

Gestor de contraseñas seguras

Alumno: Gilarranz Nieto Hugo

Tutor: Álvarez Bravo José Vicente

Gestor de contraseñas seguras (Secure Password Manager)

Gilarranz Nieto Hugo

“La llave del éxito es habituarse en tu
vida a hacer las cosas
qué tienes miedo a hacer”

Vincent van gogh

Agradecimientos

Quiero dar las gracias a todos los compañeros de clase con los que he tenido relación, en especial a mi "Pareja de baile" por aguantarme durante los trabajos en parejas, y tanto a mis compañeros de piso durante estos últimos meses como a la "Soliciata" por aguantarme en los momentos de estrés a lo largo de este proyecto.

A mi tutor José Vicente por su apoyo y guía a lo largo de estos momentos finales y a todo el resto del profesorado por lo que he aprendido de ellos.

Y por último y más importante a mis padres, ya que sin ellos esto no habría sido posible, en especial a mi madre por animarme a darle una segunda oportunidad a la universidad.

A todos vosotros muchas gracias.

Índice General

Lista de figuras

Lista de tablas

Resumen

Hoy en día con el incremento de los servicios online el número de cuentas que posee un usuario medio ha aumentado mucho y es difícil utilizar métodos tradicionales para recordar tantas contraseñas sin recurrir al uso de estas en varios lugares, creando así situaciones que comprometen la seguridad de dichas cuentas.
El objetivo principal de este TFG es el desarrollo de una aplicación de escritorio que permita gestionar contraseñas, es decir, que permita almacenar contraseñas, generar contraseñas seguras y hacer un seguimiento con el fin de poder realizar un cambio periódico de estas evitando la repetición de contraseñas usadas recientemente. Como objetivo secundario, se hará hincapié en la creación de una interfaz amigable que incentive su uso por parte de usuario no avanzados.

Palabras clave: contraseñas, java, AES, SHA

Abstract

Nowadays with the increase of online services, the number of accounts of the average users has augmented significantly and it is hard to use traditional way to remember all those passwords without ending up using the same password in different services, to the point where the security of those accounts is compromised.
The main goal of this project is the development of a desktop application that allows managing passwords, this means storage passwords, generating secure passwords and tracking with the objective of changing the password periodically evading the use of recent passwords. As a secondary goal, a friendly interface will be designed to encourage to use for non advanced users.

Keywords: password, java, AES, SHA

Parte I

Memoria del proyecto

Capítulo 1

Introducción

1.1 Introducción

Es habitual que hoy en día la gente utilice una única contraseña o como mucho un reducido set de ellas. Una pequeña modificación es el uso de algún tipo de contraseña básica obtenida a partir de una frase seguida de un elemento haciendo relación al servicio relacionado, aunque este tipo de modificación genera contraseñas menos débiles, cuando una contraseña es comprometida sigue comprometiendo el resto de ellas, además de ser una práctica poco habitual.

En el caso de guardarlas en papel, no nos será desconocido el extravío del soporte en papel con dichas contraseñas, o el de fotografias subidas a la red en las que se aprecian post-it o similares con contraseñas.

Para dar solución a los problemas de seguridad de estos métodos arcaicos, se propone utilizar la tecnología para solventar de manera eficaz todos estos problemas.

1.2 Motivación

Este proyecto está motivado por una creciente necesidad propiciada por el gran volumen de servicios que actualmente consume una persona que requieren el uso de un usuario y contraseña, siendo conveniente el centralizar todos estos datos en un único lugar, aportando una herramienta que permite almacenar convenientemente un número prácticamente ilimitado de contraseñas eliminando la necesidad de recordarlas, facilitando así el uso de contraseñas únicas para cada servicio, con unas ventajas notorias en seguridad y persistencia sobre soportes físicos tradicionales como el papel.

1.3 Objetivos

El objetivo principal de este trabajo es la creación de una aplicación de escritorio que permita centralizar la gestión de contraseñas reducción las contraseñas que el usuario final necesita a una sola, con un especial hincapié en la seguridad sin dejar completamente de lado la usabilidad.

Para lograr ese objetivo principal es necesario cumplir los siguientes objetivos:

OBJ-1: Identificación del usuario mediante una contraseña maestra.

OBJ-2: Almacenamiento de contraseñas en local bajo el algoritmo de codificación AES* y la clave maestra.

OBJ-3: Permitir crear contraseñas seguras mediante un algoritmo personalizable.

OBJ-4: Recordar al usuario de manera periódica el cambio de las contraseñas almacenadas.

OBJ-5: Recordar contraseñas anteriores para evitar el uso de estas en el futuro.

OBJ-6: Facilitar la inclusión de perfiles de usuario base para guardar contraseñas de aplicaciones no soportadas por la aplicación.

Además como objetivos secundarios para la mejora de la usabilidad se contemplarán los siguientes objetivos secundarios:

OBJS-1: Diseño e implementación de una interfaz gráfica que permita un uso fluido de la aplicación.
OBJS-2: Un sistema de exportación de las contraseñas almacenadas, para su posterior importación desde un navegador.
OBJS-3: Un sistema de exportación de las contraseñas almacenadas, para su posterior importación desde otro equipo que tenga instalada la aplicación.
OBJS-4: Un sistema de importación de las contraseñas almacenadas en un navegador, para su incorporación a los perfiles de contraseñas almacenados en la aplicación.
OBJS-5: Redacción de unos sencillos manuales de usuario y de instalación.
OBJS-6: Creación de un instalador
OBJS-7: Cambio automático de las contraseñas

1.4 Entorno tecnológico

1.4.1 Aplicaciones similares

Este proyecto cubre necesidades de primer orden en nuestro actual modelo de vida rodeados de tecnología y servicios online por lo que existen multitud de aplicaciones que cubren estas necesidades.

Podemos encontrar principalmente dos corrientes para solventar estas necesidades, una es la eliminación de las contraseñas mediante diversos procesos de verificación, habitualmente vinculados al teléfono móvil, y la otra, la cual es la aproximación que tomaremos con la aplicación que vamos a desarrollar, es la centralización de todas las contraseñas en una aplicación o servicio online, dejando en la medida de lo posible que sea la aplicación de manera automática la encargada de introducir dichas contraseñas cuando sea preciso.

Ya que para nosotros no es relevante la primera corriente, a continuación estudiaremos cuatro aplicación que utilizan la segunda aproximación:

- ► Lastpass
  - ▷ Algoritmo: Utiliza como algoritmo de codificación AES de 256 bits
  - ▷ Usuario: Verifica la identidad mediante un usuario vinculado a un correo electrónico utilizando la contraseña maestra como comprobante.
  - ▷ Almacenamiento: El almacenamiento de las contraseñas y usuarios se realiza en la nube.
  - ▷ Generador de contraseñas: Proporciona un generador de contraseñas personalizable, destacando la función de generar contraseñas “pronunciables”
  - ▷ Instalación: Para su funcionamiento es necesaria la instalación de una extensión de navegador para su uso en PCs y de una aplicación en el caso de móviles.
  - ▷ Expiración de contraseñas: No se aprecia una funcionalidad que haga referencia a este dato.

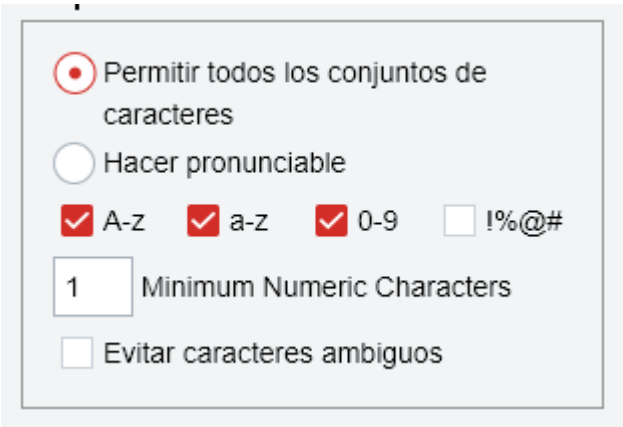


Figura 1:Generador de contraseñas en Lastpass

► Keepass:

▷ Algoritmo: Utiliza como algoritmo de codificación AES de 256 bits o Chacha 20*de 256 bits, a elección del usuario.

▷ Usuario: Tiene una cuenta monousuario de carácter local sin ningún distintivo, utiliza la contraseña maestra cómo verificación.

▷ Almacenamiento: El almacenamiento de las contraseñas y usuarios se realiza de manera local en un archivo.

▷ Generador de contraseñas: Proporciona un generador de contraseñas personalizable, con una entrada que permite introducir una selección de símbolos elegibles para formar parte de la contraseña y con la opción de utilizar algoritmos proporcionados por los usuarios.

▷ Instalación: La aplicación posee un instalador convencional que realiza una instalación en el PC, y además dispone de una versión portable.

▷ Expiración de contraseñas: Permite establecer una fecha de expiración y lo indica en el resumen global, no notifica al usuario de manera directa.

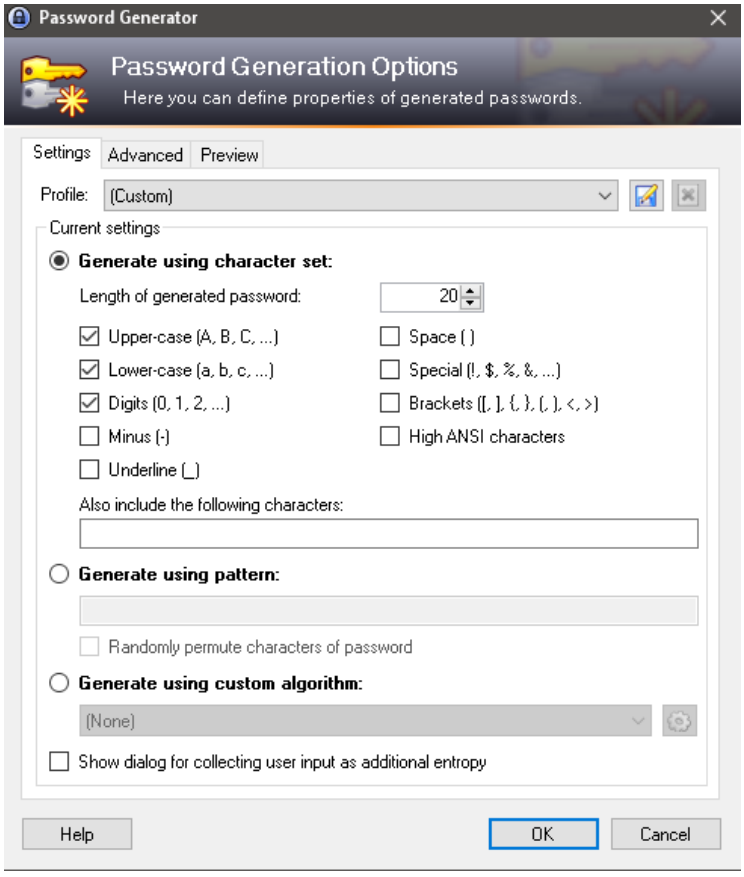


Figura 2:Generador de contraseñas en Keepass

►Sticky Password:

▷ Algoritmo: Utiliza como algoritmo de codificación AES de 256 bits

▷ Usuario: Utiliza una usuario web que se autentifica con un email y la contraseña maestra.

▷ Almacenamiento: El almacenamiento de las contraseñas y usuarios se realiza en la nube con la posibilidad de guardar una copia en local.

▷ Generador de contraseñas: Proporciona un generador de contraseñas personalizable, con una entrada que permite introducir una selección de símbolos elegibles para formar parte de la contraseña.

▷ Instalación: La aplicación posee un instalador convencional que realiza una instalación en el PC aunque es necesario también instalar una extensión en el navegador. También dispone de una versión para móviles mediante una aplicación.

▷ Expiración de contraseñas: No se aprecia una funcionalidad que haga referencia a este dato.

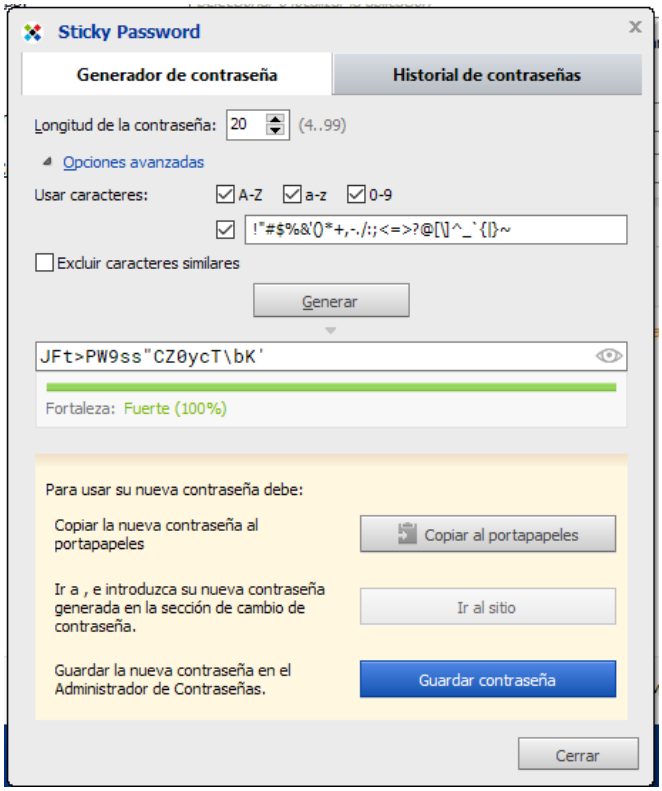


Figura 3:Generador de contraseñas en Sticky Password

►True key:

▷ Algoritmo: Utiliza como algoritmo de codificación AES de 256 bits

▷ Usuario: Utiliza una usuario web que se autentifica con un email y la contraseña maestra.

▷ Almacenamiento: El almacenamiento de las contraseñas y usuarios se realiza en la nube.

▷ Generador de contraseñas: Proporciona un generador de contraseñas personalizable.

▷ Instalación: La aplicación funciona mediante una extensión en el navegador. También dispone de una versión para móviles mediante una aplicación.

▷ Expiración de contraseñas: No se aprecia una funcionalidad que haga referencia a este dato.

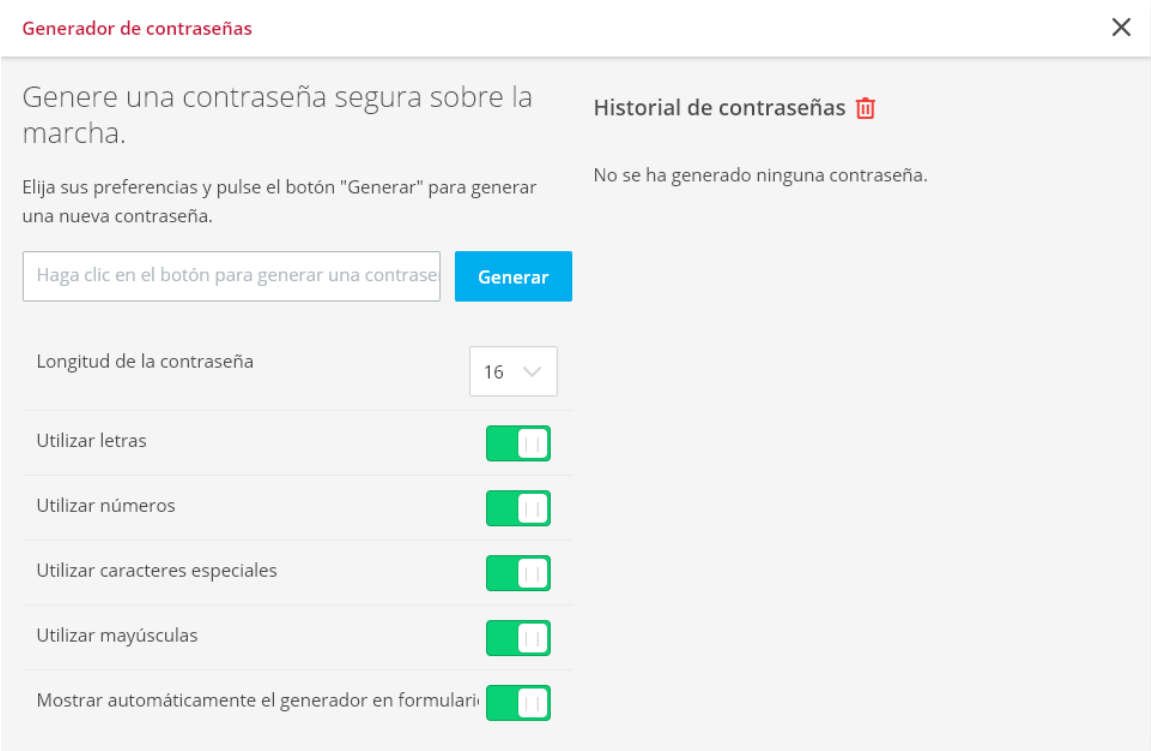


Figura 4:Generador de contraseñas en True Key

A continuación mostraremos una tabla comparativa, incluyendo las características planeadas para nuestra aplicación.

		Lastpass	Keepass	Sticky Password	True Key	SPM
Algoritmo	AES 256	✓	✓	✓	✓	✓
	Chacha20 246		✓			
Usuario	Contraseña maestra	✓	✓	✓	✓	✓
	Username	✓		✓	✓	
Almacenamiento	Nube	✓		✓	✓	
	Local		✓	✓		✓

Generador de contraseñas	Personalización básica	✓	✓	✓	✓	✓
	Inclusión de caracteres		✓	✓	✓	✓
	Algoritmos externos		✓			
Instalación	Local		✓	✓		✓
	Extensión Navegador	✓		✓	✓	
	APP Móvil	✓			✓	
Expiración de Contraseñas	Expiración		✓			✓
	Notificación a usuario					✓

Tabla 1:Comparativa de gestores de contraseñas

1.4.2 Lenguaje

Para el desarrollo de la aplicación se ha decidido utilizar el lenguaje de programación orientado a objetos Java*. Esta decisión parte de dos premisas importantes, la primera es la gran portabilidad de las aplicaciones programadas en este lenguaje, la cual permite que la aplicación desarrollada pueda ser utilizada en una amplia cantidad de equipos. La segunda atiende a la gran familiaridad adquirida a lo largo de la formación que constituye el título con el desarrollo de aplicaciones en Java, lo que disminuye los tiempos de desarrollo reduciendo la necesidad de realizar un aprendizaje extra y facilita que la planificación y estimaciones sean más cercanas a la realidad.

1.4.3 Algoritmos de codificación

Para el desarrollo de la aplicación son importantes tres algoritmos que mencionaremos a continuación.

El primero de ellos es AES (Advanced Encryption Standard), concretamente en su versión de 256 bits. Este algoritmo nos permite codificar bloques de hasta 256 bits y será el algoritmo encargado de codificar las contraseñas que guardara nuestra aplicación, no se ha encontrado una manera eficaz de romper este algoritmo y un ataque de fuerza bruta* es algo computacionalmente inviable en un tiempo razonable.

Los dos restantes son dos variantes de el algoritmo de hash SHA* (Secure Hash Algorithm). Es importante entender, que mientras un algoritmo de codificación nos permitirá posteriormente recuperar lo codificado, no sucede así con los algoritmos de hashing, con los

que a partir del mensaje codificado no se puede recuperar el original. Uno de ellos es la variante SHA-1, esta variante será la encargada de enmascarar las contraseñas antiguas que guardara nuestra aplicación para evitar la reutilización de estas, para ello cotejaremos el hash de una nueva contraseña a utilizar, y solo la usaremos si ese hash no está almacenado como una contraseña antigua. El segundo algoritmo de este tipo, y el último relevante para esta aplicación es el algoritmo SHA-384, mientras que el algoritmo SHA-1 producia una salida de 128 bits, SHA-384 produce una salida de 384 bits. Este algoritmo cumplira una doble función, utilizando la contraseña maestra como entrada, de la salida obtenida tomaremos los 128 bits como medio para guardar una referencia a esta en el sistema para verificar que es la correcta, y los 256 bits restantes serán utilizados como clave para el algoritmo AES de 256 bits que como ya hemos mencionado será el encargado de codificar nuestras contraseñas y demás datos sensibles incluidos los 128 bits guardados como comprobante.

1.5 Organización del documento

En este anexo se explica la estructura de este documento. A continuación podemos ver las diferentes partes y capítulos que componen este documento:

►Parte I: Memoria del proyecto

En esta parte se encuentran los apartados relacionados con el acometimiento del proyecto desde su concepción hasta el seguimiento de este.

▷ Capítulo 1: Introducción

Este capítulo sirve de presentación del proyecto mostrando la motivación y objetivos de este proyecto, así como el entorno tecnológico que lo engloba. Es el capitulo en el que nos encontramos.

▷ Capítulo 2: Planificación, estimaciones y presupuesto

En este capítulo se muestra la preparación previa al proyecto con estimaciones, la planificación y el presupuesto. También se explica la metodología utiliza para el desarrollo de la aplicación.

▷ Capítulo 3: Seguimiento

En esta capítulo se enfrenta la planificación con la realidad, así como explicar las medidas correctivas que se han tomado debido a la diferencia entre ambas.

►Parte II: Documentación técnica

En esta parte se detallan el análisis (identificación y especificación de requisitos), diseño (arquitecturas lógica y física y diagrama de clases), implementación (detalles relevantes de la implementación realizada) y pruebas (pruebas de caja blanca y de caja negra) de cada una de las respectivas iteraciones realizadas.

▷ Capítulo 4: Primera iteración

Este capítulo está dedicado al análisis, el diseño, detalles de implementación y pruebas de la primera iteración.

▷ Capítulo 5: Segunda iteración

Este capítulo está dedicado al análisis, el diseño, detalles de implementación y pruebas de la segunda iteración.

▷ Capítulo 6: Tercera iteración

Este capítulo está dedicado al análisis, el diseño, detalles de implementación y pruebas de la tercera iteración.

►Parte III: Manuales

Esta parte contiene el capítulo 7 dedicado a los manuales.

▷ Capítulo 7: Manuales

Este capítulo está dedicado al manual de instalación que explica como poner a funcionar la aplicación, y el manual de usuario que explica la funcionalidad que tiene la aplicación y cómo hacer uso de ella.

►Parte IV: Conclusiones

Esta parte está el último capítulo, el capítulo 8 conclusiones.

▷ Capítulo 8: Conclusiones

En esta parte se exponen las conclusiones finales tras la realización de este proyecto.

►Parte V: Anexos

En esta parte se añaden elementos que complementan a esta memoria, como un pequeño glosario, un esquema del contenido del CD adjunto a este trabajo y elementos que formaron parte de la memoria pero han sido retirados por diversos motivos.

►Parte VI: Webgrafía

En esta parte se muestran las fuentes externas utilizadas para la realización de este proyecto, indicando el motivo y fecha de su visita.

Capítulo 2

Planificación, estimaciones y presupuesto

2.1 Metodología

Para la realización de este TFG se seguirá una metodología incremental donde a partir de pequeños prototipos se vaya añadiendo funcionalidad y la mejora de la interfaz.

Se planearán con antelación las funcionales a desarrollar en cada iteración así como el número de estas. Al acabar una iteración se procederá a realizar una reunión de validación con el tutor así como una comparación con la línea base del proyecto tras la cual se tomarán medidas correctivas en la planificación de ser necesario.

2.2 Estimaciones

El desarrollo de una aplicación de estas características es un proyecto que podría extenderse infinitamente en el tiempo, mientras se implementan mejoras o se aumentan las opciones en cuanto a los cifrados usados, pero ya que el desarrollo se encuentra en el contexto de la realización de un trabajo de fin de grado que reconoce 12 créditos (300 horas), procederemos mediante esta estimación a delimitar los objetivos a cumplir de tal manera que no se exceda de manera significativa las 300 horas.

Para la estimación de la duración de las tareas de este proyecto utilizaremos el método de Albretch. Este método evalúa la cantidad de elementos pertenecientes a las categorías de: entradas, salidas, consultas, ficheros internos, ficheros externos, y asigna a cada uno de los elementos identificados un coste en puntos de función según su complejidad (baja, media o alta) siguiendo la siguiente tabla:

	Baja	Media	Alta
Entradas	3	4	6
Salidas	4	5	7
Consultas	3	4	6
F.Internos	7	10	15
F.Externos	5	7	10

Tabla 2:Costes de los elementos en el Método de Albretch

Una vez obtenidos los costes en puntos de función se aplicará un factor de ajuste de complejidad en función de la siguiente fórmula: $FA = (0.01\,x\,\Sigma FC) + 0.65$,siendo FA dicho factor de ajuste y FC el sumatorio de las evaluaciones comprendidas entre el 0 y el 5 de las características presentes en la siguiente tabla:

Respaldo y recuperación	Actualización en Línea
Comunicaciones de Datos	Facilidad de operación
Procesamiento Distribuido	Complejo de Procesamiento Interno
Objetivos de Rendimiento	Código diseñado para la reutilización
Gran uso de la configuración	Conversión/ Instalación en Diseño
Entrada de Datos en Línea	Instalaciones Múltiples
Eficiencia con el usuario final	Aplicación diseñada para Cambio

Tabla 3:Factores de complejidad en el Método de Albretch

Una vez obtenidos los puntos de función ajustados se procede a traducirlos a horas de trabajo a partir de un factor obtenido con la experiencia, ya que nuestra experiencia es mínima utilizaremos un factor de comienzo recomendado por los estándares de 1 PF = 1 día de trabajo, considerando una jornada laboral de 8 horas, es decir 1 PF = 8 horas de trabajo.

Antes de proceder con la estimación, procederemos a enumerar las divisiones en bloques de objetivos que definen cada una de las iteraciones planeadas, ordenadas de mayor a menor prioridad, de tal manera que la estimaciones realizadas posteriormente definirán si cuántas de esas iteraciones se llevarán a cabo. Las iteraciones planeadas son las 5 siguientes:

►Primera iteración, Funcionalidad principal:
OBJ-1: Identificación del usuario mediante una contraseña maestra.
OBJ-2: Almacenamiento de contraseñas en local bajo el algoritmo de codificación AES y la clave maestra.
OBJ-3: Permitir crear contraseñas seguras mediante un algoritmo personalizable.
OBJ-4: Recordar al usuario de manera periódica el cambio de las contraseñas almacenadas (notificación)
OBJ-5: Recordar contraseñas anteriores para evitar el uso de estas en el futuro.
OBJS-4: Redacción de unos sencillos manuales de usuario y de instalación.

►Segunda iteración, Interfaz Gráfica:
OBJ-6: Facilitar la inclusión de perfiles de usuario base para guardar contraseñas de aplicaciones no soportadas por la aplicación.
OBJS-1: Diseño e implementación de una interfaz gráfica que permita un uso fluido de la aplicación.

►Tercera iteración, Transferencia de contraseñas:
OBJS-2: Un sistema de exportación de las contraseñas almacenadas, para su posterior importación desde un navegador (un único navegador).

OBJS-3: Un sistema de exportación de las contraseñas almacenadas, para su posterior importación desde otro equipo que tenga instalada la aplicación.

►Cuarta iteración, Mejoras y ampliaciones:
OBJS-6: Creación de un instalador.
OBJ-4: Recordar al usuario de manera periódica el cambio de las contraseñas almacenadas (email).
OBJS-2: Un sistema de exportación de las contraseñas almacenadas, para su posterior importación desde un navegador (al menos un navegador más).

►Quinta iteración, Cambio automático:
OBJS-7: Cambio automático de las contraseñas.

Una vez definidas las posibles iteraciones comenzamos realizando la table de ajuste que compartirán todas ellas, siendo esta la tabla que se muestra a continuación:

Factor	FC
Respaldo y recuperación	5
Comunicaciones de Datos	0
Procesamiento Distribuido	0
Objetivos de Rendimiento	1
Gran uso de la configuración	1
Entrada de Datos en Línea	0
Eficiencia con el usuario final	2
Actualización en Línea	0
Facilidad de operación	5
Complejo de Procesamiento Interno	3
Código diseñado para la reutilización	3
Conversión/ Instalación en Diseño	2
Instalaciones Múltiples	3
Aplicación diseñada para Cambio	3
Total:	28

Tabla 4:Factores de complejidad en el Método de Albretch evaluados

Utilizando la fórmula ya mencionada obtenemos el siguiente factor de ajuste:

$$FA = (0.01\ x\ 28) + 0.65 \quad \Rightarrow FA = 0.93$$

Una vez calculada la parte común comenzaremos con la estimación de la primera iteración. Tras analizar las necesidades a cubrir para cumplir los objetivos de esta iteración las características encontradas son las siguientes:

Entradas:

-Contraseña maestra

-Contraseña(incluidos datos de acceso)

Salidas:

-Contraseña generada

-Recordatorio cambio

Fichero interno:

-Contraseñas actuales

-Contraseñas pasadas

Trás asignarles la complejidad adecuada obtenemos la siguiente tabla:

Entradas	Baja (3)	2	6
	Media (4)	0	0
	Alta (6)	0	0
Salidas	Baja (4)	2	8
	Media (5)	0	0
	Alta (7)	0	0
Consultas	Baja (3)	0	0
	Media (4)	0	0
	Alta (6)	0	0
F Externos	Baja (7)	0	0
	Media (10)	0	0
	Alta (15)	0	0
F Internos	Baja (5)	2	10
	Media (7)	0	0
	Alta (10)	0	0
Total			24

Tabla 5:Equivalencia de características a PF de la primera iteración

Ajustando los puntos de función obtenemos:

$$24\ x\ 0.93\ =\ 22.32 => 22.32\ x\ 8\ = 178.56\,\text{h}$$

Con lo cual la duración estimada de la primera iteración es de 178.56 horas. A partir de esas horas como subtotal procedemos con la segunda iteración, siendo sus características las siguientes:

Entradas

- Nuevo perfil de aplicación

Consultas

- Contraseña guardada
- Generar contraseña

Ficheros internos

- Perfiles de aplicación

Ajustadas por su complejidad y tipo obtenemos la siguiente tabla:

Entradas	Baja (3)	1	3
	Media (4)	0	0
	Alta (6)	0	0
Salidas	Baja (4)	0	0
	Media (5)	0	0
	Alta (7)	0	0
Consultas	Baja (3)	2	6
	Media (4)	0	0
	Alta (6)	0	0
F Externos	Baja (7)	0	0
	Media (10)	0	0
	Alta (15)	0	0
F Internos	Baja (5)	1	5
	Media (7)	0	0
	Alta (10)	0	0
Total			14

Tabla 6:Equivalencia de características a PF de la segunda iteración

Ajustando los puntos de función obtenemos:

$$14\ x\ 0.93\ =\ 13.02 \Rightarrow\ 13.02\ x\ 8\ =\ 104.16\,\text{h}$$

La duración de la segunda iteración es de 104.16 horas, y el subtotal acumulado hasta el momento es de 282.72 horas. Procedamos con las características de la tercera iteración:

Salidas

-Archivo sincronización Chrome

-Archivo sincronización

Ficheros Externos

-Archivo sincronización Chrome

Con la consiguiente tabla una vez ajustada la complejidad:

Entradas	Baja (3)	0	0
	Media (4)	0	0
	Alta (6)	0	0
Salidas	Baja (4)	2	8
	Media (5)	0	0
	Alta (7)	0	0
Consultas	Baja (3)	0	0
	Media (4)	0	0
	Alta (6)	0	0
F Externos	Baja (7)	1	7
	Media (10)	0	0
	Alta (15)	0	0
F Internos	Baja (5)	0	0
	Media (7)	0	0
	Alta (10)	0	0
Total			15

Tabla 7:Equivalencia de características a PF de la tercera iteración

Ajustando los puntos de función obtenemos:

$$15\ x\ 0.93\ =\ 13.95 => 13.95\ x\ 8\ = 111.6\,h$$

La duración estimada de la tercera iteración sería de 111.6, con un total acumulado de 394.32 horas. Ya que este total excede ampliamente las 300 horas a reconocer con este proyecto, realizaremos una estimación de nuevo manteniendo únicamente la característica más importante de esta iteración, siendo las características de esta nueva iteración reducida las siguientes:

Salidas

-Archivo sincronización

Con la correspondiente tabla:

Entradas	Baja (3)	0	0
	Media (4)	0	0
	Alta (6)	0	0
Salidas	Baja (4)	1	4
	Media (5)	0	0
	Alta (7)	0	0
Consultas	Baja (3)	0	0
	Media (4)	0	0
	Alta (6)	0	0
F Externos	Baja (7)	0	0
	Media (10)	0	0
	Alta (15)	0	0
F Internos	Baja (5)	0	0
	Media (7)	0	0
	Alta (10)	0	0
Total			4

Tabla 8:Equivalencia de características a PF de la tercera iteración reducida

Ajustando los puntos de función obtenemos:

$$4\ x\ 0.93\ =\ 3.72 => 3.72\ x\ 8\ = 29.76\,h$$

La duración estimada de esta nueva tercera iteración reducida sería de 29.76 horas, con un total de 312.48 horas para el proyecto completo, aún por encima de las 300 horas pero por un margen asumible.

Dado que ya se han superado las 300 horas de trabajo no se realizarán una cuarta y quinta iteración por lo que no será necesario estimar su duración.

2.3 Planificación

En el transcurso del desarrollo de esta aplicación podemos definir dos situaciones bien diferenciadas que determinan el número de horas disponibles para desarrollar la aplicación. Siendo la primera el periodo comprendido entre el comienzo del proyecto y el 17 de Marzo, en la cual se dispondrá de tiempo completo a excepción de los Martes y Miércoles de 18:00 a 20:00, y apartir del 18 de Marzo cuando por comienzo de las prácticas el tiempo de desarrollo disponible se reducirá drásticamente.

El trabajo se realizará bajo las tres jornadas laborales distintas, más un pequeña modificación de una de ellas, siendo estas las descritas en la siguiente tabla:

Diaria	Diaria con clases	Fin de Semana	Fin de Semana Red
12:00-14:00	11:00-14:30	10:00-12:00	10:00-12:00
16:00-18:00	15:45-17:15	12:30-14:30	12:30-14:30
19:00-21:00		16:30-18:30	16:30-18:30
		19:15-21:15	

Tabla 9:Jornadas de trabajo

Siguiendo el siguiente calendario de trabajo:

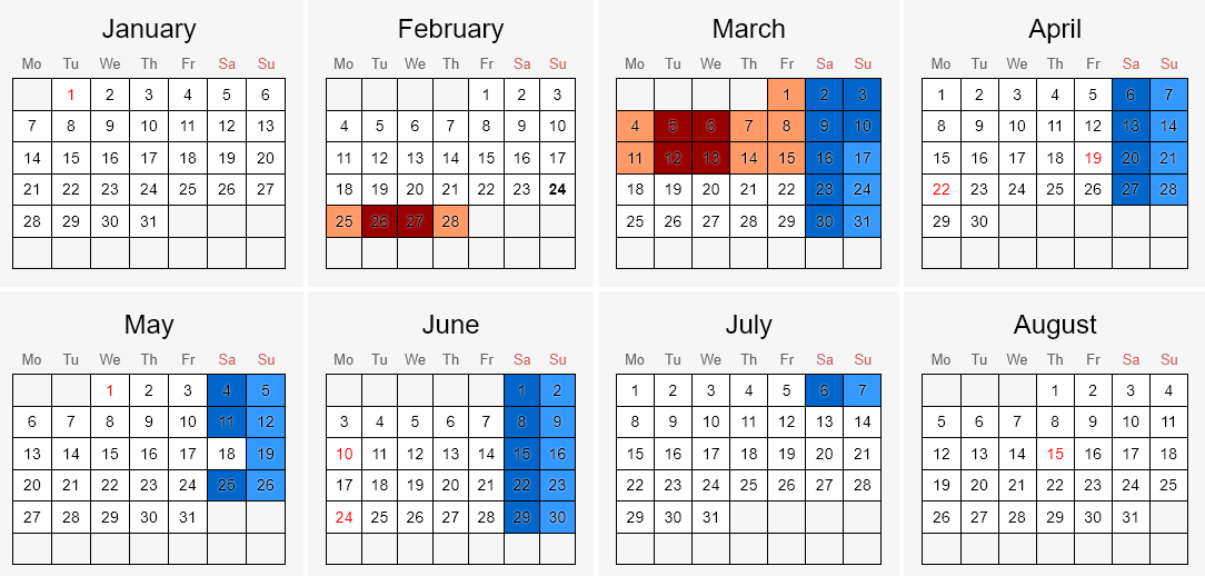


Figura 5:Calendario de trabajo

Para definir qué tareas se realizarán en cada momento primero separaremos las horas de trabajo estimadas para cada iteración siguiendo los siguientes porcentajes:

Análisis	15%
Diseño	25%
Codificación	45%
Pruebas	15%

De tal manera que nos encontramos que para la primera iteración:

Análisis	26.8(15%)
Diseño	44.6(25%)
Codificación	80.35 (45%)
Pruebas	26.81 (15%)

Para la segunda:

Análisis	15.62(15%)
Diseño	26.04(25%)
Codificación	46.8(45%)
Pruebas	15.7(15%)

Y para la tercera:

Análisis	4.46(15%)
Diseño	7.44(25%)
Codificación	13.39(45%)
Pruebas	4.47(15%)

Con esos tiempos por tarea y añadiendo las tareas previas de planificación, obtenemos el siguiente diagrama de gantt:

Figura 6:Diagrama de Gantt de planificación

Esta planificación está sujeta a posibles reajustes una vez evaluada la situación al alcanzar los hitos que representan la finalización de las distintas iteraciones, siendo estos hitos: el 13 de Abril para la primera iteración, el 9 de Junio para la segunda y el 29 de Junio para la tercera, siendo además la fecha estimada de finalización del proyecto.

2.4 Presupuesto

A continuación detallaremos el presupuesto para la realización de este proyecto en base a la planificación y estimaciones realizadas anteriormente, que servirán como base para establecer el porcentaje de uso de los diferentes recursos utilizados.

2.4.1 Hardware

Para la realización de este proyecto se va a utilizar un PC cuyo coste incluidos periféricos asciende a 1 073.30 €. Considerando una vida útil de 5 años, y un uso de 8 horas diarias, 21 días al mes, supone un total de 10.080 horas de vida útil. Como hemos estimado, la duración del proyecto será de 312.48 horas, un 3.1 % de la vida útil de dicho PC por lo tanto se le puede atribuir al proyecto ese porcentaje del precio.

Dado que se contará con los servicios de agentes externos para realizar la impresión de los documentos necesarios no se utilizarán más elementos hardware que el PC anteriormente mencionado.

Elemento	Coste	% de uso/unidades	Coste para el proyecto
PC	1 073.30 €	3.1%	33.27 €
Total			33.27 €

Tabla 10:Presupuesto Hardware

2.4.2 Software

Para el desarrollo de este proyecto se utilizara software tanto libre como de pago a continuación presentaremos una lista de ambas categorías:

- ►Libre:
  - ▷NetBeans: Entorno de desarrollo
  - ▷Google Docs: Herramienta para la edición distribuida de documentos
  - ▷OpenProject: Herramienta para realizar y seguir la planificación del proyecto
  - ▷Draw.io: Herramienta para realizar diagramas
  - ▷Pencil: Herramienta para el diseño de interfaces
  - ▷XIcon - Editor: Herramienta online para el diseño de iconos
  - ▷calendar.proinf.net: Herramienta online para gestionar calendarios
- ►Privativo
  - ▷Windows 10: Sistema operativo (Considerando una vida útil de 5 años, y un uso de 8 horas diarias, 21 días al mes, supone un total de 10.080 horas de vida útil.

Como hemos estimado, la duración del proyecto será de 312.48 horas, un 3.1 % de la vida útil de dicho SO por lo tanto se le puede atribuir al proyecto ese porcentaje del precio)

A continuación tendremos en cuenta sólo aquellos elementos software que repercuten en el presupuesto:

Elemento	Coste	% de uso/unidades	Coste para el proyecto
Windows 10	100 €	3.1%	3.1 €
Total			3.1 €

Tabla 11:Presupuesto Software

2.4.3 Personal

Esta aplicación va a ser desarrollada por un única persona, pero está va a realizar trabajos de diferente índole que tienen asociado un sueldo anual distinto. Esta diferenciación distingue entre analistas para las tareas de análisis y diseño, programador para las tareas de codificación y por último tester para las tareas de pruebas.

Obtenidos mediante una aplicación llamada indeed (referencias 02, 03 y 04 de la webgrafía),que calcula sueldos medios en función de las ofertas publicadas y de información procedentes de empresas y usuarios, los sueldos a tener en cuenta para estas tareas son los siguientes:

Puesto	Sueldo anual	Sueldo mensual	Sueldo por Hora
Analista	27 468 €	2 289 €	13.65 €
Programador	19 032 €	1 586 €	9.44 €
Tester	21 722 €	1 810.16 €	10.77€

Tabla 12:Lista de sueldos tipo

Si desglosamos las horas de trabajo por rol de cada iteración:

Iteración	Tarea	Horas	Horas Analista	Horas Programador	Horas Tester
Primera Iteración	Análisis	26.8 h	26.8 h		
	Diseño	44.6 h	44.6 h		
	Codificación	80.35 h		80.35 h	
	Pruebas	26.81 h			26.81 h
Segunda Iteración	Análisis	15.62 h	15.62 h		
	Diseño	26.04 h	26.04 h		
	Codificación	46.8 h		46.8 h	
	Pruebas	15.7 h			15.7 h
Tercera Iteración	Análisis	4.46 h	4.46 h		
	Diseño	7.44 h	7.44 h		
	Codificación	13.39 h		13.39 h	
	Pruebas	4.47 h			4.47 h
Total por tarifa			124.96 h	140.54 h	46.98 h

Tabla 13:Horas de trabajo por Categoría

Con las horas totales por categoría y los sueldo por hora los costes de personal son los siguientes:

Categoría	Horas de trabajo	Sueldo por hora	Sueldo total
Analista	124.96 h	13.65 €	1 705.70 €
Programador	140.54 h	9.44 €	1 326.69 €
Tester	46.98 h	10.77€	505.97 €
Total			3 538.36 €

Tabla 14:Presupuesto de Personal

2.4.5 Varios

En esta categoría consideraremos gastos derivados del mantenimiento en funcionamiento de los equipos así como otros gastos especiales. Los gastos que contemplaremos serán los siguientes:

▷ Impresión de documentos(Consideramos la impresión completa de la documentación a entregar estimando esta en 10000 folios, y añadimos la impresión de documentación complementaria al desarrollo con una estimación de 50 folios)

▷ Viajes para reuniones con Tutor (Se planea una reunión con el tutor por hito, dado que por la situación laboral se requerirá el traslado desde Valladolid en 3 ocasiones se cargará el coste de estos viajes en el presupuesto)

▷ Electricidad (Se considera un gasto medio de 275 W/h incluyendo el equipo y un monitor, tomando como referencia las 319.48 horas del proyecto para el número de horas con el equipo funcionando 01)

▷ Línea ADSL (Se considera su uso durante un 40% de la duración del proyecto, y con un empleo del 35% del ancho de banda, considerando como tiempo total mensual 8 horas durante 21 días)

Elemento	Coste	% de uso/unidades	Coste para el proyecto
Impresión de documentos	0.15 €/folio	1050 folios	157.50 €
Viajes para reuniones con Tutor	25 €/viaje	3 viajes	75 €
Electricidad	0.1255€/kWh	87.85 kWh	11.02 €
Línea ADSL	50 €/mes (168h)	44.72 h	13.31 €
Total			256.83 €

Tabla 15:Presupuesto de Gastos Varios

2.4.5 Resumen

Aglutinando las cantidades finales atribuibles a las categorías anteriores obtenemos el resumen del presupuesto total:

Categoría	Coste para el proyecto
Hardware	33.27 €
Software	3.1 €
Personal	3 538.36 €
Varios	256.83 €
Total	3 831.56 €

Tabla 16:Resumen del presupuesto

Capítulo 3

Seguimiento

3.1 Comparativa 1ª iteración

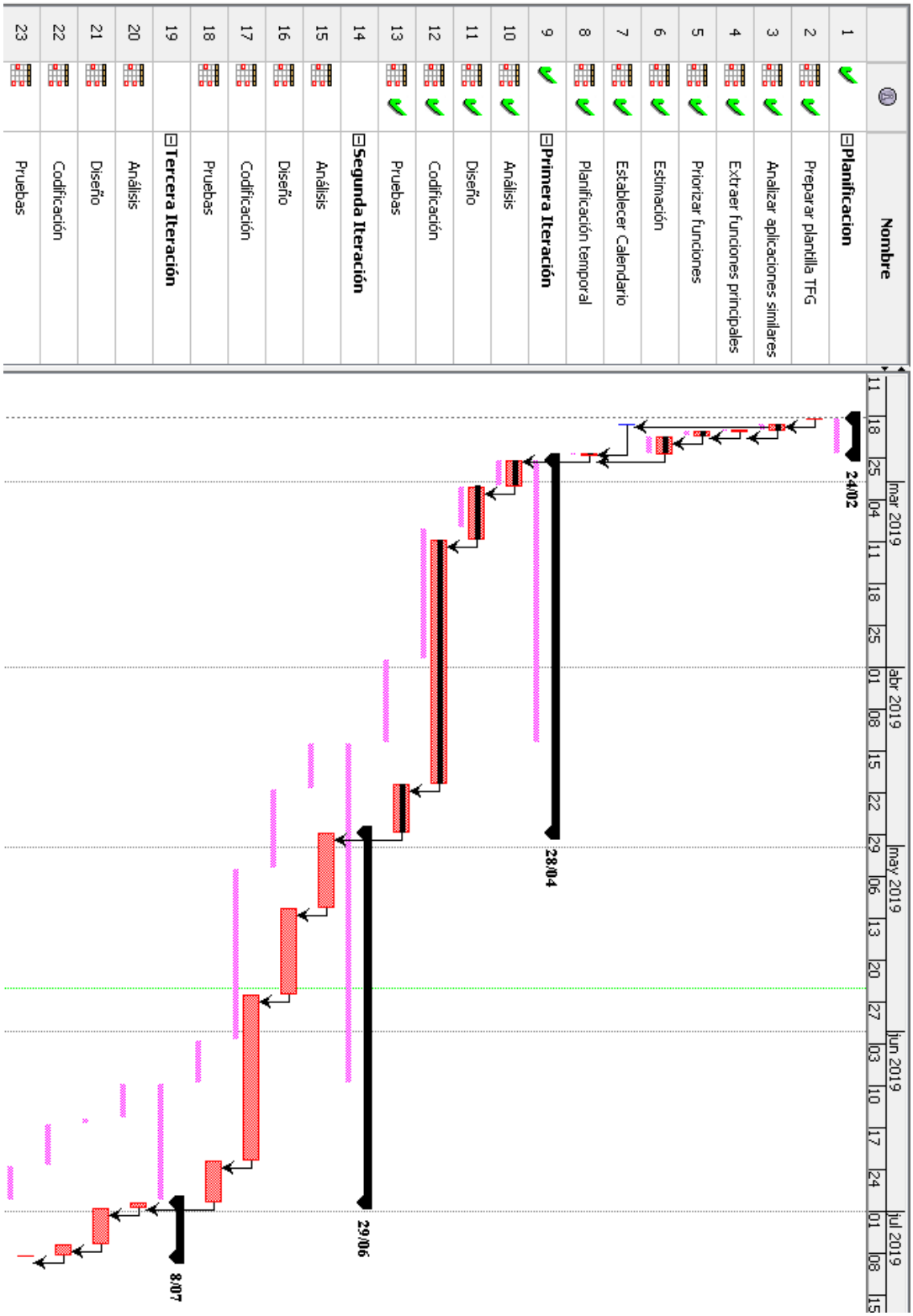


Figura 7:Diagrama de Gantt de seguimiento, fin primera iteración*

En el transcurso de de la primera iteración se ha producido un retraso de 2 semanas fruto de dos elementos no contemplados en la planificación inicial.

El primero de ellos fue un retraso de dos días producido por la necesidad de realizar un trabajo para la asignatura de “Aspectos legales, éticos y profesionales de las IT”, que debido a que en las semanas posteriores a la incorporación en la empresa para la realización de las prácticas curriculares, la jornada de trabajo planificada es de dos días semanales, supone que la pérdida de esos dos días de trabajo repercutan en una semana en el cómputo del proyecto.

Por otro lado nos encontramos con la realización optimista de la previsión del tiempo necesario para el desplazamiento hasta Valladolid los domingos y la preparación de equipaje, que en lugar de producir una reducción de la jornada de 2 horas con respecto a la jornada de los sábados ha supuesto la imposibilidad de trabajar durante la tarde reduciendo la jornada en 4 horas.

En cuanto a la distribución de horas de trabajo efectivas, estas se han ajustado bastante bien a la estimación siendo la más alejada la relativa a las pruebas. Las horas por tarea comparadas con la estimadas son las siguientes:

Análisis	24 (26.8)
Diseño	48 (44.6)
Codificación	78 (80.35)
Pruebas	20 (26.81)
Total	170 (178.56)

Dada la similitud se espera que la estimación el a siguientes iteración se acerque de manera similar a la realidad.

No se han tomado medidas directas durante la iteración para paliar el retraso temporal en el proyecto, pero si se ha realizado un reajuste en la planificación para la segunda iteración. Este reajuste se ha traducido en mover las dos horas perdidas los Domingos a la tarde del Viernes, además de añadir 2 horas más de trabajo para compensar el retraso producido en la iteración anterior.

Estos reajustes se traducen en el nuevo diagrama de Gantt que podemos ver a continuación:

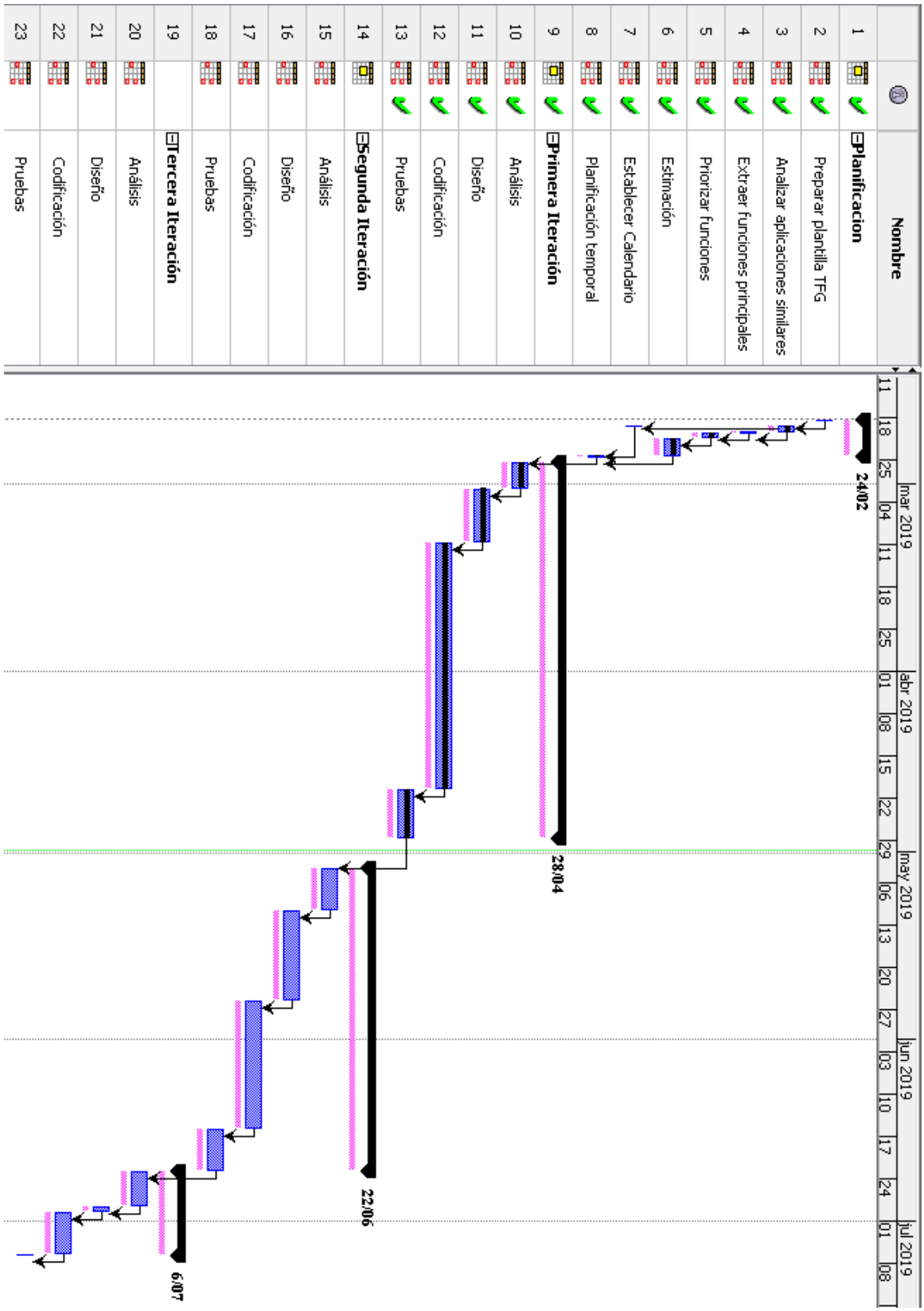


Figura 8:Diagrama de Gantt de planificación, inició segunda iteración

3.2 Comparativa 2ª iteración

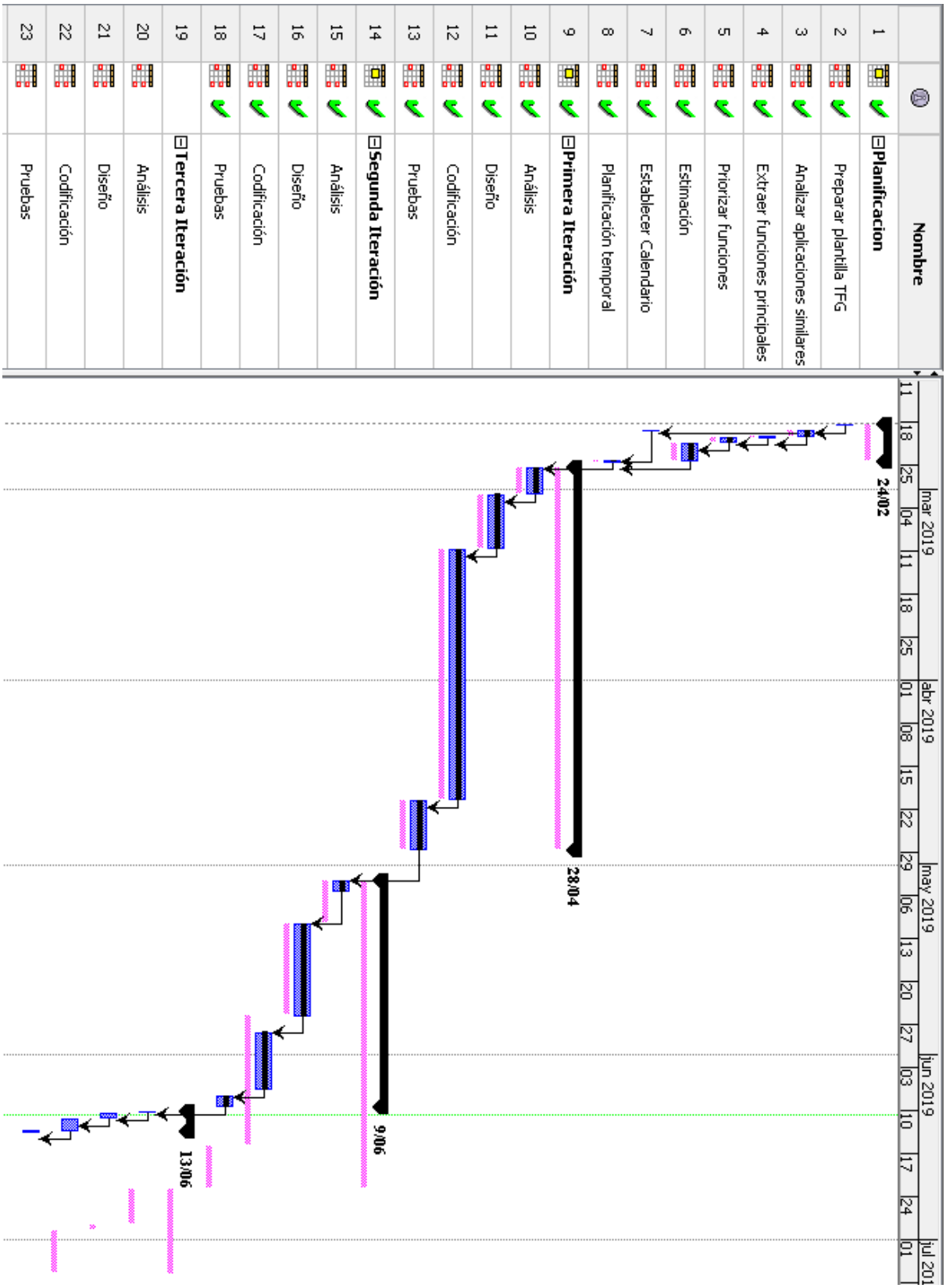


Figura 9:Diagrama de Gantt de planificación, fin segunda iteración

Esta iteración ha estado marcada por dos sucesos importantes. El primero ha sido la aparición de un evento no planificado de la empresa en la que realizó las prácticas de dos semanas del 16 de Junio al 28 que ha supuesto el adelanto de la fecha de entrega al 14 de Junio.

El otro y motivo por el cual esa fecha de entrega es posible ha sido la consecución de el objetivo principal para el periodo de prácticas en la empresa, lo que ha supuesto el conseguir permiso para utilizar horas de trabajo para la realización de este proyecto desde el día 27 de Mayo.

Esto ha supuesto que de lunes a viernes se realicen 6 horas diarias de trabajo efectivo, cambiando el horario de trabajo para los Viernes, pero manteniendo el de Sábados y Domingos

En cuanto a las horas de trabajo realizadas frente a la estimación nos encontramos con la siguiente situación:

Análisis	15 (15.62)
Diseño	30(26.04)
Codificación	68(46.8)
Pruebas	16(15.7)
Total	129(104.16)

Podemos observar como el tiempo de codificación ha sido notablemente superior, mientras que el resto son muy cercanos a la estimación. Esto tiene su explicación en el método de estimación utilizado, ya que es más preciso para elementos lógicos y métodos no recogiendo bien la dificultad de las interfaces, siendo estas el principal elemento de esta iteración.

Dado que la iteración restante es de carácter lógico se mantendrá la estimación, pero se tendrá en cuenta en el futuro este fallo para utilizar un método distinto para aplicaciones o iteraciones con una alta carga en la interfaz de usuario.

Este aumento de las horas, también ha supuesto la necesidad de tomar una decisión sobre la realización o no de la tercera iteración al haberse alcanzado las 300 horas de trabajo (170+129) . Dado la naturaleza de la aplicación y ya que la inclusión de la posibilidad de importar/exportar información facilita la creación de copias de seguridad, se ha decidido proseguir con la tercera iteración de todas formas. No obstante se resolverá la funcionalidad de la manera más sencilla posible.

Ajustando la línea base a las nuevas realidades el diagrama de Gantt esperado para la siguiente iteración es el siguiente:

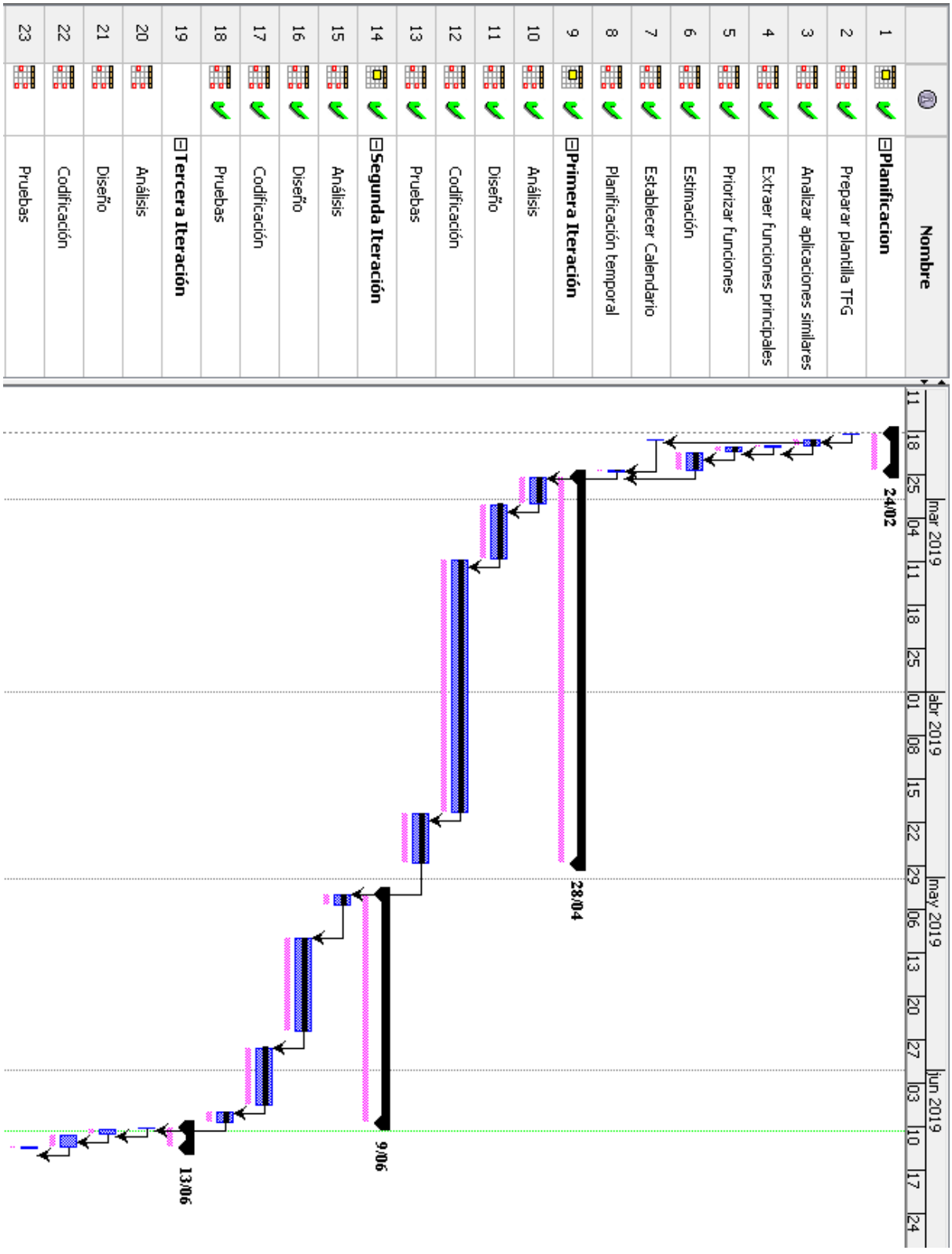


Figura 10:Diagrama de Gantt de planificación, inició tercera iteración

El nuevo fin esperado está dentro de los nuevos plazos.

3.3 Comparativa 3ª iteración

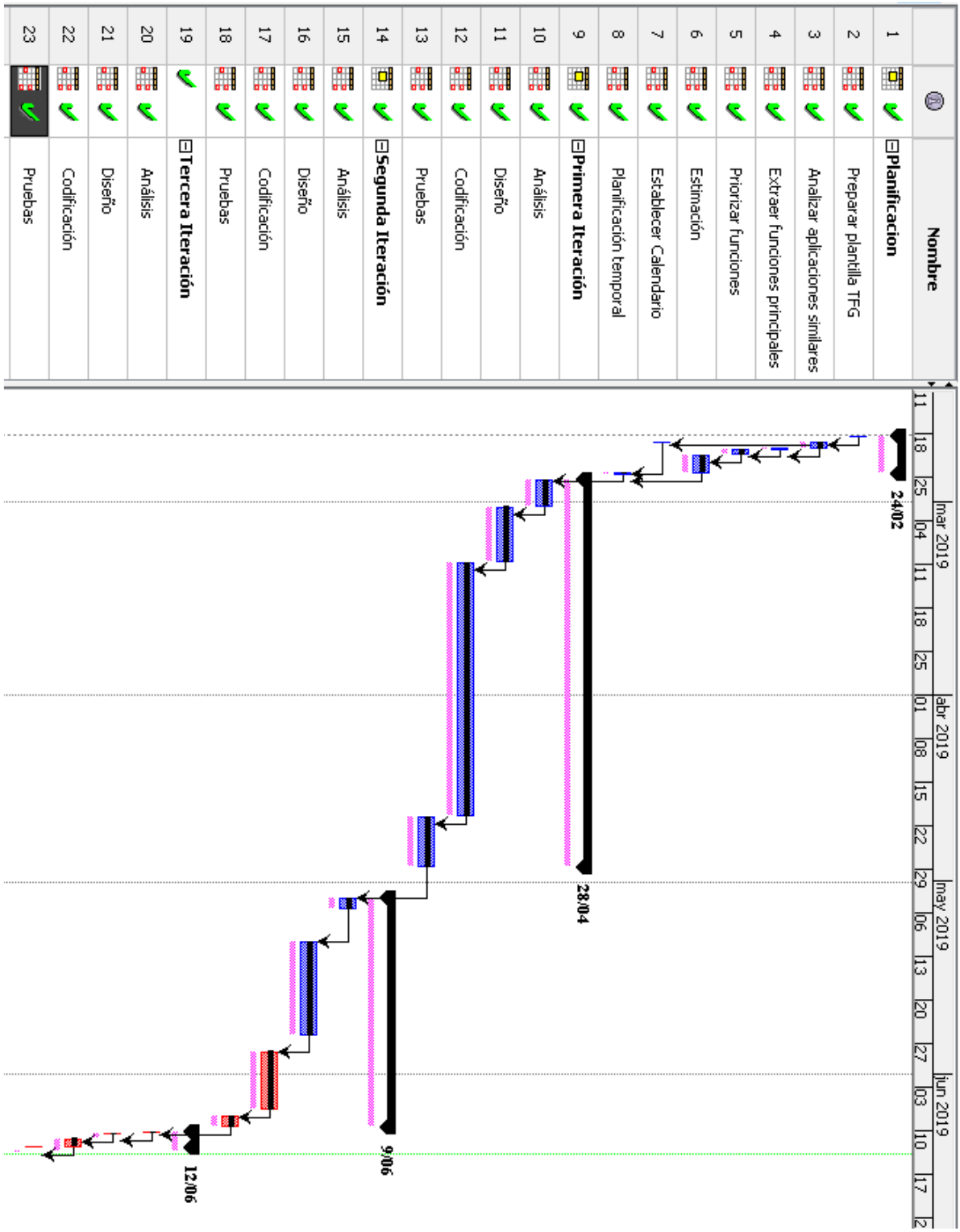


Figura 11:Diagrama de Gantt de planificación, fin de proyecto

Esta última iteración ha sido una iteración muy breve y sin incidentes.

La simplificación de la solución al mínimo que permite resolver los requisitos presentados ha resultado en una reducción del tiempo en todas las áreas más o menos proporcionado como vemos a continuación:

Análisis	2.5(4.46)
Diseño	3.5 (7.44)
Codificación	7.5(13.39)
Pruebas	2.5 (4.47)
Total	16 (29.76)

El final de esta iteración de esta iteración eleva el número de horas de desarrollo totales a 315 frente a las 312,48 que fueron estimadas. Aunque los números son similares y las estimaciones han resultado ser bastante precisas no hay que olvidar que en esta última iteración se ha reducido el trabajo premeditadamente para compensar la subestimación de la segunda iteración, además de por límites temporales debido al contexto temporal del proyecto.

En cuanto a las fechas debido a las circunstancias la fecha de finalización se ha adelantado 26 días a la fecha prevista.

A modo de resumen del proyecto a continuación se muestra la línea temporal real con la line base inicial:

Figura 12:Diagrama de Gantt de planificación, linea base inicial frente fin proyecto

Parte II

Documentación Técnica

Capítulo 4

Primera Iteración

4.1 Análisis

4.1.1 Características

A continuación se muestra el árbol de características* que modela esta iteración:

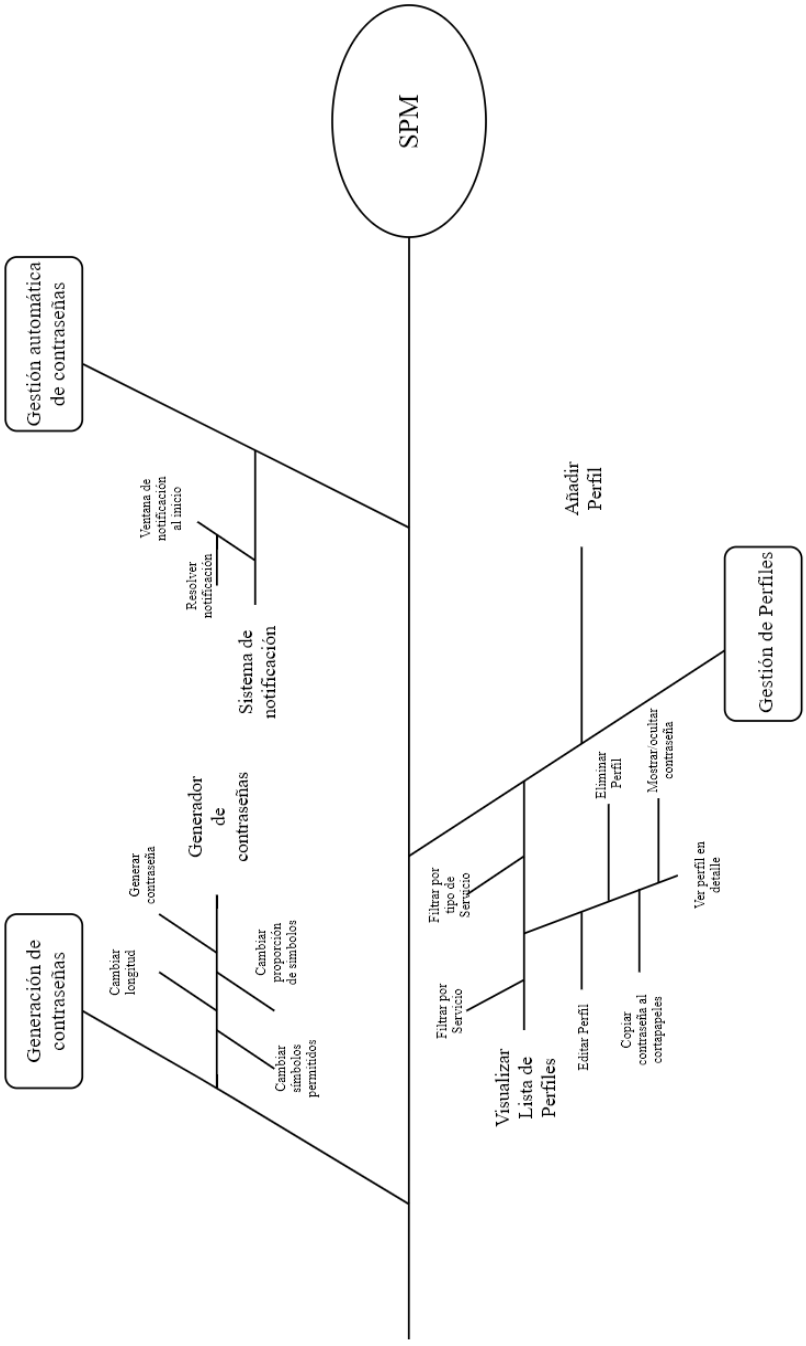


Figura 13:Árbol de características de la primera iteración

4.1.2 Actores

Dado el carácter privado y personal de esta aplicación no se plantea el uso de esta por parte de actores* distintos. Es por tanto el actor "Usuario", entendido este como un usuario que requiere la gestión de sus contraseñas y utilizará para cubrir tal necesidad esta aplicación. Este actor tendrá acceso a toda la funcionalidad de la aplicación una vez introduzca la contraseña maestra, es importante entender que aunque se utiliza ese momento para la verificación de la identidad del usuario el objetivo principal de esta operación es proporcionar a la aplicación la contraseña para descodificar los datos a partir de esa contraseña maestra, motivo por el cual no se tiene en consideración un posible actor "Usuario no identificado".

4.1.3 Requisitos de usuario

Los requisitos de usuario* son los siguientes:

RU-01: Un usuario podrá establecer una contraseña maestra.
RU-02: Un usuario podrá validar su legitimidad introduciendo la contraseña maestra.
RU-03: Un usuario podrá generar una contraseña.
RU-04: Un usuario podrá editar las reglas de generación de contraseñas.
RU-05: Un usuario podrá resolver una notificación recibida.
RU-06: Un usuario podrá visualizar la lista de perfiles
RU-07: Un usuario podrá filtrar la lista de perfiles por servicio.
RU-08: Un usuario podrá filtrar la lista de perfiles por tipo de servicio.
RU-09: Un usuario podrá visualizar un perfil de manera detallada.
RU-10: Un usuario podrá cambiar el estado de la contraseña de oculto a visible y viceversa.
RU.11: Un usuario podrá copiar la contraseña al portapapeles.
RU-12: Un usuario podrá eliminar un perfil.
RU-13: Un usuario podrá modificar un perfil.
RU-14: Un usuario podrá crear un perfil.
RU-15: Un usuario podrá seleccionar el servicio de un perfil.
RU-16: Un usuario podrá seleccionar el tipo de servicio de un perfil.
RU-17: Un usuario podrá volver a un estado previo a la validación de la contraseña maestra.

4.1.4 Diagrama de casos de uso

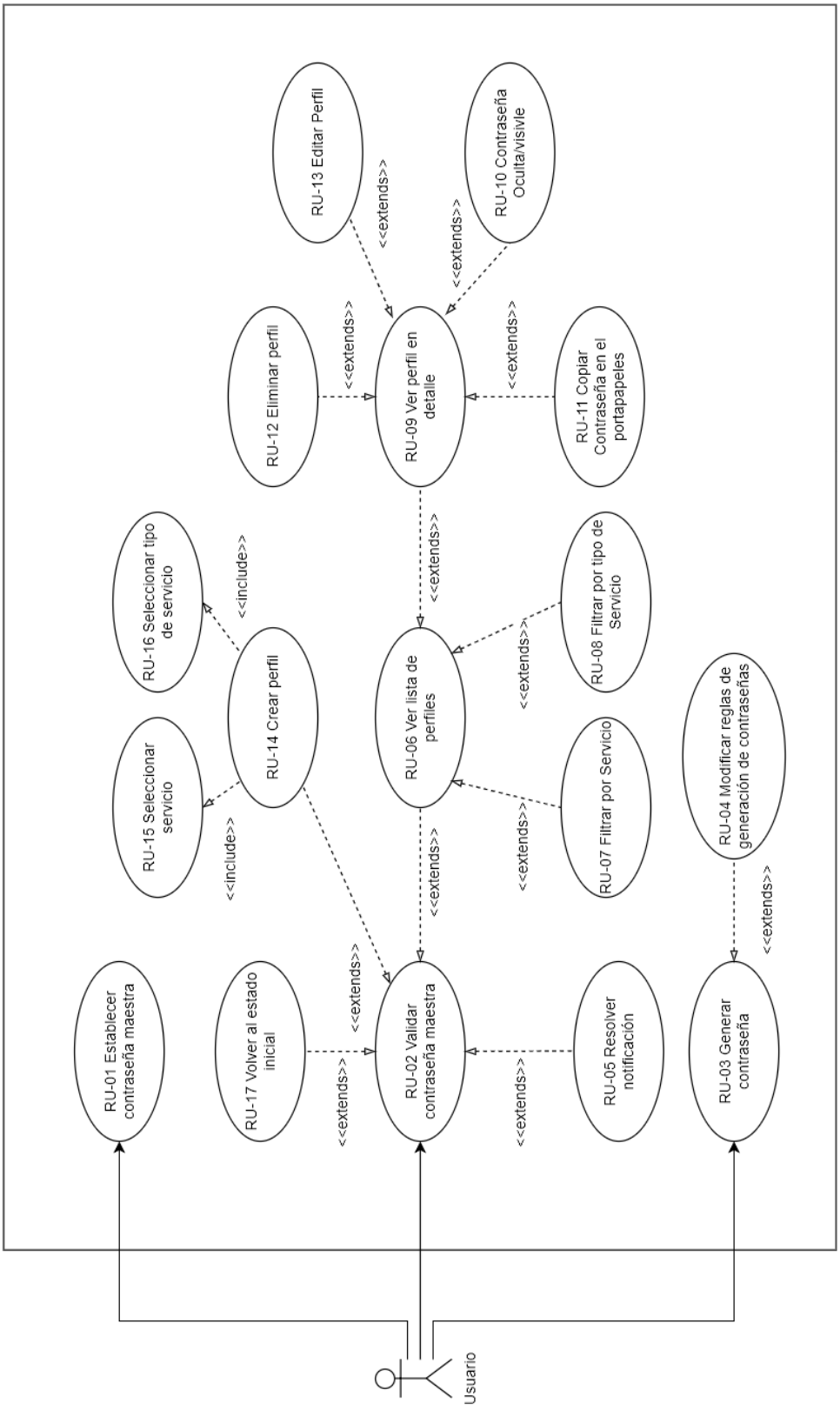


Figura 14:Diagrama de Casos de Uso

4.1.5 Especificación de requisitos de Usuario

US-01	Establecer contraseña Maestra	
Versión	1.1	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-01	
Requisitos asociados	RU-01	
Descripción	El sistema solicita al usuario una contraseña a partir de la cual se generará la clave para codificar la información.	
Precondición	No está establecida una contraseña maestra.	
Secuencia normal	Paso	Acción
	1	El usuario accede a la aplicación por primera vez.
	2	La aplicación notifica que la pérdida de la contraseña maestra supone la pérdida de la información codificada.
	3	La aplicación solicita una contraseña maestra.
	4	El usuario introduce una contraseña.
	5	La aplicación aplica el algoritmo SHA-384 sobre la contraseña introducida.
	6	La aplicación guarda los 128 primeros bits de la salida del algoritmo SHA-384 codificados mediante AES-256 con los 256 restantes, como medio para validar la contraseña maestra en futuras ocasiones.
	7	La aplicación carga en la memoria temporal del programa los 256 últimos bits de la salida de SHA-384 para usarlos como clave en la codificación con AES-256.
	8	La aplicación notifica al usuario que el proceso se ha completado.
	9	La aplicación muestral el menú principal.
	10	El caso de uso finaliza con éxito.
Postcondición	La aplicación está en el menú principal, lista para codificar información.	
Excepciones	Paso	Acción

Comentarios	

Tabla 17:Especificación del requisito US-01

US-02	Validar la contraseña maestra	
Versión	1.1	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-01	
Requisitos asociados	RU-02	
Descripción	Un usuario podrá introducir la contraseña maestra para verificar que es un usuario legítimo.	
Precondición	Se ha establecido una contraseña maestra y el usuario se encuentra en el menú de inicio.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la opción introducir contraseña.
	2	La aplicación solicita la contraseña.
	3	El usuario introduce la contraseña.
	4	La aplicación aplica el algoritmo SHA-384 a la contraseña.
	5	La aplicación codifica con AES-256 y los 256 últimos bits, los 128 primeros bits para validar la contraseña.
	6	La aplicación carga en memoria los 256 últimos bits para usarlos como clave en la codificación con AES-256.
	7	La aplicación muestral el menú principal.
	8	El caso de uso finaliza con éxito.
Postcondición	La aplicación está en el menú principal, lista para codificar información.	
Excepciones	Paso	Acción
	6 b	La aplicación notifica al usuario de que la contraseña es incorrecta y se vuelve al paso 2.
Comentarios		

Tabla 18:Especificación del requisito US-02

US-03	Generar contraseña	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-03	
Requisitos asociados	RU-03, RU-04	
Descripción	La aplicación permitirá la generación de contraseñas a partir de opciones personalizables.	
Precondición	El usuario ha accedido a la generación de contraseñas desde el menú de inicio, desde el menú principal, desde la creación de un perfil, o desde la edición de un perfil.	
Secuencia normal	Paso	Acción
	1	La aplicación muestra las opciones por defecto para generar contraseñas y un botón para generar una contraseña.
	2	El usuario pulsa en generar una contraseña.
	3	El sistema genera una contraseña con las opciones seleccionadas en ese momento.
	4	El sistema muestra la contraseña generada en un cuadro de texto seleccionable.
	5	El sistema muestra la opción de copiar la contraseña al portapapeles.
	6	El usuario pulsa la opción copiar contraseña al portapapeles.
	7	El caso de uso a finalizado con éxito.
Postcondición	El usuario ha obtenido una contraseña segura generada por la aplicación	
Excepciones	Paso	Acción
	2b	El usuario introduce nuevos símbolos permitidos. Después sigue el flujo normal.
	2c	El usuario añade o elimina grupos de caracteres de las opciones. Después sigue el flujo normal.
	5b	El sistema muestra la opción de introducir contraseña (el usuario ha de haber accedido desde la creación o la edición de un perfil) o de copiarla al portapapeles. Después sigue el flujo normal.
Comentarios		

Tabla 19:Especificación del requisito US-03

US-04	Resolver notificación	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-04,OBJ-5	
Requisitos asociados	RU-05	
Descripción	El usuario podrá iniciar el proceso de resolución de una notificación desde la propia notificación.	
Precondición	El usuario ha recibido una notificación.	
Secuencia normal	Paso	Acción
	1	El usuario pulsa en la opción resolver de la notificación.
	2	La aplicación accede a una vista que permite cambiar la contraseña del perfil cuya contraseña ha caducado.
	3	El usuario introduce una nueva contraseña.
	4	La aplicación comprueba que la contraseña no es la misma.
	5	La aplicación aplica SHA-1 a la contraseña y comprueba que esta no se haya utilizado.
	6	La aplicación aplica SHA-1 a la antigua contraseña y la guarda en el archivo de contraseñas usadas.
	7	La aplicación guarda la nueva contraseña en memoria.
	8	La aplicación guarda la nueva contraseña en el archivo que guarda los perfiles.
	9	La aplicación informa al usuario de que la contraseña se ha cambiado.
	10	El caso de uso a finalizado con éxito
Postcondición		
Excepciones	Paso	Acción
	3b	El usuario realiza el US-03 para generar la contraseña.
	5b	La aplicación informa que la contraseña es la misma. Vuelve al paso 2

	6b	La aplicación informa de que la contraseña se ha usado con anterioridad. Vuelve al paso 2.
Comentarios		

Tabla 20:Especificación del requisito US-04

US-05	Visualizar lista de perfiles	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-2	
Requisitos asociados	RU-06,RU-07,RU-08,	
Descripción	El usuario podrá visualizar una lista de los perfiles que guardan las contraseñas y a donde pertenecen esas contraseñas.	
Precondición	El usuaria está en el menú principal.	
Secuencia normal	Paso	Acción
	1	El usuario accede a la opción visualizar perfiles.
	2	La aplicación muestra la lista de perfiles
	3	El caso de uso a finalizado con éxito.
Postcondición	El usuario está visualizando la lista de perfiles	
Excepciones	Paso	Acción
	3b	El usuario utiliza la opción de filtrar por servicio.
	4b	La aplicación muestra la lista de perfiles filtrada por servicios.
	5	El caso de uso a finalizado con éxito
	3c	El usuario utiliza la opción de filtrar por tipo de servicio.
	4c	La aplicación muestra la lista de perfiles filtrada por tipo de servicios. Continua con el paso 5.
Comentarios		

Tabla 21:Especificación del requisito US-05

US-06	Visualizar perfil completo	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-02	
Requisitos asociados	RU-09,RU-10,RU-11	
Descripción	El usuario podrá elegir un perfil de la lista completa para verlo en detalle y acceder a opciones de edición.	
Precondición	El usuario está visualizando la lista de perfiles.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona un perfil para ver en detalle.
	2	La aplicación muestra toda la información del perfil con la contraseña oculta.
	3	El caso de uso a finalizado con éxito
Postcondición	El usuario a visualizado un perfil completo.	
Excepciones	Paso	Acción
	3b	El usuario selecciona la opción para hacer visible la contraseña.
	4	La aplicación muestra la información del perfil con la contraseña visible.
	5	El caso de uso a finalizado con éxito
	5b	El usuario selecciona la opción para ocultar la contraseña. Continua con el paso 2.
	3c	El usuario selecciona la opción para copiar la contraseña al portapapeles. Continua con 5
Comentarios		

Tabla 22:Especificación del requisito US-06

US-07	Eliminar perfil	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-02	
Requisitos asociados	RU-12	
Descripción	El usuario podrá eliminar un perfil	
Precondición	El usuario está visualizando una lista de perfiles que contiene a dicho perfil, o está visualizando dicho perfil en detalle.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la opción eliminar perfil.
	2	La aplicación solicita confirmación para la eliminación del perfil.
	3	El usuario acepta la confirmación.
	4	La aplicación elimina los datos del perfil en memoria.
	5	La aplicación actualiza el archivo donde guarda los perfiles.
	6	La aplicación informa al usuario de que el perfil se ha eliminado.
	7	El caso de uso a finalizado con éxito.
Postcondición	Un perfil ha sido eliminado	
Excepciones	Paso	Acción
	3b	El usuario no acepta la confirmación.
	4b	El caso de uso finaliza sin éxito.
Comentarios		

Tabla 23:Especificación del requisito US-07

US-08	Editar perfil	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-02,OBJ-5	
Requisitos asociados	RU-13	
Descripción	Un Usuario podrá modificar la información de un perfil.	
Precondición	El usuario estará visualizando los detalles de ese perfil.	
Secuencia normal	Paso	Acción
	1	El usuario accede a la opción de editar perfil
	2	La aplicación muestra los datos del perfil en cuadros de texto editables con la contraseña oculta.
	3	El usuario cambia al menos un campo.
	4	El usuario usa la opción de guardar cambios.
	5	La aplicación guarda en memoria los cambios.
	6	La aplicación guarda los cambios en el archivo donde guarda los perfiles.
	7	La aplicación informa al usuario de que se han guardado los cambios.
	8	El caso de uso a finalizado con éxito.
Postcondición	Algún campo del perfil ha sido modificado	
Excepciones	Paso	Acción
	3b	El usuario realiza el US-03. Continua con el flujo normal.
	3c	El usuario no realiza ningún cambio.
	4	El usuario usa la opción de guardar cambios.
	5c	La aplicación informa de que no hay ningún cambio.
	6c	El caso de uso finaliza sin éxito.
	5d	(en caso de que uno de los campos sea la contraseña) La aplicación comprueba que la contraseña no es la misma.
	6d	La aplicación aplica SHA-1 a la contraseña y comprueba que esta no se haya utilizado.

	7d	La aplicación aplica SHA-1 a la antigua contraseña y la guarda en el archivo de contraseñas usadas. Vuelve al paso 5.
	6db	La aplicación informa que la contraseña es la misma. Vuelve al paso 2
	7db	La aplicación informa de que la contraseña se ha usado con anterioridad. Vuelve al paso 2.
Comentarios		

Tabla 24:Especificación del requisito US-08

US-09	Crear perfil	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-02,OBJ-5	
Requisitos asociados	RU-14,RU-15,RU-16	
Descripción	Un usuario puede crear un nuevo perfil para guardar la contraseña de un nuevo servicio.	
Precondición	El usuario está en el menú principal	
Secuencia normal	Paso	Acción
	1	El usuario accede a la opción crear un perfil.
	2	La aplicación muestra los campos necesarios en cuadros de texto editables.
	3	El usuario rellena al menos los campos obligatorios.
	4	El usuario utiliza la opción crear perfil.
	5	La aplicación comprueba que se han introducido los campos obligatorios
	6	La aplicación comprueba que no existe otro perfil del mismo servicio con el mismo identificador(correo)
	7	La aplicación aplica el algoritmo SHA-01 sobre la contraseña
	8	La aplicación comprueba que la contraseña no haya sido utilizada con anterioridad.
	9	La aplicación guarda el perfil en memoria

	10	La aplicación guarda el perfil en el archivo que guarda los perfiles
	11	La aplicación informa al usuario de que se ha creado el perfil.
	12	El caso de uso a finalizado con éxito.
Postcondición	Se ha creado un nuevo perfil	
Excepciones	Paso	Acción
	6b	La aplicación informa de que ya existe un perfil para esa cuenta.Vuelve al paso 2.
	7b	La aplicación informa de que faltan campos obligatorios. Vuelve al paso 2.
	9b	La aplicación informa de que la contraseña se ha utilizado con anterioridad. Vuelve al paso 2.
Comentarios		

Tabla 25:Especificación del requisito US-09

US-10	Volver al menú de inicio	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-01,OBJ-02	
Requisitos asociados	RU-17	
Descripción	El usuario podrá volver al menú de inicio, de manera que la aplicación vaciara la memoria y será necesario introducir de nuevo la contraseña maestra.	
Precondición	El usuario está en el menú principal.	
Secuencia normal	Paso	Acción
	1	El usuario utiliza la opción volver al menú de inicio.
	2	La aplicación pide confirmación indicando al usuario que para consultar los perfiles sería necesario volver a introducir la contraseña maestra.
	3	El usuario confirma.
	4	La aplicación borra la información de los perfiles de la memoria.
	5	La aplicación muestra el menú de inicio.
	6	El caso de uso a finalizado con éxito.

Postcondición		
Excepciones	Paso	Acción
	3b	El usuario cancela la vuelta al menú de inicio.
	4b	El caso de uso finaliza sin éxito.
Comentarios		

Tabla 26:Especificación del requisito US-10

4.1.6 Requisitos de información

ENT- 01		Configuración de la aplicación		Versión	1.0	
Definición		Elemento aglutinador de los ajustes de la aplicación así como datos fundamentales para su funcionamiento.				
Consideraciones						
ATRIBUTOS						
ID	Nombre	Descripción	Dominio	UNIQUE	NULL	Notas
	Contraseña	128 primeros bits de la contraseña pasada por el algoritmo SHA-384	VARCHAR (128)	SÍ	NO	
	Tiempo de expiración	Tiempo por defecto para la caducidad de las contraseñas en días	INT(3)	NO	NO	

Tabla 27:Especificación de la ENT-01

ENT- 02	Perfil de Servicios	Versión	1.0
Definición	Perfil que representa una cuenta gestionada por la aplicación en el que se almacenan los datos de acceso así como la información fundamental para describir dicho servicio.		
Consideraciones			

ATRIBUTOS						
ID	Nombre	Descripción	Dominio	UNIQUE	NULL	Notas
	Email	Email al que está vinculado este perfil	VARCHAR (128)	SÍ	NO	
	Nombre de usuario	Nombre de usuario	VARCHAR (128)	No	Si	
	Contraseña	Contraseña de acceso a ese servicio	VARCHAR (128)	Si	No	
	Expiración contraseña	Fecha de expiración de la contraseña	DATE	NO	NO	
	Servicio	Nombre descriptivo del servicio pj:Gmail	VARCHAR (128)	No	Si	No rellenar este dato restringe el uso de filtros
	Tipo de Servicio	Tipo de servicio pj: proveedor de correo	VARCHAR (128)	No	Si	No rellenar este dato restringe el uso de filtros

Tabla 28:Especificación de la ENT-02

ENT- 03	Contraseña antigua			Versión	1.0
Definición	Contraseña que ha sido usada con antelación				
Consideraciones					
ATRIBUTOS					
ID	Nombre	Descripción	Dominio	UNIQUE	NULL	Notas
	Contraseña	SHA-1 de la contraseña	VARCHAR (128)	SÍ	NO	
	Fecha límite de guardado	Fecha a partir de la cual se borrara esta entrada	DATE	NO	NO	

Tabla 29:Especificación de la ENT-03

4.1.7 Requisitos no funcionales*

Requisitos no funcionales de Seguridad:

NFS-01 Se deben almacenar las contraseñas cifradas bajo AES-256 utilizando como clave los 256 últimos bits de la salida del algoritmo SHA-384 de la contraseña maestra.

NFS- 05 Se deben almacenar los 128 primeros bits de la salida al algoritmo SHA-384 de la contraseña maestra codificados con AES - 256 utilizando los 256 últimos bits para identificar la contraseña maestra.

NFS-03 Se deben guardar contraseña anteriores bajo el algoritmo SHA-1.

NFS-04 Se debe bloquear el uso trás un periodo de inactividad de 15 min.

Requisitos no funcionales de usabilidad:

NFU-01 Se debe redactar un sencillo manual de usuario

NFU-02 Se debe redactar un sencillo manual de instalación

4.2 Diseño

4.2.1 Arquitectura lógica

Para la arquitectura lógica* de esta aplicación se va a seguir el Modelo Vista Controlador o MVC, si bien en esta iteración sólo se implementadas el Modelo y el controlador siendo la Vista implementada para el funcionamiento de la aplicación tras el transcurso de esta iteración provisional y formada por mensajes de salida en forma de menús.

En la capa de modelo se situarán las clases que definen la lógica de negocio, siendo en este caso los algoritmos de codificación y las clases que modelan las entidades que encapsulan la información.

En la capa Vista se presentarán los datos en esta primera iteración mediante mensajes de salida y posteriormente mediante una interfaz gráfica.

Por último la capa Controlador será la encargada de comunicar la vista con los procesos de negocio que realizan los elementos de la capa de modelo.

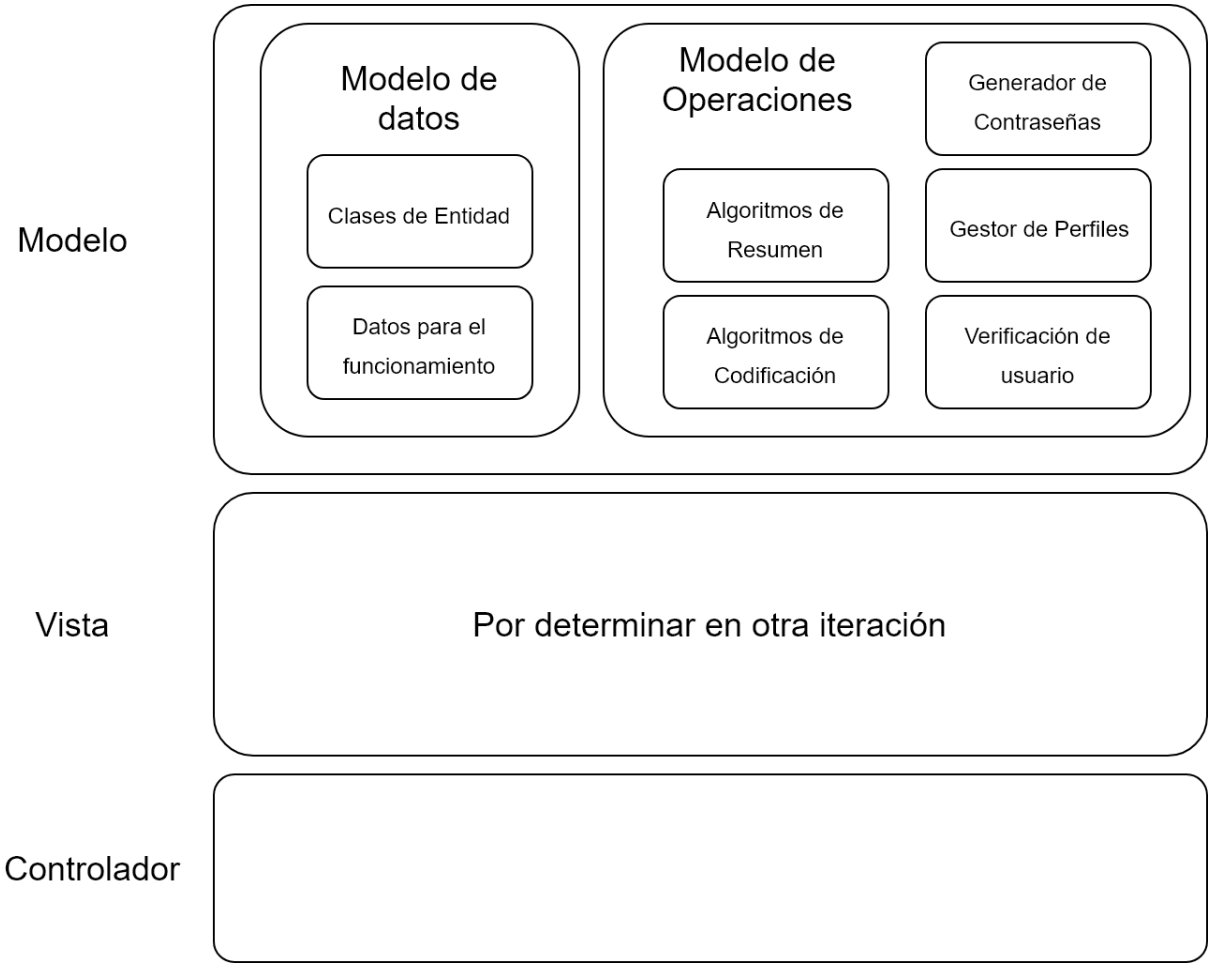


Figura 15:Especificación de la definición de la arquitectura lógica, primera iteración

4.2.2 Arquitectura física

La arquitectura física* de esta aplicación es bastante sencilla al ser un sistema aislado con la seguridad como prioridad, por lo tanto solo tendrá dos elementos diferenciados siendos estos un servidor que aloja los elementos necesarios para la instalación de la aplicación y el PC del usuario final que contendrá la aplicación instalada y los ficheros que almacenan los datos.

4.2.3 Diagrama de clases

A continuación se muestra el diagrama de clases* para la primera iteración:

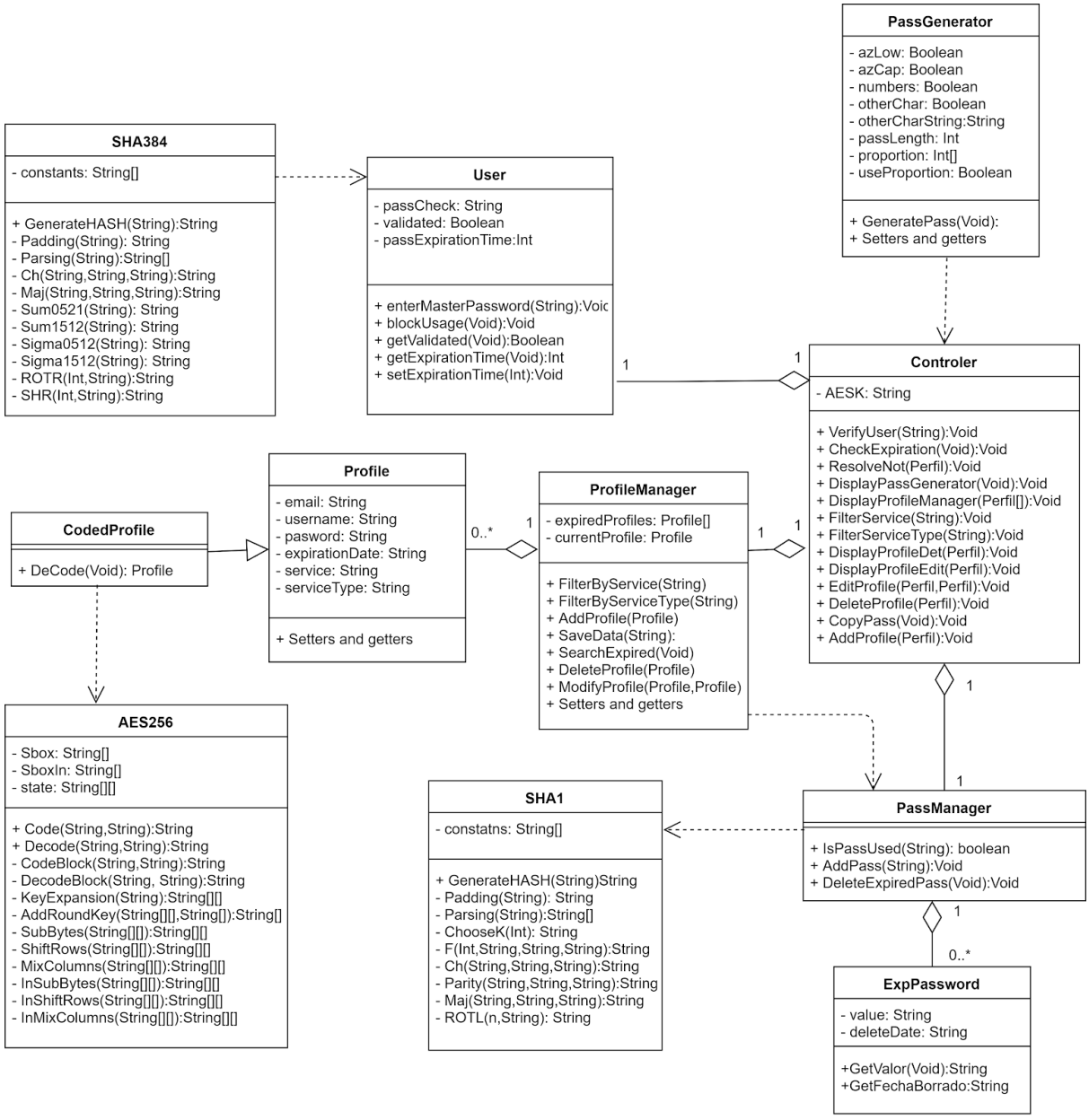


Figura 16:Diagrama de Clases

4.3 Implementación

Se han añadido métodos “SaveData” a las clases “PassManager” y “User” similares a la contemplada para “ProfileManager”, ya que modelan datos que requieren persistencia de una manera similar al “ProfileManager” para cubrir una necesidad que no se tuvo en cuenta inicialmente.

La implementación de las clases involucradas en operaciones criptográficas se ha realizado en base a los estándares definidos. Estos estándares utilizan una combinación de pseudo código y explicaciones detalladas de las operaciones de los algoritmos.

Ejemplo de pseudocódigo en el estándar para SHA-1 frente al código implementado:

```
for(int i=0;i<n;i++){
    for(int t=0;t<16;t++){
        String w = "";
        for(int j=32*t;j<32*(t+1);j++){
            w += ProccessedMessageBlocks[i].charAt(j);
        }
        W[t] = Long.parseLong(w, 2);
    }
    for(int t=16;t<80;t++){
        W[t] = ROTL(1,BitXOR(BitXOR(W[t-3],W[t-8]),
                             BitXOR(W[t-14],W[t-16])));
    }

    long a = H[0];
    long b = H[1];
    long c = H[2];
    long d = H[3];
    long e = H[4];
    long T;

    for(int t=0;t<80;t++){
        T = Suma2W(Suma2W(Suma2W(Suma2W(ROTL(5,a),F(t,b,c,d),32),e,32),
                                                  Choosek(t),32),W[t],32) ;
        e = d;
        d = c;
        c = ROTL(30,b);
        b = a;
        a = T;
```

Figura 17: Fragmento de la implantación de SHA-1

For i=1 to N:

{

1. Prepare the message schedule, $\{W_t\}$:

$$W_t = \begin{cases} M_t^{(i)} & 0 \le t \le 15 \\ ROTL^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) & 16 \le t \le 79 \end{cases}$$

2. Initialize the five working variables, ***a***, ***b***, ***c***, ***d***, and ***e***, with the (i-1)st hash value:

$$a = H_0^{(i-1)}$$
$$b = H_1^{(i-1)}$$
$$c = H_2^{(i-1)}$$
$$d = H_3^{(i-1)}$$
$$e = H_4^{(i-1)}$$

3. For t=0 to 79:

{

$$T = ROTL^5(a) + f_t(b,c,d) + e + K_t + W_t$$
$$e = d$$
$$d = c$$
$$c = ROTL^{30}(b)$$
$$b = a$$
$$a = T$$

}

}

Figura 18: Pseudocódigo del estándar para SHA-1

Ejemplo de explicación de un método de AES-256 frente al método implementado:

5.1.2 `ShiftRows()` Transformation

In the **`ShiftRows()`** transformation, the bytes in the last three rows of the State are cyclically shifted over different numbers of bytes (offsets). The first row, $r = 0$, is not shifted.

Specifically, the **`ShiftRows()`** transformation proceeds as follows:

$$s'_{r,c} = s_{r,(c+shift(r,Nb)) \bmod Nb} \quad \text{for } 0 < r < 4 \quad \text{and} \quad 0 \le c < \mathbf{Nb}, \tag{5.3}$$

where the shift value $shift(r,Nb)$ depends on the row number, r, as follows (recall that $\mathbf{Nb} = 4$):

$$shift(1,4) = 1; \quad shift(2,4) = 2; \quad shift(3,4) = 3. \tag{5.4}$$

This has the effect of moving bytes to "lower" positions in the row (i.e., lower values of c in a given row), while the "lowest" bytes wrap around into the "top" of the row (i.e., higher values of c in a given row).

Figure 8 illustrates the **`ShiftRows()`** transformation.

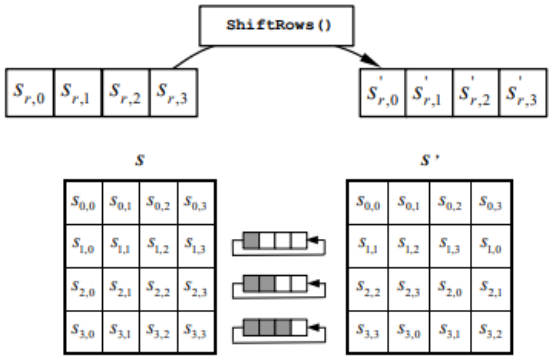


Figure 8. `ShiftRows()` cyclically shifts the last three rows in the State.

Figura 19: Explicación del estándar de un método de AES

```
private static int[][] ShiftRows(int[][] state) {
    int [][] stateShift= new  int[4][4];
    stateShift[0][0] = state[0][0];
    stateShift[1][0] = state[1][0];
    stateShift[2][0] = state[2][0];
    stateShift[3][0] = state[3][0];

    stateShift[0][1] = state[1][1];
    stateShift[1][1] = state[2][1];
    stateShift[2][1] = state[3][1];
    stateShift[3][1] = state[0][1];

    stateShift[0][2] = state[2][2];
    stateShift[1][2] = state[3][2];
    stateShift[2][2] = state[0][2];
    stateShift[3][2] = state[1][2];

    stateShift[0][3] = state[3][3];
    stateShift[1][3] = state[0][3];
    stateShift[2][3] = state[1][3];
    stateShift[3][3] = state[2][3];
    return stateShift;
}
```

Figura 20: Implementación de un método de AES

Además de los métodos principales también ha sido necesario implementar métodos para realizar operaciones matemáticas que forman parte de los algoritmos.

```
private static long Suma2W(long x, long y, int w){
    long z = (long) ((x+y)%(Math.pow(2, w)));
    return z;
}
```

Figura 21: Código de un método auxiliar

Para resolver la validación del usuario y el cálculo de la clave AES para poder recuperar la información cifrada se utiliza un único método que se detalla a continuación:

```
public String enterMasterPassword(String clave) throws FileNotFoundException{
    String claveExp = SHA384.GenerateHASH(clave);
    String AESK = "";
    String checkBits ="";
    for(int i=0;i<32;i++){
            checkBits += claveExp.charAt(i);
        }
    for(int i=32;i<96;i++){
            AESK += claveExp.charAt(i);
       }
    if(passCheck==null){
        ObjectOutputStream oos = null;
        try {
            passCheck = "";
            passCheck = AES256.Code(checkBits, AESK);
            oos = new ObjectOutputStream(new FileOutputStream("Usuario.bin"));
            oos.writeObject(this);
            oos.close();
            return AESK;
        } catch (IOException ex) {
            System.out.println("excepcion");
            return null;
        }
    }else{
        if(passCheck.equals(AES256.Code(checkBits, AESK))){
            validated = true;
            return AESK;
        }else{
            return null;
        }
    }
```

Figura 22: Código del método enterMasterPassword

Este método es usado por el controlador para comprobar si la contraseña introducida en la vista es válida y si lo es cargar los datos utilizando la clave AES que recibe, o en caso contrario volver a pedir una nueva contraseña mediante la vista.

En cuanto al funcionamiento del propio método, en primer lugar calcula el string de comprobación que se calcula utilizando la salida del algoritmo de hashing SHA 384 siguiendo el procedimiento explicado con anterioridad.

```
String claveExp = SHA384.GenerateHASH(clave);
String AESK = "";
String checkBits ="";
for(int i=0;i<32;i++){
        checkBits += claveExp.charAt(i);
    }
for(int i=32;i<96;i++){
        AESK += claveExp.charAt(i);
   }
```

Figura 23: Detalle del método enterMasterPassword (1)

Una vez calculado dicho comprobante, el método procede de dos maneras distintas dependiendo de si ya existe un usuario o por el contrario no.

En caso de que exista, compara el comprobante calculado con el almacenado. Y si son iguales devuelve la clave AES ya calculada, mientras que si no lo son devuelve el valor "null".

```
}else{
    if(passCheck.equals(AES256.Code(checkBits, AESK))){
        validated = true;
        return AESK;
    }else{
        return null;
    }
}
```

Figura 24: Detalle del método enterMasterPassword (2)

Si no existe el método crea el fichero donde se almacena el usuario y trás dar al comprobante el valor calculado, guarda al usuario en dicho fichero. Una vez guardado el nuevo usuario devuelve la clave AES calculada. La excepción presente en el método protege al programa frente a posibles errores de escritura pero no aporta información adicional tratando al caso como una validación fallida.

```
if(passCheck==null){
    ObjectOutputStream oos = null;
    try {
        passCheck = "";
        passCheck = AES256.Code(checkBits, AESK);
        oos = new ObjectOutputStream(new FileOutputStream("Usuario.bin"));
        oos.writeObject(this);
        oos.close();
        return AESK;
    } catch (IOException ex) {
        System.out.println("excepcion");
        return null;
    }
}else{
```

Figura 25: Detalle del método enterMasterPassword (3)

Del resto de elementos presentes en la aplicación el más relevante es el generador de contraseñas. Al margen de los métodos auxiliares para configurar el generador los métodos relevantes son los tres siguientes:

►Validate:

```
private boolean validate(String pass){
    boolean unaMin = pass.matches(".*[a-z]+.*");
    if(!azLow || proportion[0]==0) unaMin = true;
    boolean unaMay = pass.matches(".*[A-Z]+.*");
    if(!azCap || proportion[1]==0) unaMay = true;
    boolean unNum = pass.matches(".*[0-9]+.*");
    if(!numbers || proportion[2]==0) unNum = true;
    boolean unEspC = false;
    for(int i=0;i<otherCharString.length();i++){
        unEspC = unEspC || pass.contains(""+otherCharString.charAt(i));
    }
    if(!otherChar || proportion[3]==0) unEspC = true;
    return unaMin && unaMay && unNum && unEspC;
}
```

Figura 26: Código del método Validate

►GeneratePass:

```
public String GeneratePass(){
    if(useProportion){
        return GeneratePassWithProportion();
    }else{
        String passChar = "";
        if(azLow) passChar+=azLowS;
        if(azCap) passChar += azCapS;
        if(numbers) passChar += numbersS;
        if (otherChar) passChar += otherCharString;
        String pass = "";
        sr.nextBytes(new byte[1]);
        for( int i =0;i<passLength;i++){
            int aux = sr.nextInt(passChar.length());
            pass += passChar.charAt(aux);
        }
        sr.setSeed(System.currentTimeMillis());
        if(validate(pass)) return pass;
        else return GeneratePass();
    }
}
```

Figura 27: Código del método GeneratePass

►GeneratePassWithProportion:

```
private String GeneratePassWithProportion(){
    System.out.println("Hola?????");
    String pass = "";
    sr.nextBytes(new byte[1]);
    String passChar;
    double propSum = proportion[0]+proportion[1]+proportion[2]+proportion[3];
    for( int i =0;i<passLength;i++){
        System.out.println(i);
        double auxI = sr.nextDouble();
        double limS;
        if(auxI<proportion[0]/propSum){
            passChar = numbersS;
            limS = proportion[0]/propSum;
        }else if(auxI<(proportion[0]+proportion[1])/propSum){
            passChar = azLowS;
            auxI += -(proportion[0]/propSum);
            limS = proportion[1]/propSum;
        }else if(auxI<(proportion[0]+proportion[1]+proportion[2])/propSum){
            passChar = azCapS;
            auxI += -((proportion[0]+proportion[1])/propSum);
            limS = proportion[2]/propSum;
        }else{
            passChar = otherCharString;
            auxI += -((proportion[0]+proportion[1]+proportion[2])/propSum);
            limS = proportion[3]/propSum;
        }
        pass += passChar.charAt((int) (passChar.length()*auxI/limS));
    }
    sr.setSeed(System.currentTimeMillis());
    if(validate(pass)) return pass;
    else return GeneratePassWithProportion();    }
```

Figura 28: Código del método GeneratePassWithProportion

El primero de estos métodos es el encargado de garantizar que a pesar de la aleatoriedad la contraseña generada cumple la configuración asegurando que está contiene al menos un caracteres de cada una de las categorías seleccionadas (mayusculas, minusculas, numeros o simbolos especiales).

El segundo método es el algoritmo de generación básico cuando no se utiliza una proporción específica. Este método itera para la longitud de la contraseña extrayendo un elemento de la cadena que concatena todas las subcadenas de posibles caracteres, de manera que un número pseudo aleatorio obtenido mediante el objeto sr y el método nextInt (08), es el que define el carácter que extraemos. Además este método delega en el tercer método(“GeneratePassWithProportion”), en caso de que la configuración lo establezca (“useProportion” tiene como valor “true”).

Por último tenemos el método “GeneratePassWithProportion”. Este método pondera en función de la proporción entre cuatro valores que representan cada uno de los posibles conjuntos de caracteres.

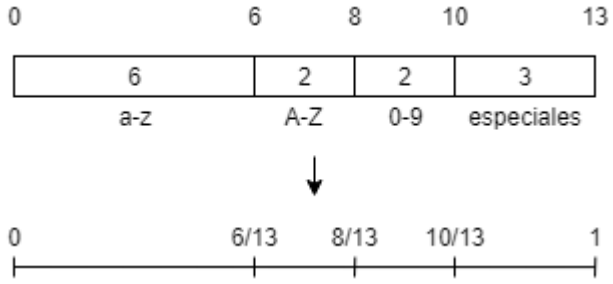


Figura 29: Esquema del funcionamiento del algoritmo para la generación de contraseñas usando la proporción, parte 1

Una vez establecidos los puntos de corte mediante dicha extrapolación se genera un número pseudo aleatorio con “sr” y el método “ nextDouble”, que se compara con los puntos de corte para determinar de qué cadena se escoge un carácter.

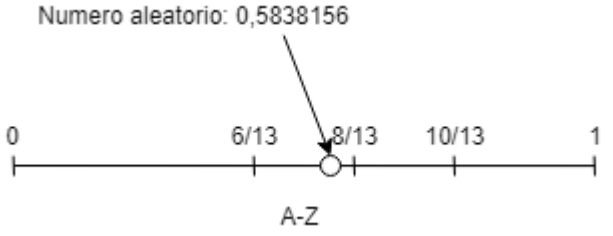


Figura 30: Esquema del funcionamiento del algoritmo para la generación de contraseñas usando la proporción, parte 2

Una vez escogida la cadena se hace una equivalencia entre el límite inferior y el 0 y entre el límite superior y la longitud de la cadena, y se obtiene la posición del número pseudo

aleatorio relativa a los límites para extrapolar a la longitud de la cadena y tras truncar el número se extrae el elemento de la cadena en dicha posición.

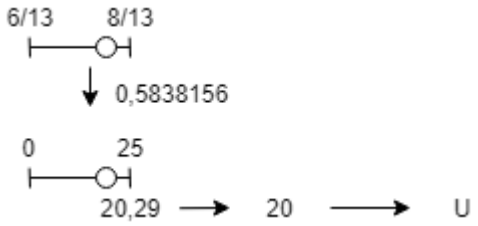


Figura 31: Esquema del funcionamiento del algoritmo para la generación de contraseñas usando la proporción, parte 3

Esta operación se repite para la longitud de la contraseña.

4.4 Pruebas

4.4.1 Pruebas de caja blanca

Esta aplicación tiene métodos especialmente sensibles dada su alta carga algorítmica por lo que se han realizado pruebas de caja blanca* para el correcto funcionamiento de todos los métodos de las clases criptográficas “SHA-1”, “SHA-384” y “AES-256”.

Estas clases utilizan algoritmos que iteran sobre los datos un número determinado de veces, para comprobar su correcto funcionamiento se ha procedido a la realización de pruebas mostrando el resultado de todas las iteraciones que realiza durante el proceso. Cotejando el valor de todas la variables en cada una de las iteraciones con muestras ya existentes como la referencia 06 o las proporcionadas en la asignatura Protocolos y comunicaciones seguras.

4.4.2 Pruebas de caja negra

A continuación se detallan las pruebas de caja negra* realizadas:

PCN-01 Generar Contraseña	
Objetivo	Comprobar que el generador de contraseñas funciona con la configuración por defecto.
Precondiciones	Estar en el generador sin haber cambiado la configuración
Datos de entrada	Ninguno
Acción esperada	Creación de una contraseña de 12 dígitos con minúsculas, mayúsculas, números y caracteres especiales
Resultado	Correcto

Tabla 30: Prueba de caja negra PCN-01

PCN-02 Generar Contraseña personalizando la configuración	
Objetivo	Comprobar que el generador de contraseñas funciona con una configuración diferente a la configuración por defecto.
Precondiciones	Estar en el generador y haber cambiado la configuración.
Datos de entrada	Uso de dígitos numéricos de true a false Uso de minúsculas de true a false Longitud cambiada a 8
Acción esperada	Creación de una contraseña de 8 dígitos con mayúsculas y caracteres especiales
Resultado	Correcto

Tabla 31: Prueba de caja negra PCN-02

PCN-03 Generar Contraseña personalizando la configuración	
Objetivo	Comprobar que el generador de contraseñas funciona con una configuración diferente a la configuración por defecto.
Precondiciones	Estar en el generador y haber cambiado la configuración.
Datos de entrada	Generar con proporción Proporción [6,2,2,2}
Acción esperada	Creación de una contraseña de 12 dígitos con mayoritariamente mayúsculas
Resultado	Correcto

Tabla 32: Prueba de caja negra PCN-03

PCN-04 Introducción de una contraseña maestra no válida	
Objetivo	Comprobar que identificar usuario funciona correctamente
Precondiciones	Que esté establecida una contraseña maestra No estar identificado
Datos de entrada	password
Acción esperada	Mensaje indicando que la contraseña es incorrecta Solicitud de introducir de nuevo una contraseña
Resultado	Correcto

Tabla 33: Prueba de caja negra PCN-04

PCN-05 Introducción de una contraseña maestra válida	
Objetivo	Comprobar que identificar usuario funciona correctamente
Precondiciones	Que esté establecida una contraseña maestra No estar identificado
Datos de entrada	prueba
Acción esperada	El usuario ha sido identificado y se muestra la lista de perfiles
Resultado	Correcto

Tabla 34: Prueba de caja negra PCN-05

PCN-06 Añadir perfil con contraseña generada	
Objetivo	Comprobar que se puede añadir un perfil correctamente, y que la generación de contraseñas desde el creador de perfiles.
Precondiciones	Usuario identificado y visualizando la lista de perfiles
Datos de entrada	email: hugo@email.com nombre:Hugo servicio: test1 tipo de servicio: test contraseña: “usar generador”
Acción esperada	Se muestra la lista de perfiles con el nuevo perfil añadido
Resultado	Correcto

Tabla 35: Prueba de caja negra PCN-06

PCN-07 Añadir perfil con contraseña introducida manualmente	
Objetivo	Comprobar que se puede añadir un perfil correctamente.
Precondiciones	Usuario identificado y visualizando la lista de perfiles
Datos de entrada	email: hugo@email.com nombre:Hugo servicio: test2 tipo de servicio: test contraseña: pass
Acción esperada	Se muestra la lista de perfiles con el nuevo perfil añadido
Resultado	Correcto

Tabla 36: Prueba de caja negra PCN-07

PCN-08 Añadir perfil con contraseña usada	
Objetivo	Comprobar que no se puede usar una contraseña que ya está en uso.
Precondiciones	Usuario identificado y visualizando la lista de perfiles
Datos de entrada	email: hugo@email.com nombre:Hugo servicio: test3 tipo de servicio: test contraseña: pass
Acción esperada	Se muestra un mensaje indicando que la contraseña está o ha estado en uso y se solicita una nueva contraseña
Resultado	Correcto

Tabla 37: Prueba de caja negra PCN-08

PCN-09 Filtrar lista por servicios	
Objetivo	Comprobar que la funcionalidad de filtrado por servicio funciona
Precondiciones	Usuario identificado y visualizando la lista de perfiles.
Datos de entrada	filtro: test1
Acción esperada	La aplicación muestra una lista de los perfiles de ese servicio.
Resultado	Correcto

Tabla 38: Prueba de caja negra PCN-09

PCN-10 Filtrar lista por tipo de servicios	
Objetivo	Comprobar que la funcionalidad de filtrado por tipo de servicio funciona.
Precondiciones	Usuario identificado y visualizando la lista de perfiles.
Datos de entrada	filtro: test
Acción esperada	La aplicación muestra una lista de los perfiles de ese tipo de servicios.
Resultado	Correcto

Tabla 39: Prueba de caja negra PCN-10

PCN-11 Modificar perfil	
Objetivo	Comprobar que la modificación de perfiles es posible y funciona correctamente
Precondiciones	Usuario identificado y visualizando un perfil en detalle.
Datos de entrada	nueva fecha de expiración: 10
Acción esperada	La fecha de expiración se ha cambiado a dentro de 10 días.
Resultado	Correcto

Tabla 40: Prueba de caja negra PCN-11

PCN-12 Modificar perfil contraseña usada	
Objetivo	Comprobar que la modificación de perfiles es posible y funciona correctamente
Precondiciones	Usuario identificado y visualizando un perfil en detalle.
Datos de entrada	nueva contraseña: pass
Acción esperada	Se muestra un mensaje indicando que la contraseña esta o ha estado en uso. Se pide una nueva contraseña.
Resultado	Correcto

Tabla 41: Prueba de caja negra PCN-12

PCN-13 Modificar perfil sin guardar cambios	
Objetivo	Comprobar que la modificación de perfiles se puede abortar tras haber realizado cambios.
Precondiciones	Usuario identificado y visualizando un perfil en detalle.
Datos de entrada	Nuevo nombre de usuario: cambio Seleccionar opción salir sin guardar
Acción esperada	La aplicación muestra la lista de perfiles, el perfil modificado no ha sufrido ningún cambio.
Resultado	Correcto

Tabla 42: Prueba de caja negra PCN-13

PCN-14 Eliminar perfil	
Objetivo	Comprobar que la eliminación de perfiles es posible y funciona correctamente
Precondiciones	Usuario identificado y visualizando un perfil en detalle.
Datos de entrada	Seleccionar la opción eliminar.
Acción esperada	El perfil del que se estaban viendo los detalles se ha eliminado, y se muestra la lista de perfiles.
Resultado	Correcto

Tabla 43: Prueba de caja negra PCN-14

PCN-15 Visualizar notificaciones sin notificaciones pendientes	
Objetivo	Comprobar que el resumen de notificaciones funciona correctamente.
Precondiciones	Usuario identificado, no existe ninguna notificación pendiente.
Datos de entrada	Seleccionar la opción notificaciones.
Acción esperada	Se muestra un mensaje indicando que no hay notificaciones pendientes
Resultado	Correcto

Tabla 44: Prueba de caja negra PCN-15

PCN-16 Visualizar notificaciones con notificaciones pendientes	
Objetivo	Comprobar que el resumen de notificaciones funciona correctamente.
Precondiciones	Usuario identificado, existen notificaciones pendientes.
Datos de entrada	Seleccionar la opción notificaciones.
Acción esperada	Se muestra una lista con las notificaciones dando la opción de seleccionarlas para resolverlas.
Resultado	Correcto

Tabla 45: Prueba de caja negra PCN-16

PCN-17 Resolver notificación	
Objetivo	Comprobar que la resolución de notificaciones funciona correctamente.
Precondiciones	Usuario identificado, visualizando la lista de notificaciones ocn notificaciones pendientes.
Datos de entrada	Seleccionar la notificación a resolver. Nueva contraseña: “notificación”
Acción esperada	Se resuelve la notificación tras cambiar la contraseña y se muestra la lista de perfiles.
Resultado	Correcto

Tabla 46: Prueba de caja negra PCN-17

PCN-18 Resolver notificación de manera indirecta	
Objetivo	Comprobar que al modificar un perfil con un cambio que afecta a sus notificaciones, estas se resuelven.
Precondiciones	Usuario identificado, viendo en detalle un perfil con una notificación pendiente
Datos de entrada	nueva contraseña: “pendiente”
Acción esperada	Se realiza un cambio de contraseña para ese perfil, y la notificación se ha resuelto no mostrandose en la lista de notificaciones.
Resultado	Correcto

Tabla 47: Prueba de caja negra PCN-18

Capítulo 5

Segunda Iteración

5.1 Análisis

5.1.1 Características

A continuación se muestra el árbol de características que aglutina la iteración anterior (en gris) con la actual (en negro):

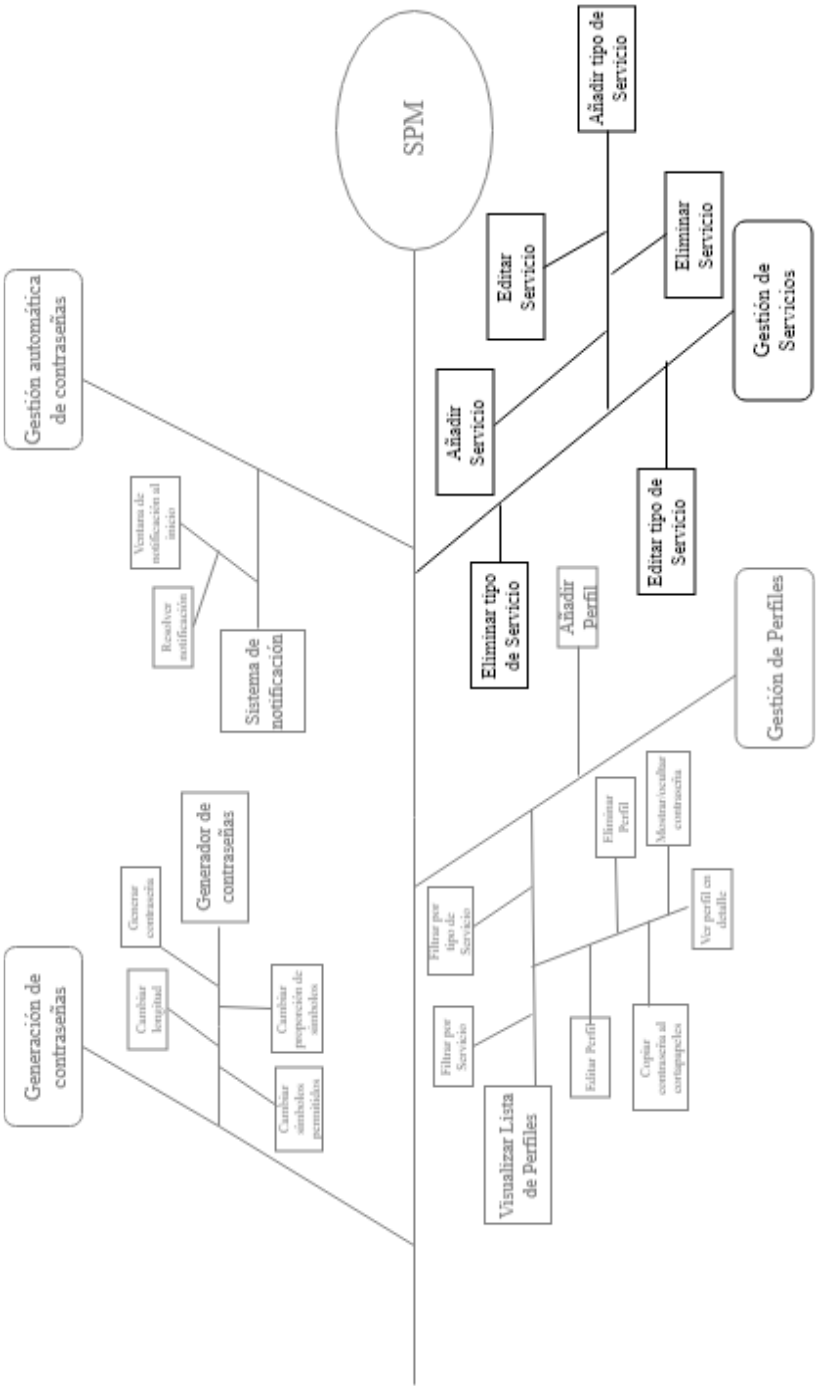


Figura 32 :Árbol de características de la segunda iteración

5.1.2 Actores

No se ha añadido ningún nuevo actor en esta iteración.

5.1.3 Requisitos de usuario

Los requisitos de usuario de esta iteración son los siguientes:

RU-18: Un usuario podrá añadir un tipo de servicio.
RU-19: Un usuario podrá editar un tipo de servicio que no pertenezca a los tipos por defectos.
RU-20: Un usuario podrá eliminar un tipo de servicio que no pertenezca a los tipos por defecto.
RU-21: Un usuario podrá añadir un servicio dentro del tipo de servicio que lo engloba.
RU-22: Un usuario podrá editar un servicio que no pertenezca a los servicios por defecto .
RU-23: Un usuario podrá eliminar un servicio que no pertenezca a los servicios por defecto.
RU-24: Un usuario podrá ver una lista de todos los servicios agrupados por tipo de servicio al que pertenecen.

Además se añaden los siguientes requisitos relativos a la iteración anterior:

RU-25: Un usuario podrá elegir el tiempo de caducidad por defecto.
RU-26: Un usuario podrá elegir el tiempo de inactividad para la desconexión .
RU-27: Un usuario podrá activar/desactivar la desconexión por inactividad.
RU-28: Un usuario podrá borrar todos sus datos.
RU-29: Un usuario podrá ver los ajustes actuales.
RU-30: Un usuario podrá visualizar una lista con las notificaciones pendientes

Los cambios producidos en esta iteración eliminan la necesidad de los siguientes requisitos:
~~RU-16: Un usuario podrá seleccionar el tipo de servicio de un perfil.~~

5.1.4 Diagrama de casos de uso

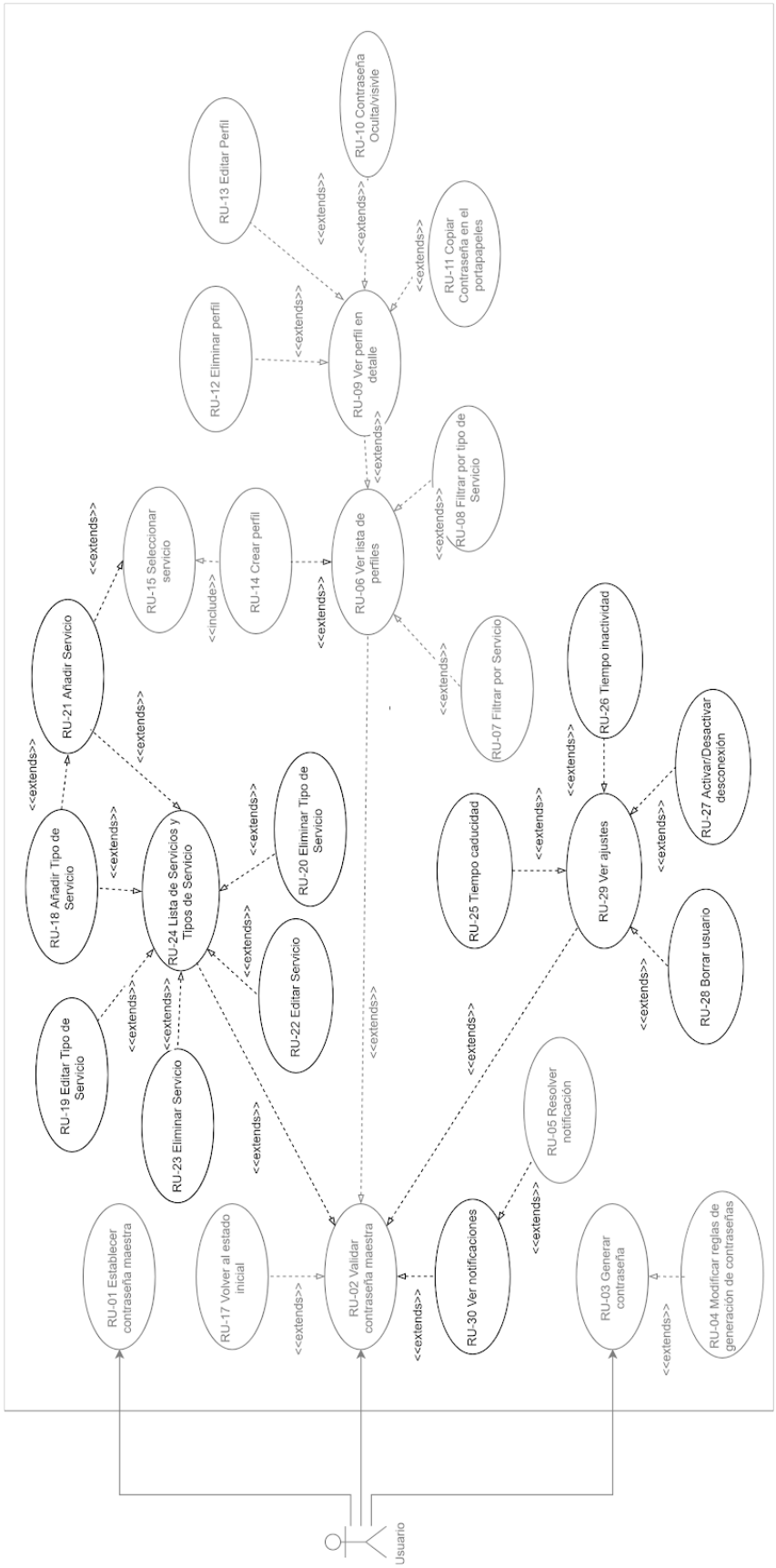


Figura 33:Diagrama de Casos de Uso

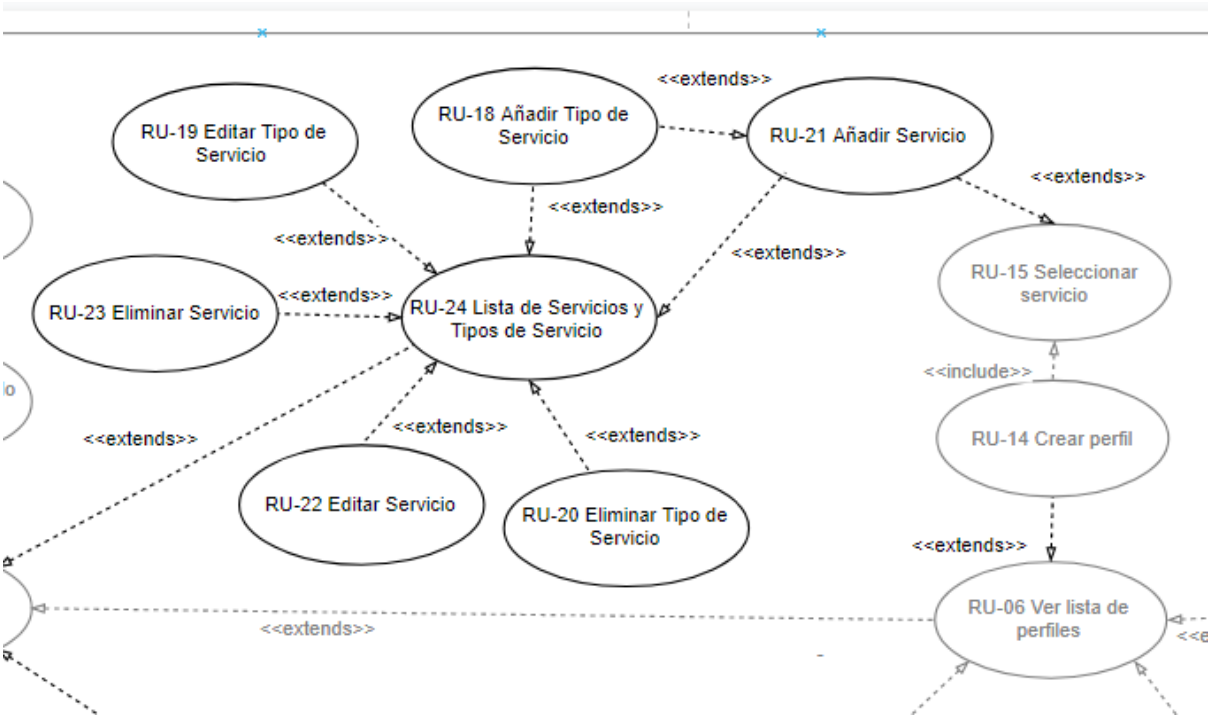


Figura 34: Detalle del Diagrama de Casos de Uso (Administración de servicios)

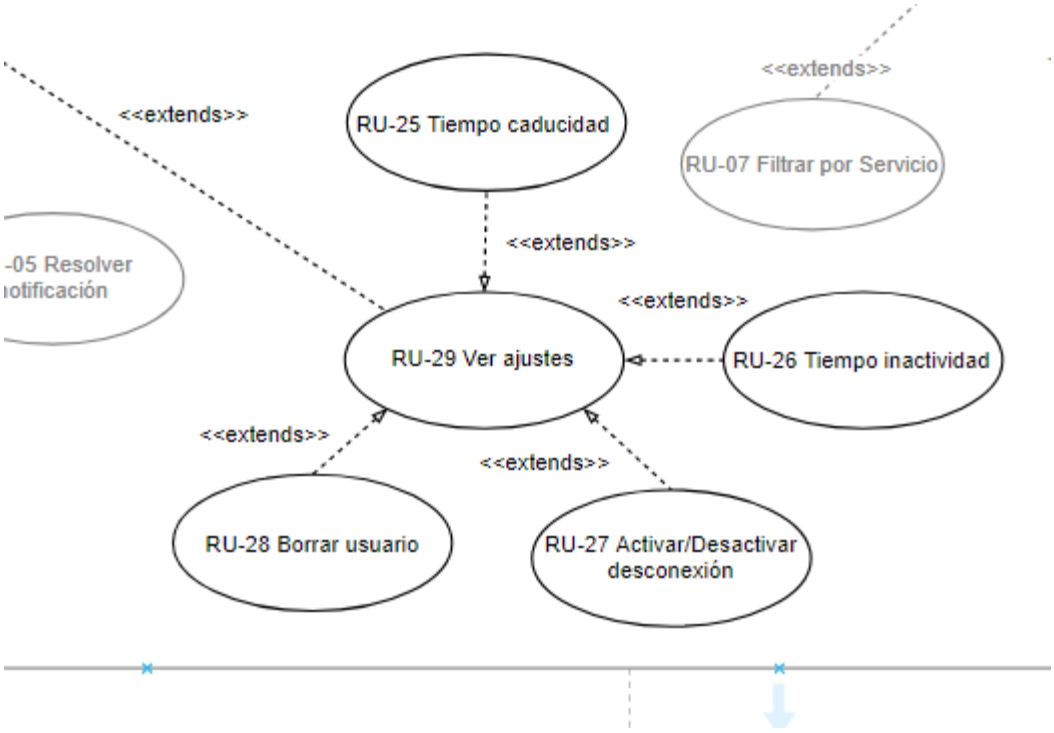


Figura 35:Detalle del Diagrama de Casos de Uso(Ajustes)

5.1.5 Especificación de requisitos de Usuario

US-04	Resolver notificación	
Versión	1.1	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-04,OBJ-5	
Requisitos asociados	RU-05	
Descripción	El usuario podrá iniciar el proceso de resolución de una notificación desde la propia notificación, o desde la lista de notificaciones.	
Precondición	El usuario ha recibido una notificación o el usuario visualiza las notificaciones	
Secuencia normal	Paso	Acción
	1	El usuario pulsa en la opción resolver de la notificación.
	2	La aplicación accede a una vista que permite cambiar la contraseña del perfil cuya contraseña ha caducado.
	3	El usuario introduce una nueva contraseña.
	4	La aplicación comprueba que la contraseña no es la misma.
	5	La aplicación aplica SHA-1 a la contraseña y comprueba que esta no se haya utilizado.
	6	La aplicación aplica SHA-1 a la antigua contraseña y la guarda en el archivo de contraseñas usadas.
	7	La aplicación guarda la nueva contraseña en memoria.
	8	La aplicación guarda la nueva contraseña en el archivo que guarda los perfiles.
	9	La aplicación informa al usuario de que la contraseña se ha cambiado.
	10	El caso de uso a finalizado con éxito
Postcondición		
Excepciones	Paso	Acción
	3b	El usuario realiza el US-03 para generar la contraseña.
	5b	La aplicación informa que la contraseña es la misma. Vuelve al

		paso 2
	6b	La aplicación informa de que la contraseña se ha usado con anterioridad. Vuelve al paso 2.
Comentarios		

Tabla 48:Especificación del US-04

US-05	Visualizar lista de perfiles	
Versión	1.1	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-2	
Requisitos asociados	RU-06,RU-07,RU-08,	
Descripción	El usuario podrá visualizar una lista de los perfiles que guardan las contraseñas y a donde pertenecen esas contraseñas.	
Precondición	El usuaria está en el menú principal.	
Secuencia normal	Paso	Acción
	1	El usuario accede a la opción visualizar perfiles.
	2	La aplicación muestra la lista de perfiles
	3	El caso de uso a finalizado con éxito.
Postcondición	El usuario está visualizando la lista de perfiles	
Excepciones	Paso	Acción
	3b	El usuario utiliza la opción de filtrar por servicio.
	4b	La aplicación muestra la lista de perfiles filtrada por servicios.
	5	El caso de uso a finalizado con éxito
	3c	El usuario utiliza la opción de filtrar por tipo de servicio.
	4c	La aplicación muestra la lista de perfiles filtrada por tipo de servicios. Continua con el paso 5.
	1b	El usuario se identifica correctamente
Comentarios	La lista de perfiles es la vista por defecto trás una identificación exitosa.	

Tabla 49:Especificación del US-05

US-09	Crear perfil	
Versión	2.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-02,OBJ-5	
Requisitos asociados	RU-14,RU-15	
Descripción	Un usuario puede crear un nuevo perfil para guardar la contraseña de un nuevo servicio.	
Precondición	El usuario está en el menú principal	
Secuencia normal	Paso	Acción
	1	El usuario accede a la opción crear un perfil.
	2	La aplicación muestra los campos necesarios en cuadros de texto editables.
	3	El usuario rellena al menos los campos obligatorios.
	4	El usuario utiliza la opción crear perfil.
	5	La aplicación comprueba que se han introducido los campos obligatorios
	6	La aplicación comprueba que no existe otro perfil del mismo servicio con el mismo identificador(correo)
	7	La aplicación aplica el algoritmo SHA-01 sobre la contraseña
	8	La aplicación comprueba que la contraseña no haya sido utilizada con anterioridad.
	9	La aplicación guarda el perfil en memoria
	10	La aplicación guarda el perfil en el archivo que guarda los perfiles
	11	La aplicación informa al usuario de que se ha creado el perfil.
	12	El caso de uso a finalizado con éxito.
Postcondición	Se ha creado un nuevo perfil	
Excepciones	Paso	Acción
	6b	La aplicación informa de que ya existe un perfil para esa cuenta.Vuelve al paso 2.
	7b	La aplicación informa de que faltan campos obligatorios. Vuelve al paso 2.

	9b	La aplicación informa de que la contraseña se ha utilizado con anterioridad. Vuelve al paso 2.
	9c	Se inicia el US-14(Añadir servicio) . Se continúa con el paso 10 una vez finalizado el requisito de usuario.
Comentarios		

Tabla 50:Especificación del US-09

US-11	Añadir tipo de servicio	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-06	
Requisitos asociados	RU-18	
Descripción	El usuario podrá añadir tipos de servicio a los ya existentes.	
Precondición	El usuario está visualizando la lista de servicios. O el usuario selecciona nuevo tipo de servicio en la creación de un servicio.	
Secuencia normal	Paso	Acción
	1	El usuario clickea en la opción añadir tipo de servicio
	2	La aplicación muestra la ventana de creación de tipos de servicio
	3	El usuario rellena los campos.
	4	El usuario usa la opción crear tipo de servicio
	5	La aplicación comprueba que no existe otro tipo de servicio con el mismo nombre.
	6	La aplicación crea el tipo de servicio
	7	La aplicación guarda el tipo de servicio
	8	La aplicación muestra la lista de servicios
	9	El caso de uso finaliza con éxito.
Postcondición	Se ha creado un nuevo tipo de servicio.	
Excepciones	Paso	Acción
	1b	El usuario usa la opción crear servicio habiendo seleccionado nuevo tipo de servicio, como tipo de servicio para el servicio a crear.

	4b	El usuario pulsa la opción volver. El caso de uso finaliza sin éxito.
	6b	La aplicación notifica al usuario de que ya existe un tipo de servicio con ese nombre. vuelve al paso 2.
Comentarios		

Tabla 51:Especificación del US-11

US-12	Modificar tipo de servicio	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-06	
Requisitos asociados	RU-19	
Descripción	El usuario podrá modificar un tipo de servicio que no sea por defecto.	
Precondición	El usuario está visualizando la lista de servicios.	
Secuencia normal	Paso	Acción
	1	El usuario pulsa la opción editar tipo de servicio.
	2	La aplicación muestra la pantalla de edición del tipo de servicio.
	3	El usuario introduce cambios
	4	El usuario pulsa la opción guardar cambios
	5	La aplicación comprueba que no exista un tipo de servicio con el nuevo nombre
	6	La aplicación guarda los cambios.
	7	La aplicación muestra la lista de servicios.
	8	El caso de uso finaliza con éxito
Postcondición	El tipo de servicio ha sido modificado	
Excepciones	Paso	Acción
	4b	El usuario pulsa la opción volver. El caso de uso no finaliza con éxito.
	6b	La aplicación notifica al usuario de que ya existe una aplicación con el nuevo nombre. Vuelta al paso 2.
Comentarios		

Tabla 52:Especificación del US-12

US-13	Eliminar tipo de servicio	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-06	
Requisitos asociados	RU-20	
Descripción	El usuario podrá eliminar tipos de servicio a los ya existentes.	
Precondición	El usuario está visualizando la lista de servicios.	
Secuencia normal	Paso	Acción
	1	El usuario pulsa la opción borrar tipo de servicio.
	2	La aplicación muestra un diálogo de confirmación.
	3	El usuario pulsa borrar en el diálogo de confirmación.
	4	La aplicación cambia a desconocido el tipo de servicio de los servicios pertenecientes al tipo de servicio borrado.
	5	La aplicación borra el tipo de servicio
	6	La aplicación muestra la lista de servicios.
	7	El caso de uso ha finalizado con éxito.
Postcondición	Se ha eliminado un tipo de servicio.	
Excepciones	Paso	Acción
	3b	El usuario pulsa en la opción cancelar. El caso de uso no finaliza con éxito.
Comentarios		

Tabla 53:Especificación del US-13

US-14	Añadir servicio	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-06	
Requisitos asociados	RU-21	
Descripción	El usuario podrá añadir servicio a los ya existentes.	
Precondición	El usuario está visualizando la lista de servicios. O el usuario selecciona nuevo servicio en la creación de un perfil.	
Secuencia normal	Paso	Acción
	1	El usuario pulsa la opción añadir servicio dentro del tipo de servicio al que pertenece.
	2	La aplicación muestra la pantalla de creación de servicios.
	3	El usuario rellena los campos
	4	El usuario pulsa la opción crear servicio
	5	La aplicación comprueba que no exista un servicio con el mismo nombre
	6	La aplicación crea el nuevo servicio
	7	La aplicación guarda los datos.
	8	La aplicación muestra la lista
	9	El caso de uso finaliza con éxito
Postcondición	Se ha creado un nuevo servicio.	
Excepciones	Paso	Acción
	1b	El usuario pulsa la opción crear perfil teniendo seleccionada la opción nuevo servicio.
	4b	El usuario pulsa la opción volver. El caso de uso finaliza sin éxito.
	6b	La aplicación notifica al usuario de que ese servicio ya existe. vuelve al paso 2.
Comentarios	El tipo de servicio por defecto es en el que se ha añadido, pero se puede modificar.	

Tabla 54:Especificación del US-14

US-15	Editar servicio	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-06	
Requisitos asociados	RU-22	
Descripción	El usuario podrá editar servicios que no sean por defecto.	
Precondición	El usuario está visualizando la lista de servicios.	
Secuencia normal	Paso	Acción
	1	El usuario pulsa la opción editar servicio
	2	La aplicación muestra la pantalla de edición de servicios
	3	El usuario realiza modificaciones
	4	El usuario pulsa la opción guardar cambios.
	5	La aplicación comprueba que no exista otro servicio con el nuevo nombre
	6	La aplicación guarda los cambios
	7	La aplicación muestra la lista de servicios
	8	El caso de uso ha finalizado con éxito.
Postcondición	Se ha creado un nuevo tipo de servicio.	
Excepciones	Paso	Acción
	4b	El usuario pulsa la opción volver. El caso de uso finaliza sin éxito.
	6b	La aplicación notifica al usuario de que ya existe un servicio con el nuevo nombre. Vuelta al paso 2.
Comentarios		

Tabla 55:Especificación del US-15

US-16	Eliminar servicio	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-06	
Requisitos asociados	RU-23	
Descripción	El usuario podrá eliminar servicios que no sean por defecto.	
Precondición	El usuario está visualizando la lista de servicios.	
Secuencia normal	Paso	Acción
	1	El usuario pulsa la opción eliminar servicio
	2	La aplicación muestra una ventana de confirmación.
	3	El usuario pulsa la opción eliminar.
	4	La aplicación adjudica el servicio “sin asignar” a los perfiles que tuvieran este servicio.
	5	La aplicación borra el servicio.
	6	La aplicación muestra la lista de servicios
	7	El caso de uso ha finalizado con éxito.
Postcondición	Se ha eliminado un servicio.	
Excepciones	Paso	Acción
	3b	El usuario pulsa la opción cancelar. El caso de uso finaliza sin éxito.
Comentarios		

Tabla 56:Especificación del US-16

US-17	Visualizar lista de servicios	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-06	
Requisitos asociados	RU-24	
Descripción	Un usuario podrá ver una lista de todos los servicios agrupados por tipo de servicio al que pertenecen.	
Precondición	El usuario identificado y en el menú principal	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la pestaña “Administrar Servicios”
	2	La aplicación muestra la lista de servicios.
	3	El caso de uso ha finalizado con éxito.
Postcondición	Se está visualizando una lista de servicios con opciones para administrar estos servicios.	
Excepciones	Paso	Acción
Comentarios		

Tabla 57:Especificación del US-17

US-18	Elegir caducidad de las contraseñas	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-04	
Requisitos asociados	RU-25	
Descripción	Un usuario podrá cambiar el tiempo de caducidad por defecto de las contraseñas.	
Precondición	El usuario identificado y en la pantalla de ajustes.	
Secuencia normal	Paso	Acción
	1	El usuario modifica el tiempo de caducidad
	2	La aplicación guarda los cambios
	3	El caso de uso a terminado con éxito
Postcondición	Se ha modificado el tiempo de caducidad de las contraseñas.	
Excepciones	Paso	Acción
Comentarios	El tiempo de caducidad afecta a las nuevas contraseñas, ya sean nuevos perfiles o cambios de contraseña, pero no afecta a las contraseñas usadas en ese momento.	

Tabla 58:Especificación del US-18

US-19	Modificar el tiempo de inactividad para desconexión	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-02	
Requisitos asociados	RU-26	
Descripción	Un usuario podrá cambiar el tiempo de inactividad para desconexión por defecto.	
Precondición	El usuario identificado y en la pantalla de ajustes.	
Secuencia normal	Paso	Acción
	1	El usuario realiza un cambio
	2	La aplicación guarda el cambio
	3	El caso de uso ha finalizado con éxito
Postcondición	Se ha modificado el tiempo de inactividad para desconexión por defecto.	
Excepciones	Paso	
Comentarios		

Tabla 59:Especificación del US-19

US-20	Desactivar la desconexión por inactividad	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-04	
Requisitos asociados	RU-27	
Descripción	Un usuario podrá desactivar la desconexión por inactividad	
Precondición	El usuario identificado y en la pantalla de ajustes.	
Secuencia normal	Paso	Acción
	1	El usuario cambia el estado del checkbox
	2	La aplicación guarda el cambio

	3	El caso de uso ha finalizado con éxito
Postcondición	Se ha desactivado la desconexión por inactividad	
Excepciones	Paso	
Comentarios	Se puede volver a activar siguiendo el mismo proceso.	

Tabla 60:Especificación del US-20

US-21	Borrar usuario	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-01,OBJ-02,OBJ-05	
Requisitos asociados	RU-28	
Descripción	Un usuario podrá borrar todos sus datos	
Precondición	El usuario identificado y en la pantalla de ajustes.	
Secuencia normal	Paso	Acción
	1	El usuario pulsa la opción borrar usuario
	2	El sistema muestra una pantalla de confirmación
	3	El usuario pulsa “si”
	4	La aplicación borra los datos
	5	La aplicación muestra el menú de inicio
	6	El caso de uso finaliza con éxito
Postcondición	Se han borrado los datos del usuario	
Excepciones	Paso	Acción
	3b	El usuario pulsa “no”. El caso de uso finaliza sin éxito.
Comentarios		

Tabla 61:Especificación del US-21

US-22	Visualizar ajustes		
Versión	1.0		
Autor	Hugo Gilarranz		
Objetivos asociados	OBJ-04		
Requisitos asociados	RU-25		
Descripción	Un usuario podrá ver los ajustes actuales		
Precondición	El usuario identificado y en el menú principal		
Secuencia normal	Paso	Acción	
	1	El usuario pulsa en botón ajustes	
	2	La aplicación muestra la pantalla de ajustes	
	3	El caso de uso ha finalizado con éxito	
Postcondición	Se están visualizando los ajustes actuales		
Excepciones	Paso	Acción	
Comentarios			

Tabla 62:Especificación del US-22

5.1.6 Requisitos de información

ENT- 01	Configuración de la aplicación	Versión	2.0
Definición	Elemento aglutinador de los ajustes de la aplicación así como datos fundamentales para su funcionamiento.		
Consideraciones			

ATRIBUTOS						
ID	Nombre	Descripción	Dominio	UNIQUE	NULL	Notas
	Contraseña	128 primeros bits de la contraseña pasada por el algoritmo SHA-384	VARCHAR (128)	SÍ	NO	
	Tiempo de expiración	Tiempo por defecto para la caducidad de las	INT(3)	NO	NO	

		contraseñas en días				
	Tiempo de desconexión	Tiempo para la desconexión	int(2)	No	No	
	Desconexión	Si se realiza la conexión por inactividad o no	Boolean	No	No	
	Última actividad	Tiempo desde la última vez que el usuario lanzo un evento.	int(5)	No	No	

Tabla 63:Especificación de la ENT-01

ENT- 02	Perfil de Servicios		Versión	2.0
Definición	Perfil que representa una cuenta gestionada por la aplicación en el que se almacenan los datos de acceso así como la información fundamental para describir dicho servicio.			
Consideraciones				
ATRIBUTOS				
ID	Nombre	Descripción	Dominio	UNIQUE	NULL	Notas
	Email	Email al que está vinculado este perfil	VARCHAR (128)	SÍ	NO	
	Nombre de usuario	Nombre de usuario	VARCHAR (128)	No	Si	
	Contraseña	Contraseña de acceso a ese servicio	VARCHAR (128)	Si	No	
	Expiración contraseña	Fecha de expiración de la contraseña	DATE	NO	NO	
	Servicio	Nombre descriptivo del servicio pj:Gmail	Enum	No	No	

Tabla 64:Especificación de la ENT-02

ENT- 04	Tipo de Servicio	Versión	1.0
Definición	Perfil que representa una cuenta gestionada por la aplicación en el que se almacenan los datos de acceso así como la información fundamental para describir dicho servicio.		
Consideraciones			

ATRIBUTOS						
ID	Nombre	Descripción	Dominio	UNIQUE	NULL	Notas
	Nombre	Nombre identificador de este tipo de servicios	VARCHAR (128)	SÍ	NO	
	Color	Color que marca a los Servicios perteneciente s al tipo		No	No	

Tabla 65:Especificación de la ENT-04

ENT- 05	Servicios	Versión	1.0
Definición	Perfil que representa una cuenta gestionada por la aplicación en el que se almacenan los datos de acceso así como la información fundamental para describir dicho servicio.		
Consideraciones			

ATRIBUTOS						
ID	Nombre	Descripción	Dominio	UNIQUE	NULL	Notas
	Nombre	Nombre del Servicio	VARCHAR (128)	SÍ	NO	
	Url	Url que conecta con el sitio web del servicio (si existe)	VARCHAR (128)	Si	Si	
	Icono	Icono fácilmente reconocible que representa al tipo en las listas		No	Si	

	Tipo de Servicio	Tipo de Servicio que engloba a este Servicio	Enum	No	No	

Tabla 66:Especificación de la ENT-05

5.1.7 Requisitos no funcionales

Requisitos no funcionales de Seguridad:

~~NFS-05 Se debe bloquear el uso trás un periodo de inactividad de 15 min.~~

NFS-06 Se debe poder activar/desactivar el bloqueo por inactividad

MFS-07 Se debe poder elegir el periodo para el bloqueo por inactividad

Requisitos no funcionales de usabilidad:

NFU-03 Se debe diseñar una interfaz sencilla, con la usabilidad como prioridad.

5.2 Diseño

5.2.1 Arquitectura lógica

En esta iteración se ha definido los componentes pertenecientes a la Vista del MVC que habíamos definido en la iteración anterior. El diagrama ya completo de la arquitectura lógica es el siguiente:

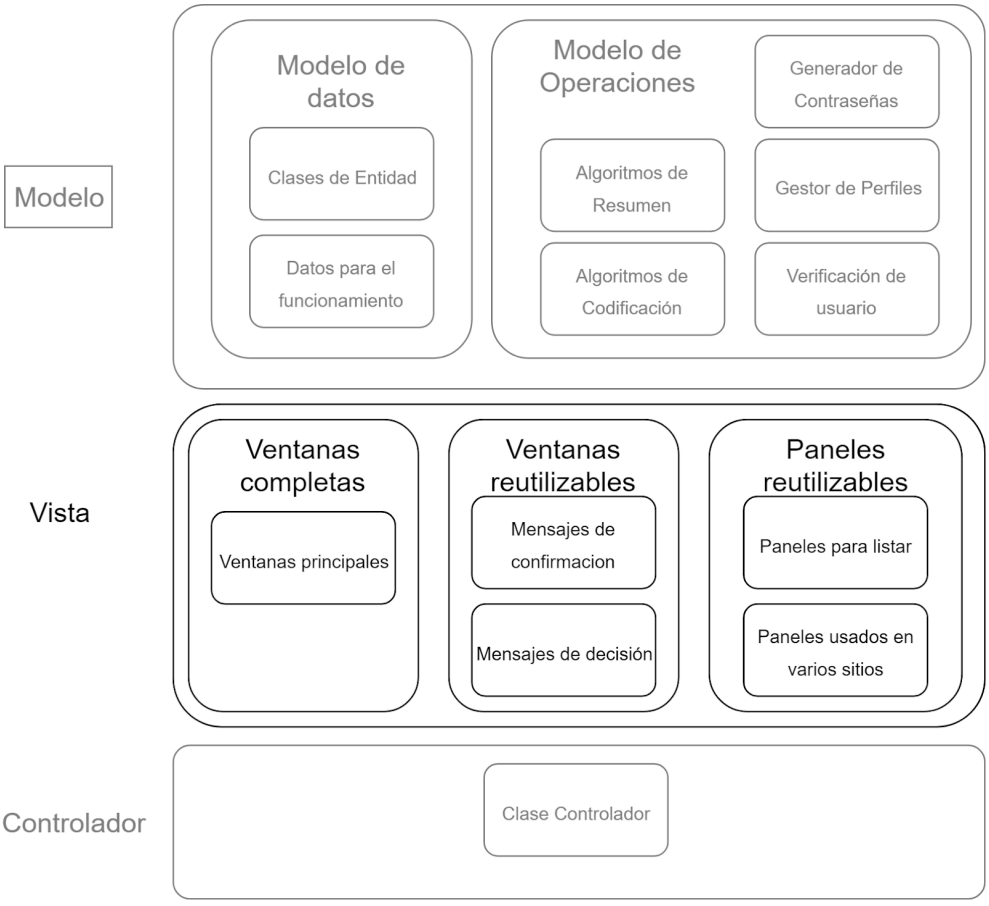


Figura 36:Especificación de la definición de la arquitectura lógica, segunda iteración

5.2.2 Arquitectura física

La arquitectura física de esta aplicación no ha sufrido cambios con respecto a lo expuesto en la iteración anterior.

5.2.3 Diagrama de clases

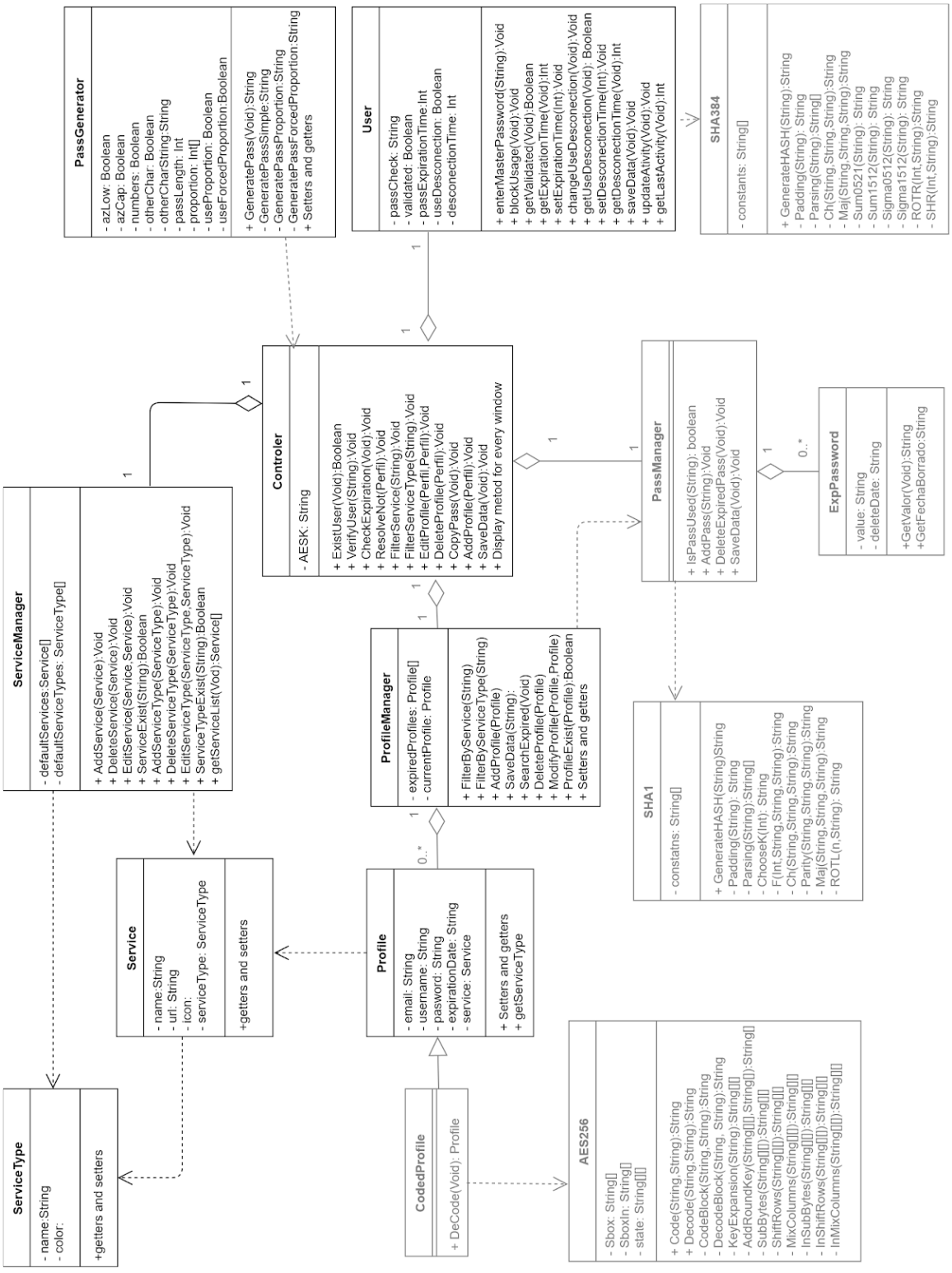


Figura 37:Diagrama de clases de la segunda iteración

5.2.4 Interfaz gráfica

A continuación se detalla el diseño de las vistas que componen la interfaz gráfica de la aplicación.

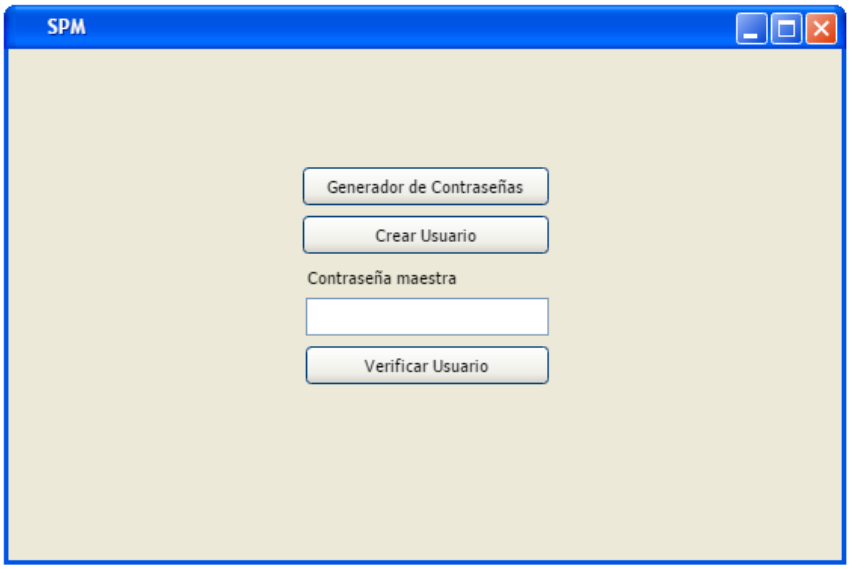
Menú de Inicio	
Descripción	Este diseño muestra la vista que aparece al iniciar la aplicación
Activación	Esta vista aparece al iniciar la aplicación, o tras producirse la desconexión, ya sea manual o automática, del usuario.
Boceto	
Eventos	Botón “Generador de Contraseñas”: este botón redirige a la ventana del generador de contraseñas. Botón“Crear Usuario”: este botón redirige a la ventana de creación de usuario Botón “Verificar Usuario”: este botón procede a verificar al usuario con el contenido del campo de texto ofuscado “Contraseña maestra”.

Tabla 67:Diseño de la ventana Menú de inicio

Generador de Contraseñas	
Descripción	Esta ventana muestra el generador de contraseñas, permitiendo modificar su configuración y generar contraseñas.
Activación	Esta ventana se muestra desde el menú de inicio con el botón “Generador de contraseñas” También se puede acceder mediante los paneles de creación y edición, en cuyo caso el botón “copiar” tendrá como texto “insertar” y un evento distinto asignado.

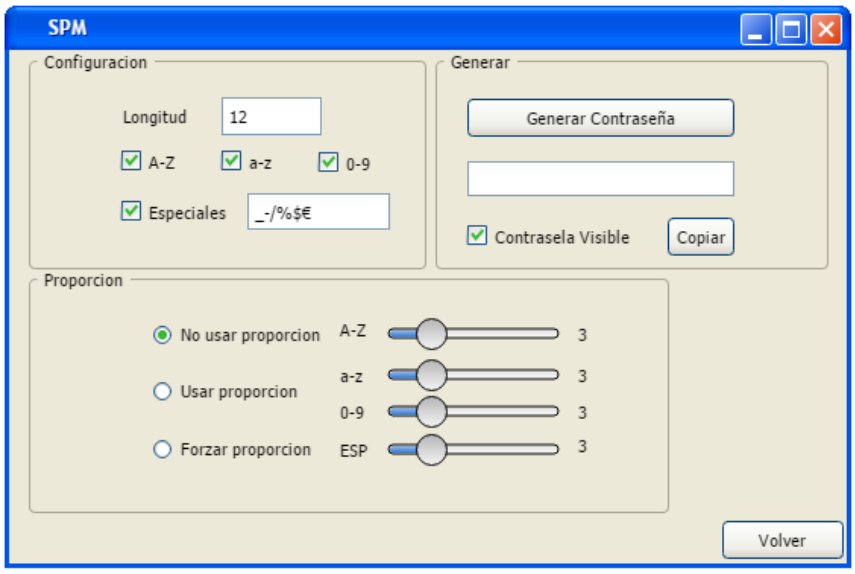
Boceto	
Eventos	-El campo de texto Longitud: este campo muestra el valor actual de la longitud para la contraseña generada y permite modificarlo. - Checkbox “A-Z”: este checkbox muestra si se utilizan o no mayúsculas al generar contraseñas, y permite modificar su valor. - Checkbox “a-z”: este checkbox muestra si se utilizan o no minusculas al generar contraseñas, y permite modificar su valor. - Checkbox “0-9”: este checkbox muestra si se utilizan o no números al generar contraseñas, y permite modificar su valor. - Checkbox “Especiales”: este checkbox muestra si se utilizan o no caracteres especiales al generar contraseñas, y permite modificar su valor. -El campo de texto “Especiales”: este campo muestra los caracteres especiales a utilizar y permite modificarlos. -Los botones “No usar proporción”, “Usar proporción” y Forzar proporción”: sólo uno de los tres puede estar activado, indican si se utiliza o no una proporción y en caso de usarse como se usa. -Los sliders “A-Z”, “a-z”, “0-9” y “Esp”: Cada uno de los slider muestra la proporción asignada a cada tipo de carácter, cuando se modifica uno se ajustan el resto para que la suma sea igual a la longitud. -El botón “Generar contraseña”: genera una contraseña en el campo de texto situado por debajo. -El checkbox “Contraseña visible”: alterna el campo que muestra las contraseñas generadas entre ofuscado y visible. -El botón “copiar”: copia la contraseña generada al portapapeles. -El botón “insertar”: regresa a la pantalla desde la que ha sido llamado insertando la contraseña generada en el campo contraseña y en el campo confirmar contraseña de la pantalla que lo ha llamado. -El botón “volver”: regresa a la pantalla desde la que ha sido llamado si realizar ninguna acción.

Tabla 68:Diseño de la ventana Generador de contraseñas

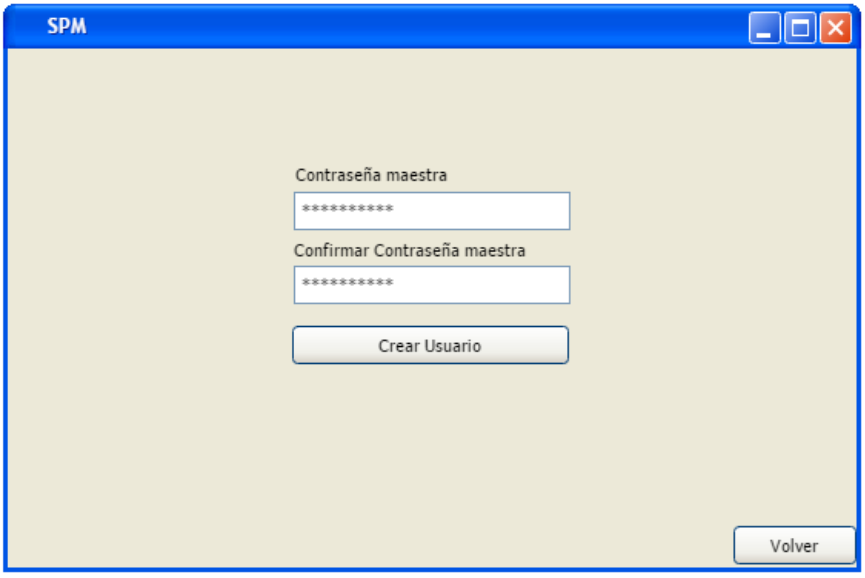
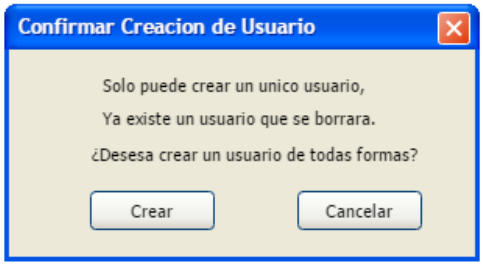
Crear usuario	
Descripción	Esta pantalla permite introducir la contraseña maestra por primera vez
Activación	Utilizar el botón “Crear usuario” desde el menú de inicio.
Boceto	
Eventos	-El botón “Crear usuario”: si las contraseñas introducidas en los campos “Contraseña maestra” y “Confirmar contraseña maestra” coinciden y no existe un usuario, se crea un usuario con esa contraseña maestra. Si existe un usuario se llama a la pantalla “Confirmar creación de usuario”. Si las contraseñas no coinciden o no están introducidas se muestra un mensaje indicando el problema. -El botón “Volver”: vuelve a la pantalla “Menú de inicio”.

Tabla 69:Diseño de la ventana Crear usuario

Confirmar creación de usuario	
Descripción	Ventana para confirmar la creación de un usuario nuevo sobrescribiendo al actual.
Activación	Crear usuario desde la ventana de creación de usuarios cuando ya exist un usuario.
Boceto	

Eventos	-El botón “Crear”: procederá a borrar al usuario actual y crear uno nuevo con la contraseña maestra introducida en la ventana anterior. Al finalizar la creación se muestra el menú principal con el nuevo usuario verificado. -El botón “Cancelar”: volverá a la pantalla de creación de usuario sin crear a un nuevo usuario.

Tabla 70:Diseño de la ventana Confirmar creación de usuario

Menú principal (Ver perfiles)	
Descripción	Ventana principal de la aplicación, por defecto muestra la pestaña de “Ver perfiles”. Los botones situados en la esquina superior derecha son compartidos por todas las pestañas. Y desde todas las pestañas se puede cambiar a todas las pestañas.
Activación	Verificar usuario en la aplicación. Terminar alguna operación que redirija a la pestaña de “Ver perfiles”
Boceto	
Eventos	Generales del menú principal: -Pestañas superiores: permiten cambiar a los paneles con dicho nombre en cualquier momento. -El botón “Ajustes”: redirige a la ventana de visualización de ajustes. -El botón “Bloquear uso”:muestra el popup “Bloquear uso”. Específicas del panel Ver perfiles: -”Seleccionar tipo de filtro” dropdown”: elige el tipo de filtro que usará el botón filtrar para filtrar la lista de perfiles entre “servicio”, “tipo de servicio” y “sin filtrar”. -”Seleccionar filtro” dropdown”: elige de la lista de servicios/tipos de servicios (según lo seleccionado en el dropdown “Seleccionar tipo de filtro” -El botón “Filtrar”: filtra la lista de perfiles en función de los parámetros seleccionados en los dropdowns contiguos. -El botón “Añadir perfil”: redirige a la ventana de creación de perfiles. -Los botones “Ver perfil”: redirigen a la ventana de ver perfil que muestra el perfil con el que están relacionados (el situado a su izquierda). -Los botones “Eliminar”: muestra el popup “Eliminar perfil”.

Tabla 71:Diseño de la ventana Menú principal (Ver perfiles)

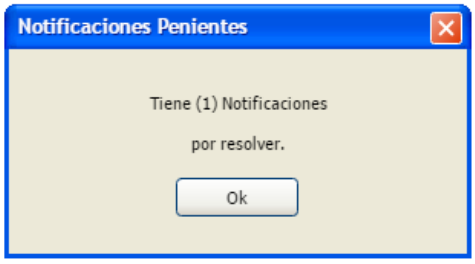
Notificaciones Pendientes	
Descripción	Este popup aparece al verificar usuario antes de mostrar el menú principal si existen notificaciones pendientes.
Activación	Al verificar usuario con notificaciones pendientes.
Boceto	
Eventos	-El botón “Ok”: cierra el popup y permite el acceso completo a menú principal.

Tabla 72:Diseño de la ventana Notificaciones pendientes

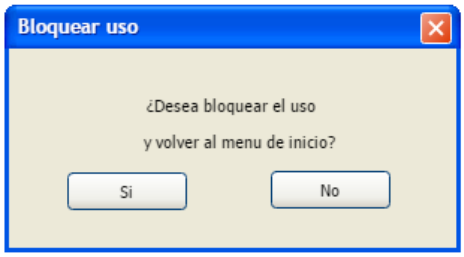
Bloquear uso	
Descripción	Ventana de confirmación para bloquear el uso. El menú principal se muestra por debajo pero su uso está bloqueado.
Activación	Pulsar el botón bloquear uso desde cualquier panel del menú principal.
Boceto	
Eventos	-El botón “Si”: procede al bloqueo de uso llendo al menú de inicio y haciendo necesario volver a introducir la contraseña para acceder a los datos. -El botón “No”: cancela el bloqueo y vuelve a permitir usar el menú principal.

Tabla 73:Diseño de la ventana Bloquear uso

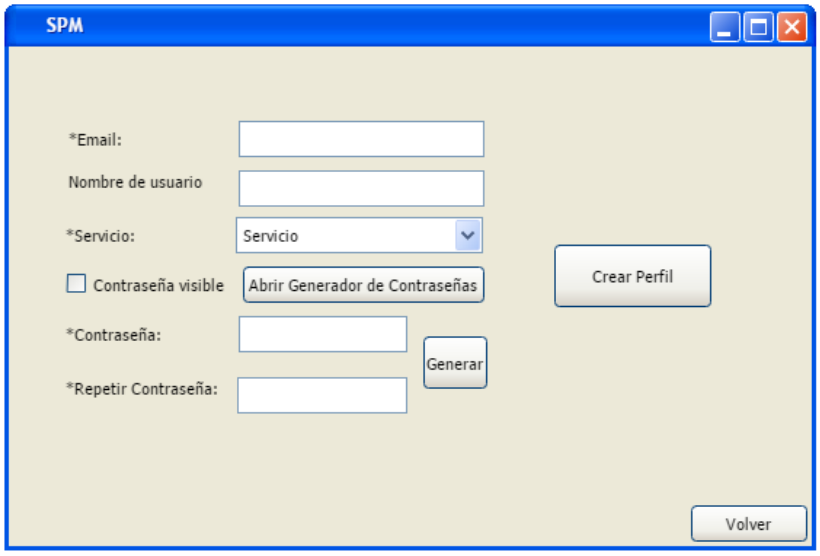
Añadir perfil	
Descripción	Ventana para la creación de nuevos perfiles.
Activación	Pulsar el botón “Añadir perfil” en el panel de Ver perfiles del menú principal.
Boceto	
Eventos	-El campo de texto“*Email”: recoge el email para el nuevo perfil. Es un campo obligatorio. -El campo de texto “Nombre de usuario”: recoge el nombre de usuario para el nuevo perfil. Si el campo no se rellena se usa el email como nombre de usuario. -El dropdown “Servicio”: permite elegir entre los servicios existentes o uno nuevo para el nuevo perfil. -El checkbox “Contraseña visible: permite alternar entre enmascarado y visible para los campos contraseña y repetir Contraseña -El campo “Contraseña”: recoge la contraseña para el nuevo perfil. -El campo “Repetir contraseña”: recoge la contraseña para el nuevo perfil como comprobante. -El botón “Abrir Generador de contraseñas”: abre el generador de contraseñas con el botón “Iniciar” en lugar de “Copiar”. -El botón “Generar”: genera una contraseña con la configuración actual del generador de contraseñas. -El botón “Crear perfil”: si no existe un perfil del tipo seleccionado con ese email,las dos contraseñas coinciden y la contraseña no está en uso, crea un nuevo perfil con los datos introducidos. Si falla algo notifica al usuario del fallo. -El botón “Volver”: regresa al panel “Ver perfiles” del menú principal sin añadir un perfil.

Tabla 74:Diseño de la ventana Añadir perfil

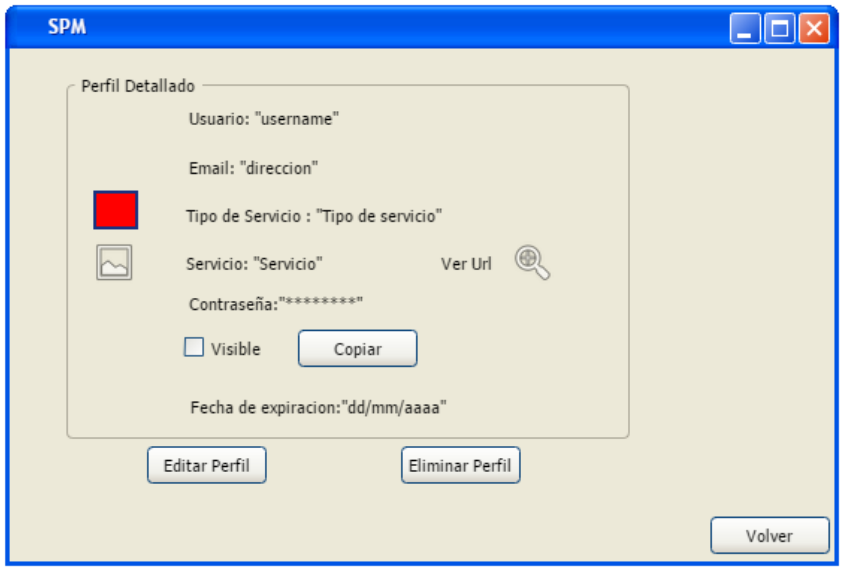
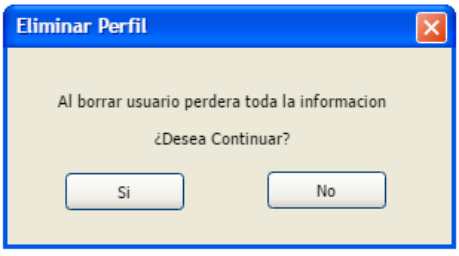
Ver perfil	
Descripción	Muestra la información completa de un perfil.
Activación	Pulsar el botón “Ver perfil” del perfil visualizado en el panel de “Ver perfiles” del menú principal.
Boceto	
Eventos	-El checkbox “Visible”: cambia la visibilidad de la contraseña entre ofuscada y visible. -El botón “Ver Url”: abre un pequeño panel con la url (hacer alguna acción fuera del panel lo oculta). -El botón “Copiar”: copia la contraseña al portapapeles. -El botón “Editar Perfil”: abre la pantalla de edición de perfiles. -El botón “Eliminar perfil”: abre el popup de confirmación “Eliminar perfil”. -El botón “Volver”:vuelve al panel de “Ver perfiles” del menú principal.

Tabla 75:Diseño de la ventana Ver perfil

Eliminar perfil	
Descripción	Popup para confirmar la eliminación de un perfil.
Activación	Pulsar el botón “Eliminar perfil” en la ventana “Ver perfil”.
Boceto	

Eventos	-El botón “Sí”: confirma el borrado del perfil, borrandolo y mostrando la lista de perfiles en el panel “Ver perfiles” le menu principal” -El botón “No”: cancela el borrado del perfil, regresa a la vista “Ver perfil”.

Tabla 76:Diseño de la ventana Eliminar perfil

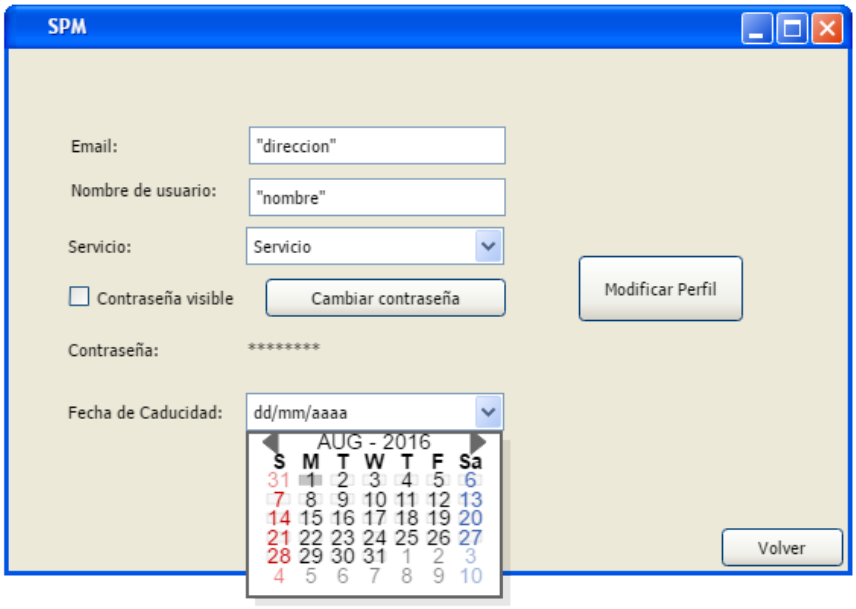
Editar perfil	
Descripción	Pantalla para la modificación de un perfil.
Activación	Pulsar al botón “Editar perfil” de la pantalla “Ver perfil”.
Boceto	
Eventos	-El campo de texto “Email”: muestra la dirección de correo actual y permite modificarla. -El campo de texto “Nombre de usuario”: muestra el nombre de usuario actual y permite modificarlo. -El dropdown “Servicio”: muestra el servicio actual y permite modificarlo seleccionando entre los servicios existentes o uno nuevo. -El checkbox “Contraseña visible”: alterna la visibilidad de la contraseña entre ofuscada y visible. -E botón “Cambiar contraseña”: abre la pantalla de modificación de contraseña. -El datePicker “Fecha de caducidad”: muestra la fecha de expiración actual de la contraseña y permite modificarla. El botón “Modificar perfil”: guarda las modificaciones realizadas de haberse realizado alguna y después regresa a la ventana “Ver perfil” -El botón “Volver”: cancela la edición sin hacer cambios y regresa a la ventana “Ver perfil”

Tabla 77:Diseño de la ventana Editar perfil

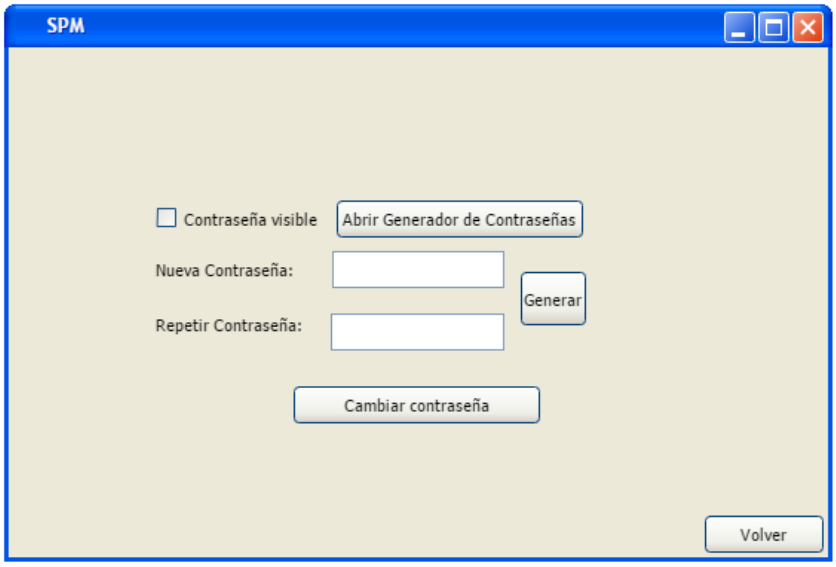
Cambiar contraseña	
Descripción	Ventana para el cambio de contraseña.
Activación	Pulsar el botón “Cambiar contraseña” de la ventana de edición de un perfil. Pulsar el botón “Resolver notificación” de una notificación de contraseña expirada.
Boceto	
Eventos	-El checkbox “Contraseña visible”: alterna entra ofuscado y visible para los campos nueva contraseña y repetir contraseña. -El campo “Contraseña”: recoge la contraseña para el nuevo perfil. -El campo “Repetir contraseña”: recoge la contraseña para el nuevo perfil como comprobante. -El botón “Abrir Generador de contraseñas”: abre el generador de contraseñas con el botón “Iniciar” en lugar de “Copiar”. -El botón “Generar”: genera una contraseña con la configuración actual del generador de contraseñas. -El botón “Cambiar contraseña”: si las contraseñas coinciden y no están en uso realiza el cambio de contraseña (para el caso de editar un perfil los datos no se guardan hasta que no se confirma en la pantalla de edición). Y regresa a la pantalla que realizó la llamada. -El botón “Volver”: regresa a la pantalla que ha llamado a la ventana actual sin realizar la

Tabla 78:Diseño de la ventana Cambiar contraseña

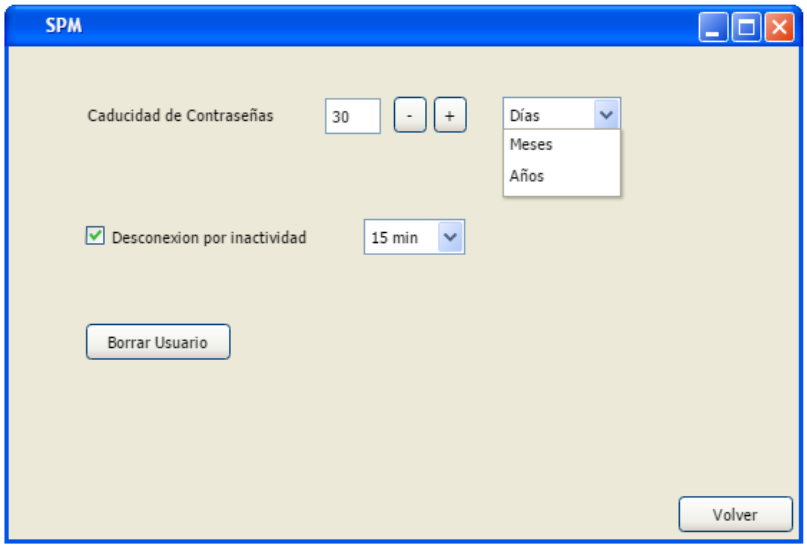
Ajustes	
Descripción	Pantalla que permite visualizar y modificar los ajustes de la aplicación
Activación	Desde el menú principal usar el botón “Ajustes” en la esquina superior derecha.
Boceto	
Eventos	-El campo de texto “Caducidad de contraseña”:Permite visualizar y modificar las cantidad del tiempo de caducidad por defecto. -Los botones “+” y “-”: permiten aumentar o disminuir en una unidad el valor del campo “Caducidad de contraseña”. -El dropdown “Días”: permite seleccionar la unidad para la cantidad de tiempo de caducidad por defecto. -El checkbox “Desconexión por inactividad”: permite activar/desactivar la Desconexión por inactividad. -El dropdown”Desconexión por inactividad”:permite elegir el tiempo de espera para la desconexión por inactividad entre unos valores predeterminados. -El botón “Borrar usuario”: abre el popup de confirmación para la eliminación del usuario -El botón “Volver”: regresa al menú principal.

Tabla 79:Diseño de la ventana Ajustes

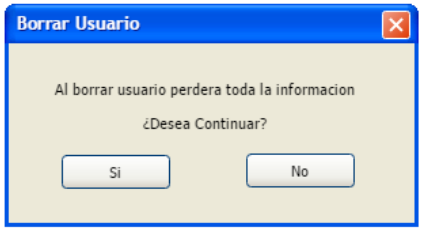
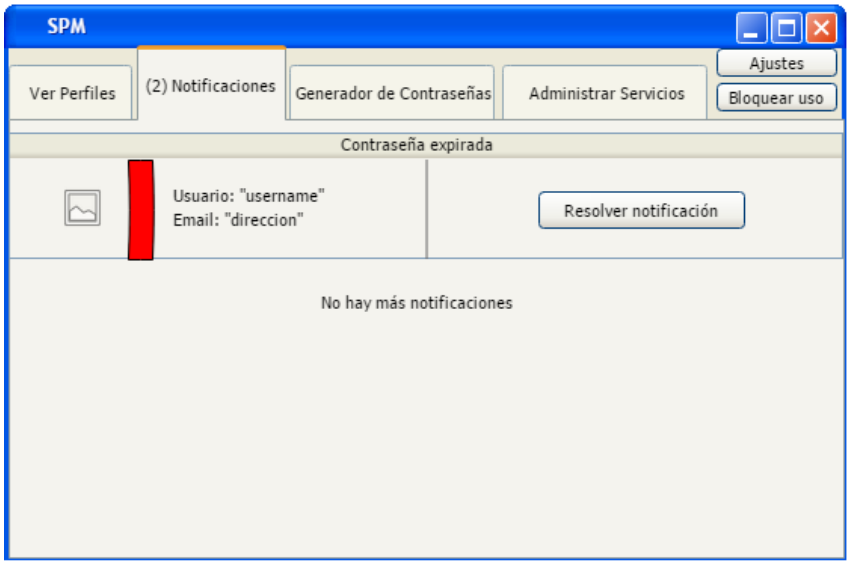
Borrar Usuario	
Descripción	Popup de confirmación para borrar al usuario
Activación	Pulsar el botón “Borrar usuario” en la ventana de ajustes.
Boceto	
Eventos	-El botón “Sí”: Confirma el borrado del usuario, la aplicación borra los datos del usuario y muestra el menú de inicio. -El botón “No”: Cancela el borrado del usuario, devuelve el control a la ventana de ajustes.

Tabla 80:Diseño de la ventana Borrar Usuario

Menú principal (notificaciones)	
Descripción	Panel del menú principal que muestra la notificaciones pendientes por resolver.
Activación	Desde cualquier otra parte del menú principal hacer click en la pestaña “Notificaciones”
Boceto	

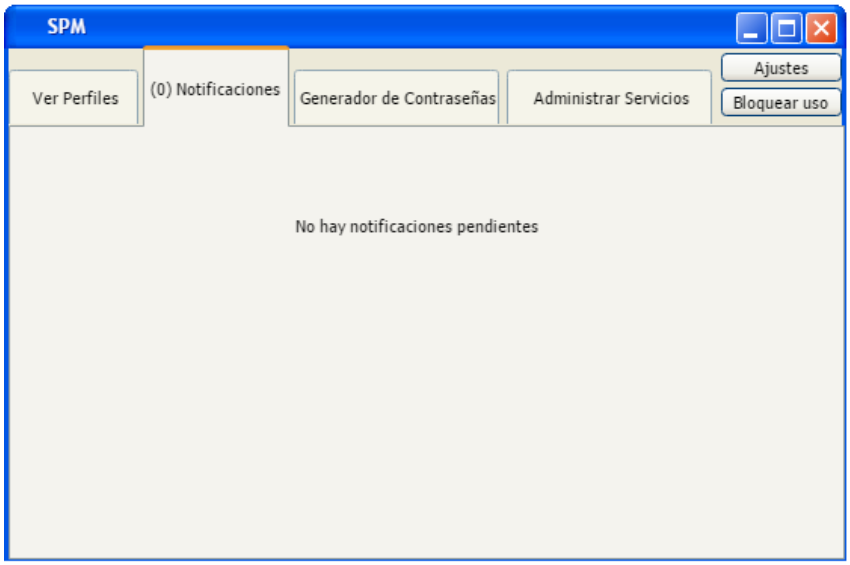
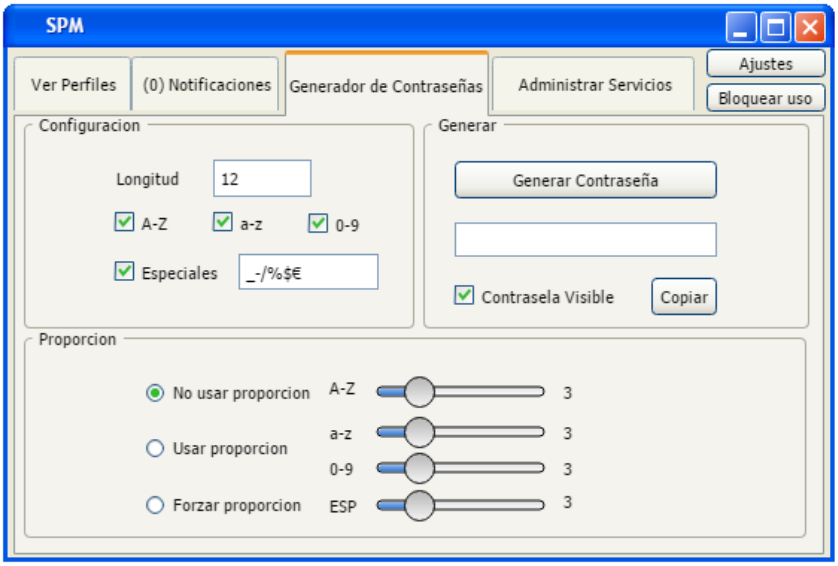
	
Eventos	Ver “Menú principal (Ver perfiles)” para los elementos comunes del menú principal. -El botón “Resolver Notificación”: muestra la ventana de cambio de contraseña para el perfil relacionado.

Tabla 81:Diseño de la ventana Menú principal (notificaciones)

Menú principal (generador de contraseñas)	
Descripción	Panel del menú principal que permite el uso del generador de contraseñas.
Activación	Desde cualquier parte del menú principal hacer click en la pestaña “Generador de Contraseñas”
Boceto	

Eventos	Ver “Menú principal (Ver perfiles)” para los elementos comunes del menú principal. Ver “Generador de Contraseñas” para los elementos propios del panel.

Tabla 82:Diseño de la ventana Menú principal(generador de contraseñas)

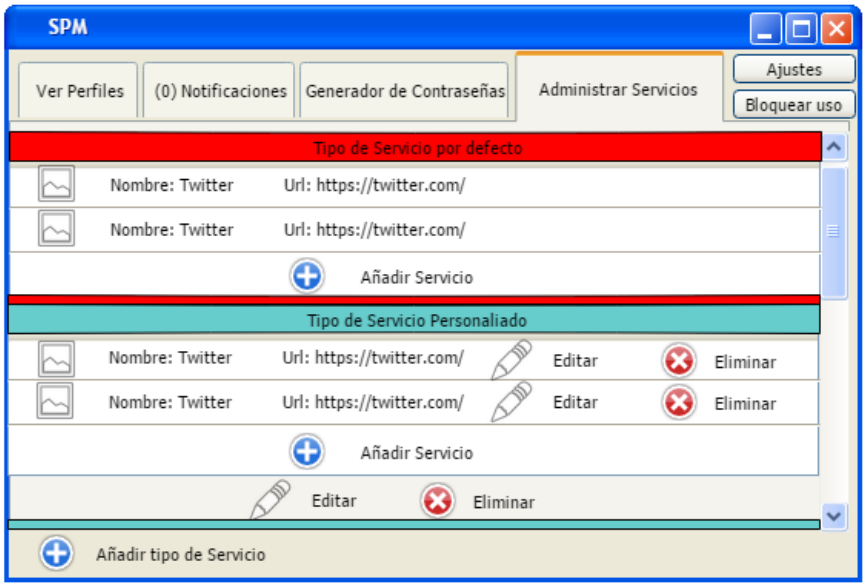
Menú principal (Administrar servicios)	
Descripción	Panel del administrador de servicios dentro del menú principal. Permite visualizar los servicios organizados por el tipo de servicio al que pertenecen.
Activación	Desde cualquier punto del menú principal haciendo click en la pestaña “Administrar servicios”.
Boceto	
Eventos	Ver “Menú principal (Ver perfiles)” para los elementos comunes del menú principal. -El botón “Añadir servicio”: abre la ventana de creación de servicios. -El botón “Editar”(a la derecha de un servicio): abre la ventana de edición del servicio. -El botón “Eliminar”(a la derecha de un servicio): abre el popup de confirmación para eliminar ese servicio. -El botón “Editar”(Debajo de añadir Servicios dentro de la sección de un tipo de servicio que no sea por defecto): abre la ventana de edición del tipo de servicio. -El botón “Eliminar”(Debajo de añadir Servicios dentro de la sección de un tipo de servicio que no sea por defecto): abre el popup de confirmación para eliminar ese tipo de servicio. -El botón “Añadir tipo de Servicio”: abre la ventana de creación de tipos de servicios.

Tabla 83:Diseño de la ventana Menú principal (administrar servicios)

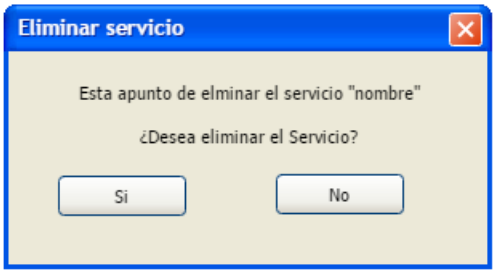
Eliminar servicio	
Descripción	Popup de confirmación para eliminar un servicio.
Activación	Pulsar al botón “Eliminar” del servicio en la ventana “Administrar servicios”
Boceto	
Eventos	-El botón “Sí”: Confirma la eliminación del servicio y devuelve el control al panel de “Administrar Servicios” del menú principal. -El botón “No”: Cancela la eliminación del servicio y devuelve el control al panel de “Administrar Servicios” del menú principal.

Tabla 84:Diseño de la ventana Eliminar servicio

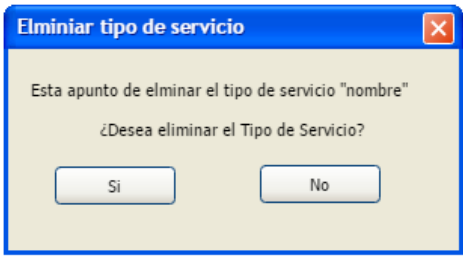
Eliminar tipo de servicio	
Descripción	Popup de confirmación para eliminar un tipo servicio.
Activación	Pulsar al botón “Eliminar” del tipo de servicio en la ventana “Administrar servicios”
Boceto	
Eventos	-El botón “Sí”: Confirma la eliminación del tipo de servicio y devuelve el control al panel de “Administrar Servicios” del menú principal. -El botón “No”: Cancela la eliminación del tipo de servicio y devuelve el control al panel de “Administrar Servicios” del menú principal.

Tabla 85:Diseño de la ventana Eliminar tipo de servicio

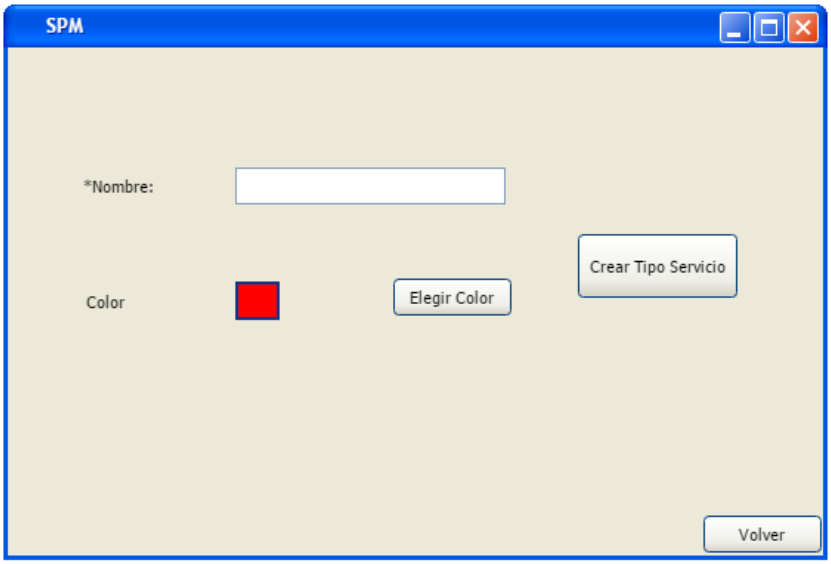
Crear tipo de servicio	
Descripción	Ventana para la creación de nuevos tipos de servicio
Activación	Pulsar el botón “Añadir tipo de servicio” en el panel “Administrar Servicios” del menú principal. Elegir nuevo tipo de servicio, como el tipo de servicio para un nuevo servicio. elegir nuevo tipo de servicio, como el tipo de servicio para modificar un servicio.
Boceto	
Eventos	-Campo de texto “Nombre”: recoge el nombre del tipo de servicio. -El botón “Elegir Color”: abre un color picker para escoger el color asociado a el nuevo tipo de servicio. -El botón “Crear tipo de servicio”: Crea un nuevo tipo de servicio con los datos introducidos si no existe ya uno con el mismo nombre y muestra el panel “Administrar Servicios” del menú principal”. En caso de que ya exista otro tipo de servicio con ese nombre se lo notifica al usuario. -El botón “Volver”: muestra el panel “Administrar Servicios” del menú principal cancelando la adición de un nuevo tipo de servicio.

Tabla 86:Diseño de la ventana Crear tipo de servicio

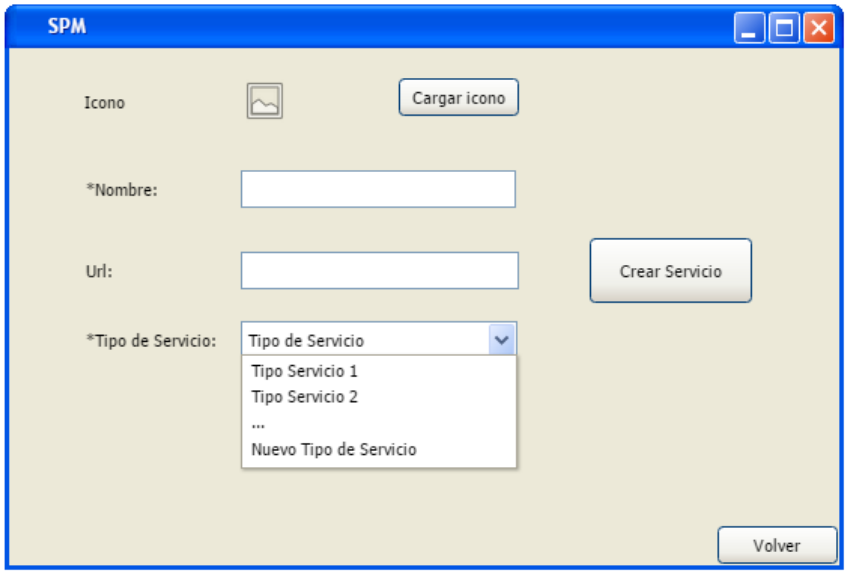
Crear servicio	
Descripción	Ventana para la creación de nuevos servicios
Activación	Pulsar el botón “Añadir servicio” dentro de un tipo de servicio en el panel “Administrar servicios” en el menú principal. Seleccionar nuevo servicio como el servicio para un nuevo perfil. Seleccionar nuevo servicio como el servicio para la modificación de un perfil.
Boceto	
Eventos	-El botón “Cargar icono”: permite examinar el equipo para elegir un icono para el nuevo servicio. -Campo de texto “Nombre”: permite introducir el nombre para el nuevo servicio. -Campo de texto “Url”: permite introducir la url para el nuevo servicio. -Dropdown “Tipo de Servicio”: permite escoger el tipo de servicio para el nuevo servicio entre la lista de existentes o crear uno nuevo. -El botón “Crear servicio”: Si no existe otro servicio con el mismo nombre crea un nuevo servicio con los datos introducidos y muestra el panel “Administrar Servicios” del menú principal. Si existe otro se lo notifica al usuario. -El botón “Volver”: cancela la adición de un nuevo servicio y muestra el panel “Administrar Servicios” del menú principal.

Tabla 87:Diseño de la ventana Crear servicio

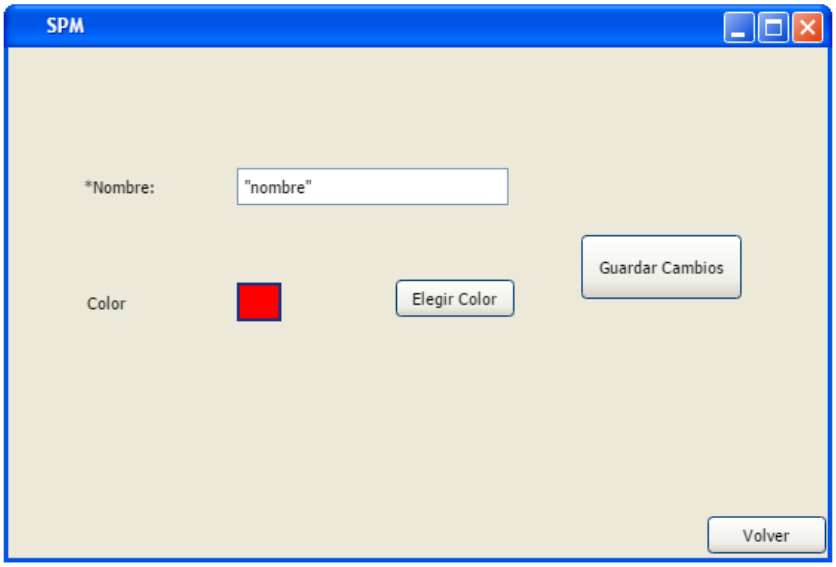
Editar tipo de servicio	
Descripción	Ventana que permite la edición de un tipo de servicio.
Activación	Pulsar el botón editar tipo de servicio de un tipo de servicio que no sea por defecto desde el panel “Administrar servicios” del menú principal.
Boceto	
Eventos	-Campo de texto “Nombre”: muestra y permite modificar el nombre del tipo de servicio. -El botón “Elegir Color”: abre un color picker para elegir el color asignado al tipo de servicio. -El botón “Guardar Cambios”: si el nuevo nombre no coincide con un servicio ya existente guarda los cambios en el tipo de servicio y muestra el panel “Administrar servicios” del menú principal. Si coincide se lo notifica al usuario. -El botón “Volver”: cancela la edición del tipo de servicio y muestra el panel “Administrar servicios” del menú principal.

Tabla 88:Diseño de la ventana Editar tipo de servicio

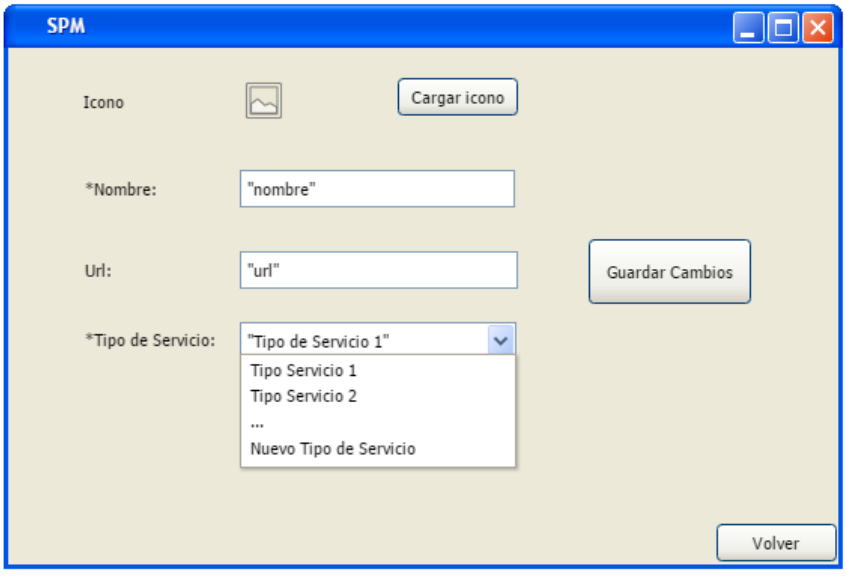
Editar servicio	
Descripción	Ventana que permite la edición de un servicio
Activación	Pulsar el botón “Editar” de un servicio en el panel “Administrar Servicios” del menú principal.
Boceto	
Eventos	-El botón “Cargar icono”: permite examinar el equipo para buscar un nuevo icono para el servicio. -El campo de texto “Nombre”: muestra y permite modificar el nombre del servicio que está siendo editado. -El campo de texto “url”: muestra y permite modificar la url del servicio que está siendo editado. -El dropdown “Tipo de servicio”: muestra el tipo de servicio del servicio que está siendo editado y permite modificarlo por uno de la lista de tipos existentes o crear uno nuevo. -El botón “Guardar Cambios”: si el nuevo nombre no coincide con el de uno de los servicios ya existentes guarda los cambios en el servicio y muestra el panel “Administrar servicios” del menú principal. Si coincide notifica al usuario. -El botón “Volver”: cancela la edición del servicio y muestra el panel “Administrar servicios” del menú principal.

Tabla 89:Diseño de la ventana Editar servicio

A modo de resumen y para comprender con mayor facilidad la relación entre las vistas a continuación se mostraran unos diagramas con la relación entre las vistas.

El primero muestra un resumen general de la aplicación:

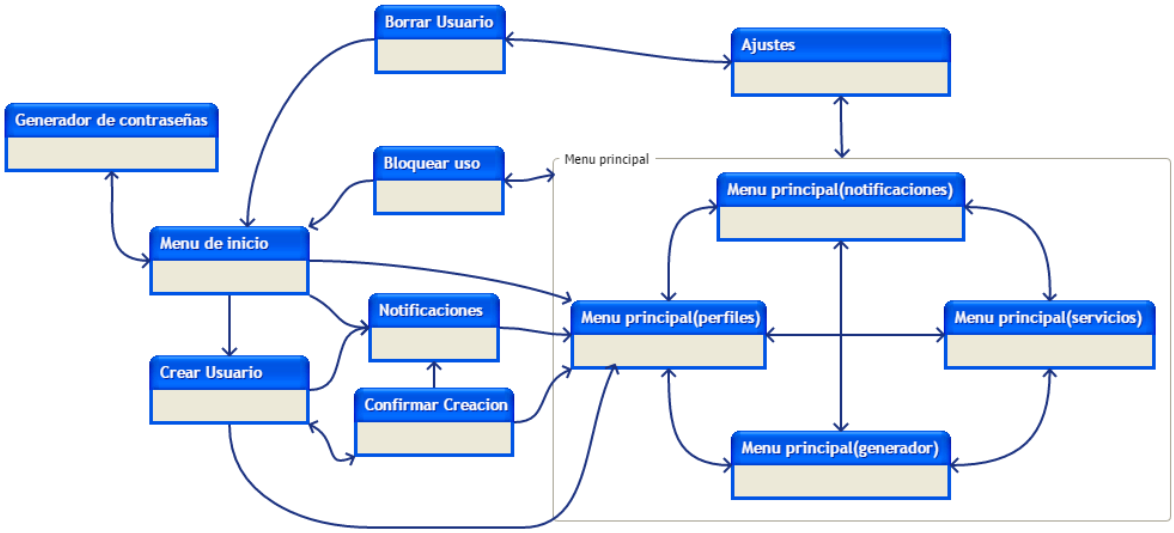


Figura 38: Resumen de relación de vistas global

El siguiente muestra la relación de la pestaña de perfiles del menú principal con las vistas que dependen de él:

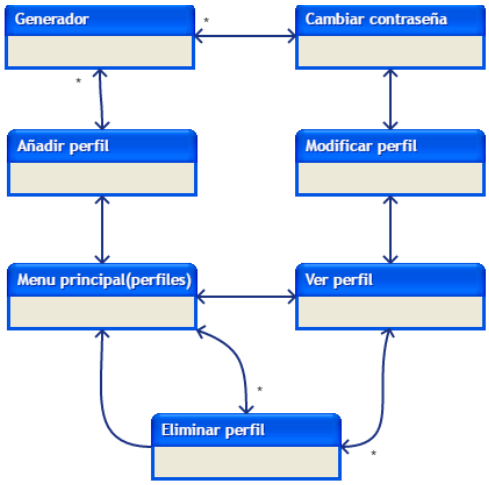


Figura 39: Resumen de relación de vistas desde Ver perfiles

El siguiente muestra las relaciones del panel administrar servicios del menú principal (las flechas con "*" significan que se utiliza la misma vista pero crea entidades distintas para cada operación):

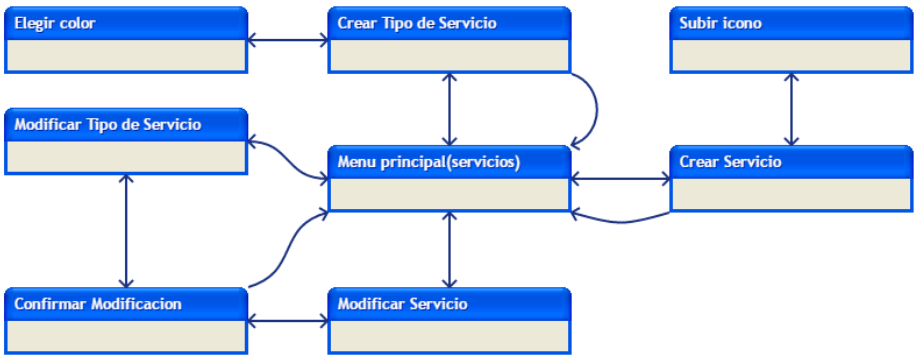


Figura 40: Resumen de relación de vistas desde Administrar servicios

Por último con las pantallas dependientes del panel Notificaciones del menú principal:

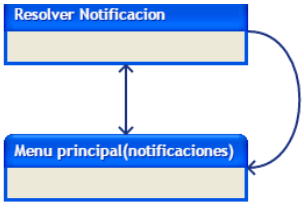


Figura 41: Resumen de relación de vistas desde Notificaciones

5.3 Implementación

La interfaz gráfica se ha implementado usando la librería de java Swing, además se ha utilizado la librería externa swingx àra utilizar el componente JXDatepicker, para el componente Datepicker utilizado en la edición de perfiles.

Se han sustituido los métodos display por ventana por uno solo que realiza la llamada de la ventana pertinente gracias a un conjunto de variables estáticas numéricas que permiten identificar a la ventana que se desea llamar así como el modo en caso de que lo tenga (por ejemplo en generador de contraseñas puede aparecer en una ventana independiente desde el menú de inicio o como una dependiente desde la creación de usuario.

```
public static final int START_MENU_WINDOW = 0;
public static final int EXTERNAL_PASS_GENERATOR = 1;
public static final int CREATE_USER_WINDOW = 2;
public static final int MAIN_MENU_WINDOW_PROFILES = 3;
public static final int SETTINGS_WINDOW = 4;
public static final int ADD_PROFILE = 5;
public static final int MAIN_MENU_WINDOW_NOT = 6;
public static final int MAIN_MENU_WINDOW_SERVICES = 7;
public static final int CREATE_SERVICE_TYPE_WINDOW = 8;
public static final int EDIT_SERVICE_TYPE_WINDOW = 9;
public static final int CREATE_SERVICE_WINDOW = 10;
public static final int EDIT_SERVICE_WINDOW = 11;
public static final int VIEW_PROFILE_WINDOW = 12;
public static final int EDIT_PROFILE_WINDOW = 13;
public static final int RESOLVE_NOT_WINDOW = 14;
public static final int CHANGE_PASS = 15;
public static final int INTERNAL_PASS_GENERATOR = 16;
public static final int SERVICE_CREATE_SERVICE_TYPE_WINDOW = 17;
public static final int EDIT_SERVICE_CREATE_SERVICE_TYPE_WINDOW = 18;
public static final int PROFILE_CREATE_SERVICE_TYPE_WINDOW = 19;
public static final int EDIT_PROFILE_CREATE_SERVICE_TYPE_WINDOW = 20;
public static final int PROFILE_CREATE_SERVICE_WINDOW = 21;
public static final int EDIT_PROFILE_CREATE_SERVICE_WINDOW = 22;
```

Figura 42: Variables estáticas del controlador

```
public void DisplayWindow(int window,Object oIn,boolean removePrev) {

    switch(window){
        case START_MENU_WINDOW:
            windowStack.add(new StartMenuWindow(this));
            break;
        case EXTERNAL_PASS_GENERATOR:
            windowStack.add(new PassGeneratorWindow(this,null,PassGeneratorWindow.EXTERNAL_PASSGEN));
            break;
        case CREATE_USER_WINDOW:
            windowStack.add(new CreateUserWindow(this));
            break;
        case MAIN_MENU_WINDOW_PROFILES:
            windowStack.add(new MainMenuWindow(this,MAIN_MENU_WINDOW_PROFILES));
            break;
        case SETTINGS_WINDOW:
            windowStack.add(new SettingsWindow(this));
            break;
```

Figura 43: Cabecera del método displayWindow

```
    if(previousWindow!=null && removePrev){
        windowStack.remove(previousWindow);
        previousWindow.dispose();
    }else if(previousWindow!=null)previousWindow.setVisible(false);
    previousWindow = currentWindow;
    currentWindow = windowStack.get(windowStack.size()-1);
    currentWindow.setVisible(true);
    if(previousWindow!=null)currentWindow.setLocation(previousWindow.getLocation());
    if(previousWindow!=null)previousWindow.setVisible(false);
    if(removePrev){
        int i =0;
        while(windowStack.size()>1){
            if(!(windowStack.get(i).equals(currentWindow))){
                JFrame removing = windowStack.get(i);
                windowStack.remove(removing);
                removing.dispose();
            }else i++;
        }
    }
}
```

Figura 44: Operaciones comunes del método displayWindow

Para manejar el flujo de ventanas y permitir el regreso a pantallas anteriores se han utilizado tres variables. Dos variables contenedoras de JFrames para controlar cual es la ventana anterior y actual, y otra con una lista de todas las ventanas de la cadena de flujo(Arraylist de JFrames).

```
private ArrayList<JFrame> windowStack;
private JFrame currentWindow;
private JFrame previousWindow;
```

Figura 45: Variables del controlador para el control del flujo de navegación

Para gestionar estas variables la llamada a ventanas incluye una variable que indica si hay que ampliar la lista de ventanas del flujo o resetearlo. Unido a esto el método back da un paso atrás volviendo a la ventana anterior y desechando la actual (la implementación realizada no permite realizar operaciones “adelante” tras realizar una acción “volver”).

```
public void Back(){
    previousWindow.setLocation(currentWindow.getLocation());
    windowStack.remove(currentWindow);
    currentWindow.dispose();
    currentWindow=previousWindow;
    currentWindow.setVisible(true);
    if(windowStack.size()>1)previousWindow=windowStack.get(windowStack.indexOf(currentWindow)-1);
    else previousWindow=null;
}
```

Figura 46: Método back del controlador

Para que los botones con forma de icono en lugar de forma rectangular de la mayoría de botones (ejemplo: editar servicio) se han creado pares de iconos con fondo blanco/gris, para resaltarlos cuando un usuario pase el ratón sobre ellos y no los confunda con imágenes.

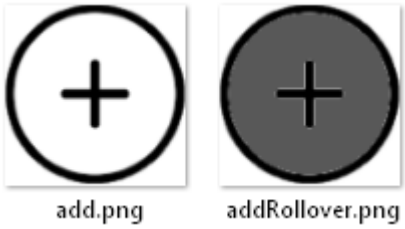


Figura 47: Iconos del botón añadir

```
Image img = new ImageIcon(getClass().getResource("/spm/Images/add.png")).getImage();
lowerButtonP.setLayout(new BoxLayout(lowerButtonP, BoxLayout.X_AXIS));
lowerButtonP.add(Box.createRigidArea(new Dimension(5, 0)));
Image newimg = null;
if(img!=null)  newimg = img.getScaledInstance( 20,20,  java.awt.Image.SCALE_SMOOTH ) ;
JButton addProfileB = new JButton(new ImageIcon( newimg ));
    img = new ImageIcon(getClass().getResource("/spm/Images/addRollover.png")).getImage() ;
    newimg = img.getScaledInstance( 20,20,  java.awt.Image.SCALE_SMOOTH ) ;
    addProfileB.setRolloverIcon(new ImageIcon(newimg));
    addProfileB.setBorderPainted(false);
    addProfileB.setBorder(null);
    addProfileB.setMargin(new Insets(0, 0, 0, 0));
    addProfileB.setContentAreaFilled(false);
    addProfileB.setAlignmentX(Component.CENTER_ALIGNMENT);
    addProfileB.setAlignmentY(Component.CENTER_ALIGNMENT);
    addProfileB.setPreferredSize(new Dimension(20,20));
    addProfileB.setMaximumSize(new Dimension(20,20));
lowerButtonP.add(addProfileB);
lowerButtonP.add(Box.createRigidArea(new Dimension(5, 0)));
JLabel addProfileL = new JLabel("Añadir perfil");
    addProfileL.setAlignmentX(Component.CENTER_ALIGNMENT);
    addProfileL.setAlignmentY(Component.CENTER_ALIGNMENT);
    addProfileL.setPreferredSize(new Dimension(80,20));
    addProfileL.setMaximumSize(new Dimension(80,20));
lowerButtonP.add(addProfileL);
```

Figura 48: Código de un botón con icono

Figura 49: Aspecto del botón Añadir perfil con el ratón encima y en otra posición

Ha sido necesario añadir un método “recalculateProportión”, para que al cambiar la proporción para un set de caracteres se mantenga que la suma de las proporciones es igual a la longitud de la contraseña.

Este método primero calcula si tiene que aumentar o disminuir el valor de la proporción, y cual de las proporciones, o si es la longitud, ha cambiado.

Después calcula si hay alguna proporción que no puede ser modificada por condiciones especiales(bloqueada o ya es 0 cuando el reajuste es negativo).

Por último distribuye el reajuste equitativamente, empezando por una proporción de manera aleatoria.

```
public void recalculateProp(int changed){
    boolean skipCaps = false;
    boolean skipLower = false;
    boolean skipNumbers = false;
    boolean skipSpecial = false;
    int fix = -1;
    if((proportion[0]+proportion[1]+proportion[2]+proportion[3])<passLength) fix = 1;
    switch(changed){
        case CAPS:
            skipCaps=true;
            break;
        case LOWER:
            skipLower=true;
            break;
        case NUMBERS:
            skipNumbers=true;
            break;
        case SPECIAL:
            skipSpecial=true;
            break;
    }
```

Figura 50: Detalle del método recalculateProp(1)

```
if(otherCharString.equals("")) skipSpecial = true;
if(locked[0]) skipCaps = true;
if(locked[1]) skipLower = true;
if(locked[2]) skipNumbers = true;
if(locked[3]) skipSpecial = true;
if(proportion[0]==0 && fix==-1) skipCaps = true;
if(proportion[1]==0 && fix==-1) skipLower = true;
if(proportion[2]==0 && fix==-1) skipNumbers = true;
if(proportion[3]==0 && fix==-1) skipSpecial = true;
```

Figura 51: Detalle del método recalculateProp(2)

```
while((proportion[0]+proportion[1]+proportion[2]+proportion[3])!=passLength){
    switch(i){
    case 0:
        if(!skipCaps) proportion[0] += fix;
        break;
    case 1:
        if(!skipLower) proportion[1] += fix;
        break;
    case 2:
        if(!skipNumbers) proportion[2] += fix;
        break;
    case 3:
        if(!skipSpecial) proportion[3] += fix;
        break;
    }
    i++;
    if (i==4) i=0;
}
```

Figura 52: Detalle del método recalculateProp(3)

Además para permitir que ese método no impida la selección de proporciones exactas (al cambiar una proporción sería posible que otra proporción que ya cambiastes cambia automáticamente), se han añadido botones para bloquear la proporción de un set de caracteres una vez hemos elegido el valor de esa proporción.

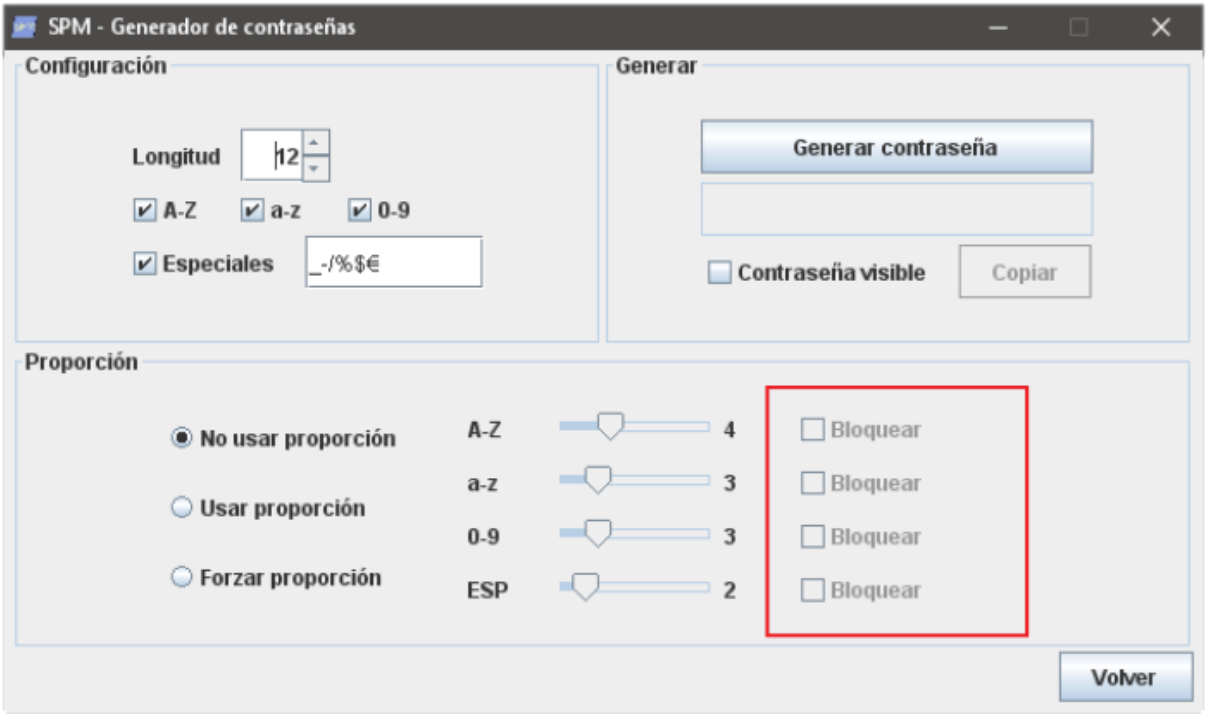


Figura 53: Pantalla del generador de contraseñas

```
AZLockCheckB.addActionListener((ActionEvent e) -> {
    if(AZLockCheckB.isSelected()){
        AZSlider.setEnabled(false);
        passGenerator.setLocked(true, PassGenerator.CAPS);
        if(passGenerator.getLocked(PassGenerator.LOWER)
                        || passGenerator.getLocked(PassGenerator.NUMBERS)
                        || passGenerator.getLocked(PassGenerator.SPECIAL)){
            if(!passGenerator.getLocked(PassGenerator.LOWER)){
                azLockCheckB.setEnabled(false);
            }
            if(!passGenerator.getLocked(PassGenerator.NUMBERS)){
                numbersLockCheckB.setEnabled(false);
            }
            if(!passGenerator.getLocked(PassGenerator.SPECIAL)){
                specialLockCheckB.setEnabled(false);
            }
        }
    }else{
        AZSlider.setEnabled(true);
        passGenerator.setLocked(false, PassGenerator.CAPS);
        if(passGenerator.getLocked(PassGenerator.LOWER)
                        || passGenerator.getLocked(PassGenerator.NUMBERS)
                        || passGenerator.getLocked(PassGenerator.SPECIAL)){
            if(!passGenerator.getLocked(PassGenerator.LOWER)){
                azLockCheckB.setEnabled(true);
            }
            if(!passGenerator.getLocked(PassGenerator.NUMBERS)){
                numbersLockCheckB.setEnabled(true);
            }
            if(!passGenerator.getLocked(PassGenerator.SPECIAL)){
                specialLockCheckB.setEnabled(true);
            }
        }
    }
    controler.updatePassGenerator();
});
```

Figura 54: ActionListener de un checkbox bloquear

Una vez bloqueamos 2 proporciones no podemos seguir bloqueando opciones ya que bloquear una opción más seria equivalente a bloquearlas todas.

5.4 Pruebas

5.4.1 Pruebas de caja blanca

Esta iteración no cubre ninguna operación especialmente sensible, aún así se han realizado pruebas de caja blanca para cotejar que la información del modelo coincide con la mostrada en la interfaz.

Para ello se ha realizado un procedimiento similar a la iteración anterior mostrando la información del modelo mientras se realizaban operaciones CRUD* en la aplicación, comprobando que siempre el modelo y la información mostrada por la interfaz coinciden.

5.4.2 Pruebas de caja negra

Se han repetido con éxito las pruebas realizadas en la primera iteración (PCN-01 a PCN-18).

PCN-19 Generar Contraseña personalizando la configuración	
Objetivo	Comprobar que el generador de contraseñas funciona con una configuración diferente a la configuración por defecto.
Precondiciones	Estar en el generador y haber cambiado la configuración.
Datos de entrada	Generar con proporción forzada Proporción [6,2,2,2}
Acción esperada	Creación de una contraseña de 12 dígitos siguiendo la distribución de la proporción (6 mayusculas, 2 minusculas, 2 números 2 caracteres especiales)
Resultado	Correcto

Tabla 90: Prueba de caja negra PCN-19

PCN-20 Introducción de una contraseña maestra no válida al crear usuario	
Objetivo	Comprobar que los requisitos de seguridad mínima de la contraseña maestra se validan
Precondiciones	Crear usuario
Datos de entrada	password
Acción esperada	Mensaje indicando que la contraseña no es válida ya que tiene que tener 8 dígitos y al menos 2 tipos de caracteres distintos
Resultado	Correcto

Tabla 91: Prueba de caja negra PCN-20

PCN-21 Generar Contraseña vaciar el campo de caracteres especiales	
Objetivo	Comprobar que el generador de contraseñas funciona con una configuración diferente a la configuración por defecto.
Precondiciones	Estar en el generador y haber cambiado la configuración.
Datos de entrada	Borra el campo de caracteres especiales
Acción esperada	Se bloquean las opciones relacionadas con los caracteres especiales excepto el campo.
Resultado	Correcto

Tabla 92: Prueba de caja negra PCN-21

PCN-22 Añadir perfil con un nuevo servicio	
Objetivo	Comprobar que se puede añadir un perfil correctamente y que la opción nuevo servicio crea un nuevo servicio en la misma operación.
Precondiciones	Usuario identificado y visualizando la lista de perfiles
Datos de entrada	email: hugo@email.com nombre:Hugo servicio: Nuevo servicio contraseña: “usar generador” tipo de servicio:desconocido icono:por defecto url: “campos vacio”
Acción esperada	Se crea un nuevo perfil y un nuevo servicio, siendo el nuevo servicio el servicio del perfil creado
Resultado	Correcto

Tabla 93: Prueba de caja negra PCN-22

PCN-23 Editar servicio con un nuevo tipo servicio	
Objetivo	Comprobar que se puede editar un servicio correctamente y que la opción nuevo tipo servicio crea un nuevo tipo de servicio en la misma operación.
Precondiciones	Usuario identificado y visualizando la lista de perfiles
Datos de entrada	tipo de servicio:Nuevo tipo de servicio icono:por defecto url: “campos vacio” Nombre: prueba color: rojo
Acción esperada	Se modifica un nuevo servicio y se crea un nuevo tipo servicio, siendo el nuevo tipo de servicio el tipo de servicio del servicio modificado.
Resultado	Correcto

Tabla 94: Prueba de caja negra PCN-23

PCN-24 Editar servicio con un nuevo tipo servicio	
Objetivo	Comprobar que se puede editar un servicio correctamente y que la opción nuevo tipo servicio crea un nuevo tipo de servicio en la misma operación.
Precondiciones	Usuario identificado y visualizando la lista de perfiles
Datos de entrada	tipo de servicio:Nuevo tipo de servicio icono:por defecto url: “campos vacio” Nombre: prueba color: rojo
Acción esperada	Se bloquean las opciones relacionadas con los caracteres especiales excepto el campo.
Resultado	Correcto

Tabla 95: Prueba de caja negra PCN-24

PCN-25 Prueba de desconexión por inactividad activada	
Objetivo	Comprobar que la aplicación bloquea el uso trás un periodo de inactividad.
Precondiciones	Usuario identificado y desconexión activada en ajustes
Datos de entrada	esperar 15 minutos
Acción esperada	Se realiza el bloqueo de uso y se notifica de la desconexión por inactividad
Resultado	Correcto

Tabla 96: Prueba de caja negra PCN-25

PCN-26 Prueba de desconexión por inactividad activada cambio de ajustes	
Objetivo	Comprobar que la aplicación bloquea el uso trás un periodo de inactividad.
Precondiciones	Usuario identificado y desconexión activada en ajustes
Datos de entrada	esperar 30 minutos
Acción esperada	Se realiza el bloqueo de uso y se notifica de la desconexión por inactividad
Resultado	Correcto

Tabla 97: Prueba de caja negra PCN-26

PCN-27 Prueba de desconexión por inactividad desactivada	
Objetivo	Comprobar que la aplicación no bloquea el uso trás un periodo de inactividad.
Precondiciones	Usuario identificado y desconexión desactivada en ajustes
Datos de entrada	esperar 15 minutos
Acción esperada	No ocurre nada
Resultado	Correcto

Tabla 98: Prueba de caja negra PCN-27

PCN-28 Servicios y Tipos de servicio por defecto	
Objetivo	Comprobar que la aplicación tiene un conjunto de servicios y tipos de servicio por defecto
Precondiciones	Crear usuario
Datos de entrada	ninguna
Acción esperada	La aplicación tiene cargados los servicios y tipos de servicio por defecto
Resultado	Correcto

Tabla 99: Prueba de caja negra PCN-28

PCN-29 Crear tipo de servicio	
Objetivo	Comprobar que la aplicación crea tipos de servicio correctamente
Precondiciones	Crear usuario
Datos de entrada	Nombre:Prueba Color:verde
Acción esperada	La aplicación crea el nuevo tipo de servicio
Resultado	Correcto

Tabla 100: Prueba de caja negra PCN-29

PCN-30 Edición solo para tipos y servicios personalizados	
Objetivo	Comprobar que la aplicación crea solo permite editar y eliminar servicios y tipos de servicio creados por el usuario.
Precondiciones	Usuario identificado, visualizando administrar servicios, al menos un servicio y tipo de servicio creado por el usuario almacenados en la aplicación.
Datos de entrada	ninguna
Acción esperada	La aplicación solo permite editar y eliminar los servicios y tipos de servicio creados por el usuario.
Resultado	Correcto

Tabla 101: Prueba de caja negra PCN-30

PCN-31 Edición solo para tipos y servicios personalizados	
Objetivo	Comprobar que la aplicación crea solo permite editar y eliminar servicios y tipos de servicio creados por el usuario.
Precondiciones	Usuario identificado, visualizando administrar servicios, al menos un servicio y tipo de servicio creado por el usuario almacenados en la aplicación.
Datos de entrada	ninguno
Acción esperada	La aplicación solo permite editar y eliminar los servicios y tipos de servicio creados por el usuario.
Resultado	Correcto

Tabla 102: Prueba de caja negra PCN-31

PCN-32 Eliminar servicio	
Objetivo	Comprobar que la aplicación elimina servicios correctamente y que adjudica el servicio desconocido a los perfiles de dicho servicio.
Precondiciones	Usuario identificado, visualizando administrar servicios, al menos un servicio con un perfil de dicho servicio almacenados en la aplicación
Datos de entrada	ninguno
Acción esperada	El servicio es eliminado y el perfil cambia su servicio a desconocido
Resultado	Correcto

Tabla 103: Prueba de caja negra PCN-32

PCN-33 Eliminar tipo de servicio	
Objetivo	Comprobar que la aplicación elimina servicios correctamente y que adjudica el servicio desconocido a los servicios de dicho tipo de servicio.
Precondiciones	Usuario identificado, visualizando administrar servicios, al menos un tipo de servicio con un servicio de dicho tipo de servicio almacenados en la aplicación
Datos de entrada	ninguno
Acción esperada	El servicio es eliminado y el servicio cambia su tipo de servicio a desconocido
Resultado	Correcto

Tabla 104: Prueba de caja negra PCN-33

Capítulo 6

Tercera Iteración

6.1 Análisis

6.1.1 Características

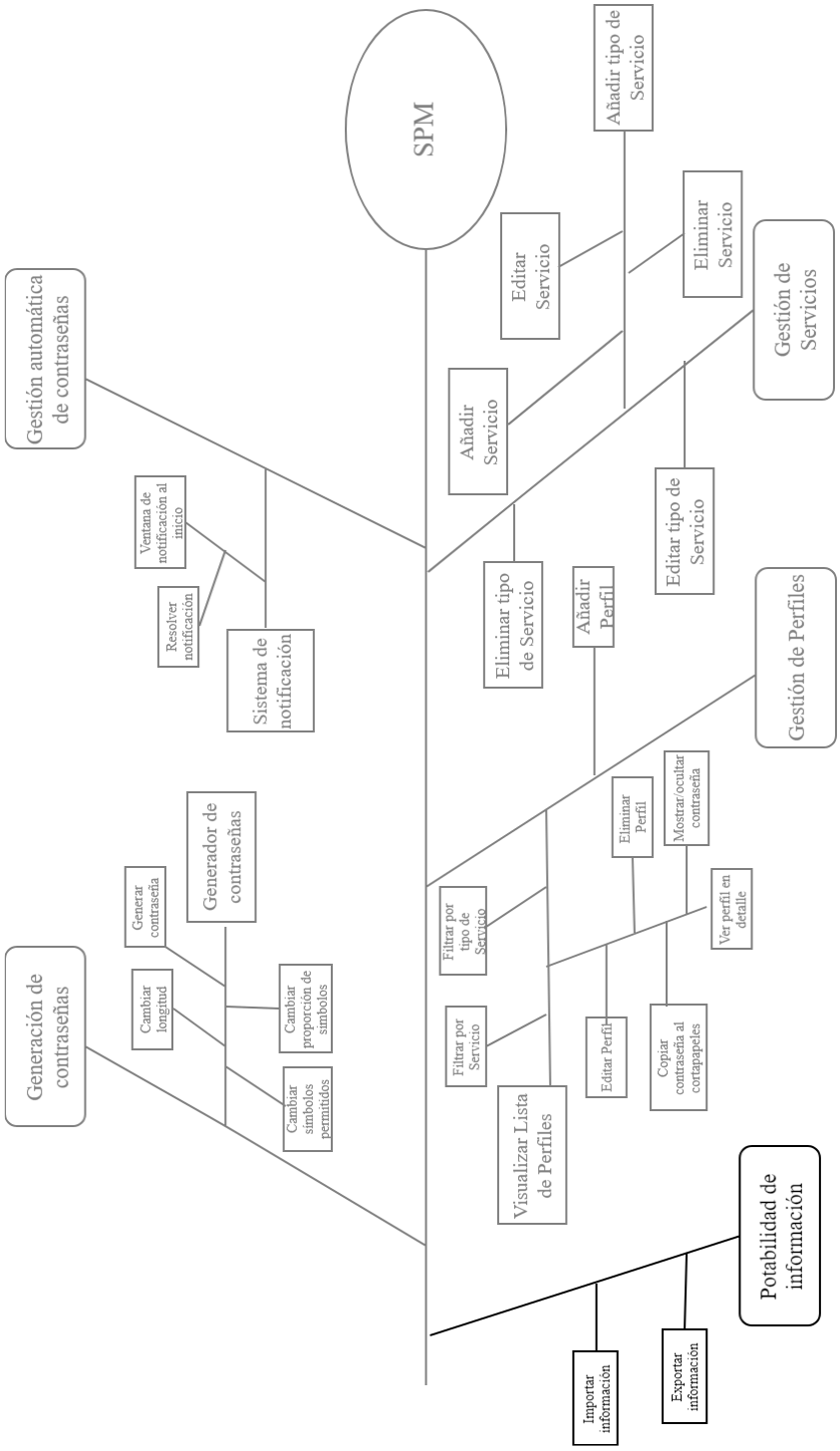


Figura 55: árbol de características de la tercera iteración

6.1.2 Actores

No se ha añadido ningún nuevo actor en esta iteración.

6.1.3 Requisitos de usuario

Los requisitos de usuario de esta iteración son los siguientes:

RU-31: Un usuario podrá generar un archivo para exportar la información almacenada por la aplicación

RU-32: Un usuario podrá importar información de otra instancia de la aplicación

RU-33: Un usuario podrá elegir si la información importada prevalece o no sobre la que guarda la aplicación

6.1.4 Diagrama de casos de uso

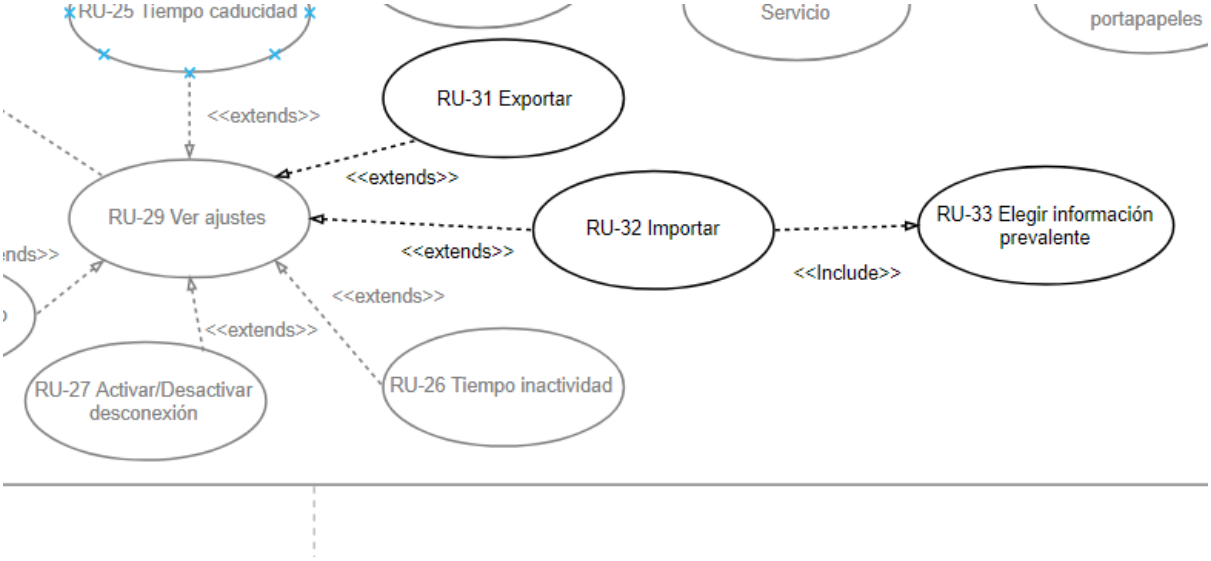


Figura 56:Detalle del diagrama de casos de uso de la tercera iteración

6.1.5 Especificación de requisitos de Usuario

US-23	Exportar información	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJS-3	
Requisitos asociados	RU-31	
Descripción	El usuario podrá generar un archivo para exportar la información almacenada por la aplicación.	
Precondición	El usuario está en la pantalla de ajustes	
Secuencia normal	Paso	Acción
	1	El usuario pulsa en la opción exportar
	2	La aplicación permite elegir el directorio donde guardar el archivo
	3	El usuario elige el directorio
	4	La aplicación crea el archivo de exportación.
	5	El caso de uso a finalizado con éxito
Postcondición		
Excepciones	Paso	Acción
	3b	El usuario usuario cancela la elección de fichero. El caso de uso finaliza sin éxito.
Comentarios		

Tabla 105:Especificación del US-23

US-24	Importar información
Versión	1.0
Autor	Hugo Gilarranz
Objetivos asociados	OBJS-3
Requisitos asociados	RU-32, RU-32
Descripción	El usuario podrá importar información desde un archivo externo generado

	por otra instancia de la aplicación	
Precondición	El usuario está en la pantalla de ajustes	
Secuencia normal	Paso	Acción
	1	El usuario pulsa en la opción importar
	2	La aplicación permite elegir el archivo para importar
	3	El usuario elige el archivo
	4	La aplicación carga el archivo
	5	La aplicación comprueba si el el comprobante coincide con el del usuario
	6	La aplicación pregunta al usuario qué información prevalece
	7	El usuario elige la información almacenada en la aplicación
	8	La aplicación guarda la información del archivo descartando los perfiles, servicios y tipos de servicio que estén en la aplicación
	9	El caso de uso a finalizado con éxito
Postcondición		
Excepciones	Paso	Acción
	3b	El usuario usuario cancela la elección del archivo. El caso de uso finaliza sin éxito.
	4b	El usuario selecciona un archivo no válido. Vuelta al punto 2.
	6b	El comprobante no coincide, y la aplicación solicita al usuario que introduzca la contraseña para el archivo de importación.
	7b	El usuario introduce la contraseña
	8b	La aplicación comprueba que la contraseña es válida (si no lo es vuelve al paso 7b).Si lo es continúa con el paso 6 normal
	7c	El usuario elige la información importada
	8c	La aplicación guarda la información del archivo sobreescribiendo los perfiles, servicios y tipos de servicio que estén en la aplicación
	9c	el caso de uso a finalizado con éxito
Comentarios		

Tabla 106:Especificación del US-24

6.1.6 Requisitos de información

ENT- 06	Archivo Exp/Imp		Versión	1.0
Definición	Clase que encapsula la información para exportar/importar			
Consideraciones				
ATRIBUTOS				
ID	Nombre	Descripción	Dominio	UNIQUE	NULL	Notas
	Contraseña	128 primeros bits de la contraseña pasada por el algoritmo SHA-384	VARCHAR (128)	SÍ	NO	
	Perfiles	Perfiles codificados que se exportan	ArrayList	NO	NO	
	Servicios	Servicios que se exportan	ArrayList	NO	NO	
	Tipo de Servicio	Tipos de Servicio que se exportan	ArrayList	NO	NO	
	Contraseñas caducadas	Contraseñas caducadas que se exportan	ArrayList	NO	NO	

Tabla 107:Especificación de la ENT-06

6.1.7 Requisitos no funcionales

Requisitos no funcionales de Seguridad:

NFS-08 Se deben codificar con AES-256 los perfiles que se copien a un archivo de Exp/Imp

6.2 Diseño

6.2.1 Arquitectura lógica

La arquitectura lógica de esta aplicación no ha sufrido cambios con respecto a lo expuesto en la iteración anterior.

6.2.2 Arquitectura física

La arquitectura física de esta aplicación no ha sufrido cambios con respecto a lo expuesto en la iteración anterior.

6.2.3 Diagrama de clases

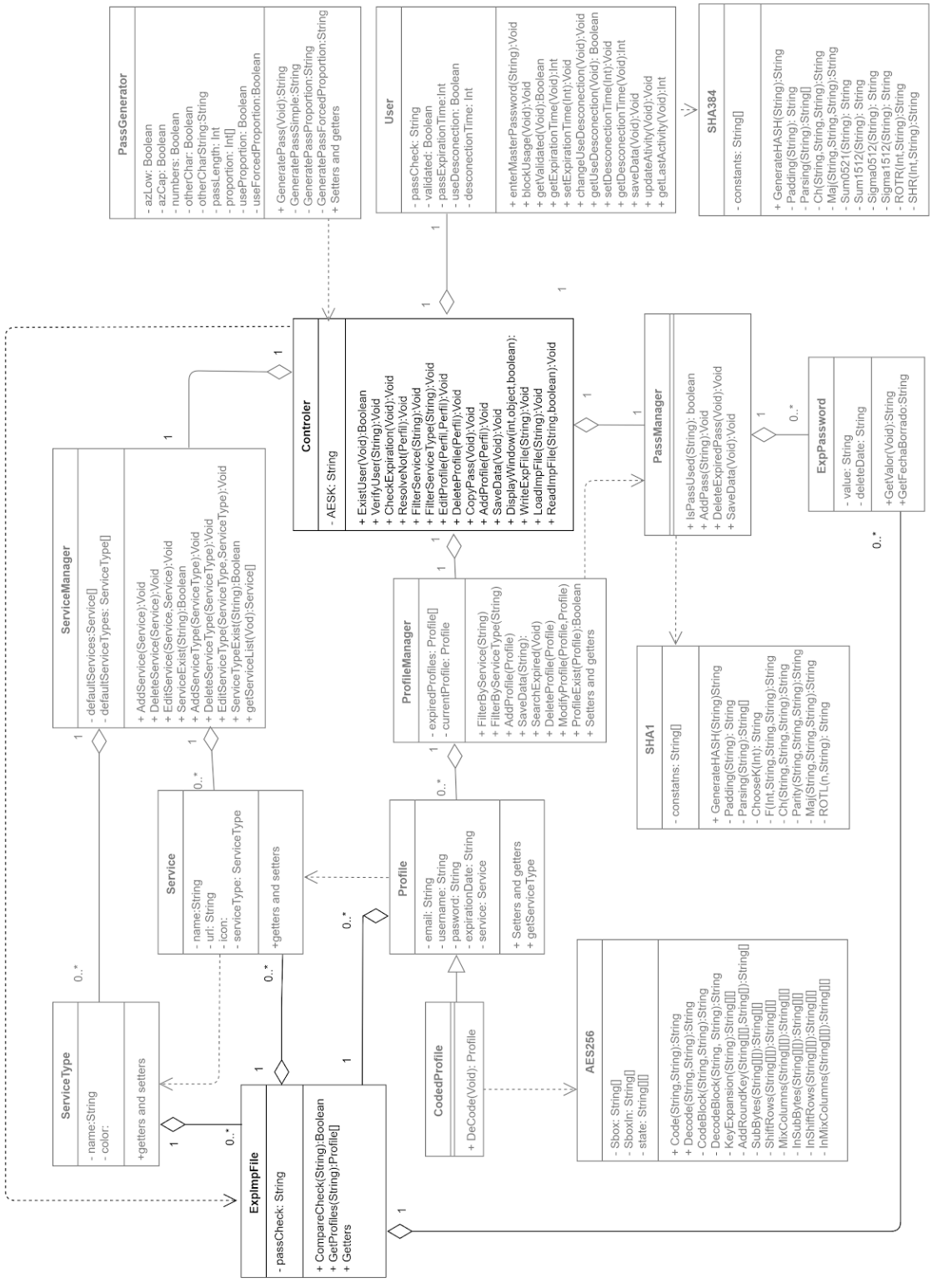


Figura 57:Diagrama de clases de la tercera iteración

6.2.4 Interfaz gráfica

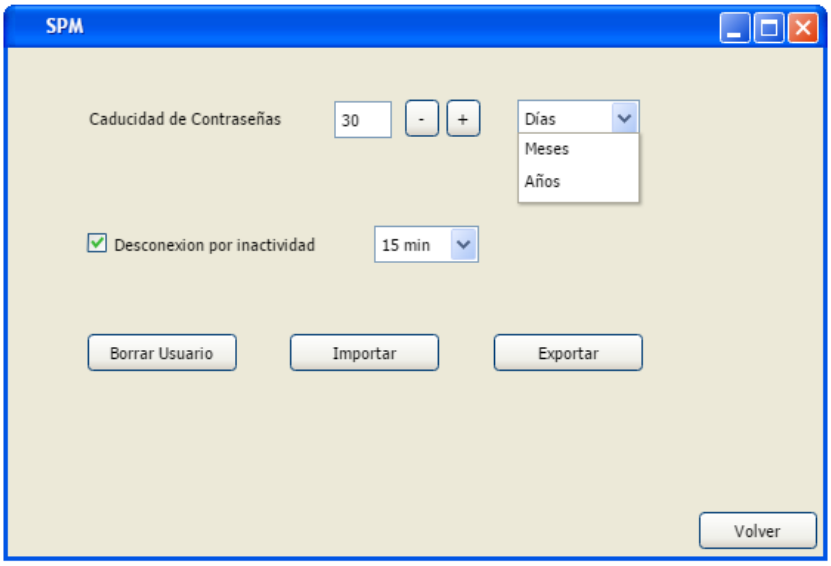
Ajustes	
Descripción	Pantalla que permite visualizar y modificar los ajustes de la aplicación
Activación	Desde el menú principal usar el botón “Ajustes” en la esquina superior derecha.
Boceto	
Eventos	Ver anterior diseño para ajustes -El botón “Importar”: abre un explorador para seleccionar el archivo a importar. Si el archivo es válido muestra la ventana importar. -El botón “Exportar”: abre un explorador de archivos. Una vez seleccionado un directorio guarda en ese directorio un archivo con la información de la aplicación

Tabla 108: Diseño de la ventana Ajustes

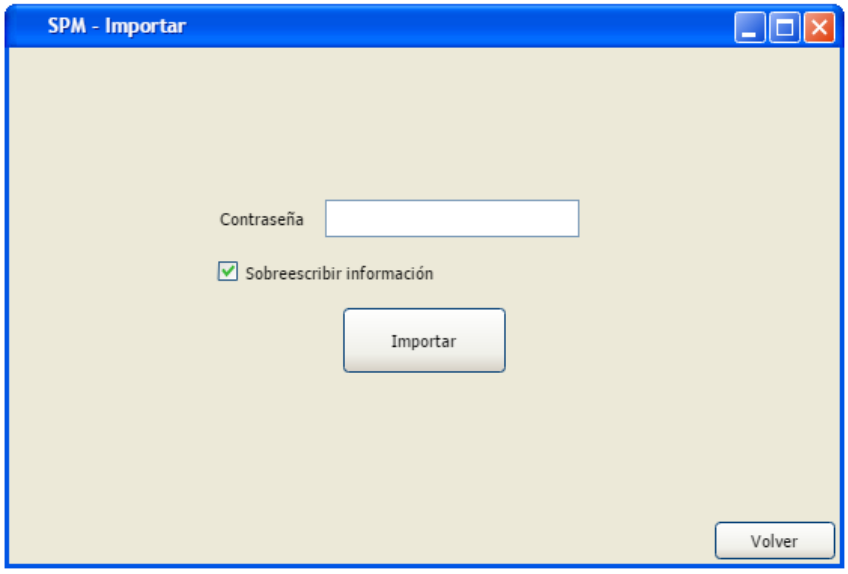
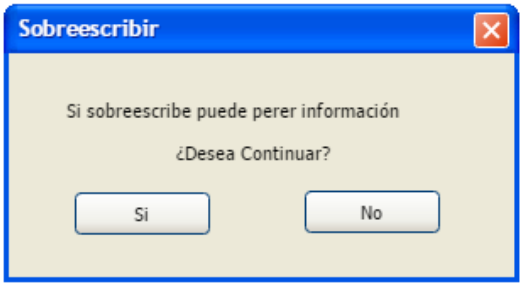
Importar	
Descripción	Pantalla que permite introducir datos necesarios para importar un archivo
Activación	Desde menú de ajustes, pulsar el botón importar
Boceto	
Eventos	-El campo de texto contraseña aparece desactivado si el comprobante para la contraseña del archivo a importar coincide con el de la contraseña comprobante de la aplicación. -El checkbox “Sobreescribir información” indica si prevalece la información de la aplicación (no seleccionado), o la del archivo (seleccionado) -El botón “importar”, si el checkbox está seleccionado muestra una pantalla de confirmación. Si no lo está importar el archivo -El botón “volver”, regresa a la pantalla de ajustes

Tabla 109: Diseño de la ventana Importar

Confirmar importar sobreescribiendo	
Descripción	Pantalla para confirmar la importación de un archivo sobreescribiendo
Activación	Desde la pantalla importar, pulsar el botón “importar” con el checkbox “Sobreescribir información” seleccionado.
Boceto	

Eventos	-El botón “Si” importa el archivo sobreescribiendo la información -El botón “No”, mantiene la vista en la pantalla de importar archivo.

Tabla 110: Diseño de la ventana importar sobrescribiendo

6.3 Implementación

Dada la brevedad de esta iteración el único elemento de interés es la lectura del archivo importado, más concretamente la elección de los elementos que se guardan y cuáles se desechan, o que elementos ya existentes se sobrescriben.

Una vez seleccionado un archivo válido se pide al usuario la contraseña para decodificar el archivo si no coincide con la del programa, y la política para sobreescribir los datos. Una vez obtenidos se llama a la siguiente función:

```
public void readImpFile(String AESK, boolean overwrite){
```

Figura 58: Cabecera del método readImpFile

En caso de que la contraseña sea la de la aplicación se utiliza esta otra llamada que hace uso de la anterior

```
public void readImpFile(boolean overwrite){
    readImpFile(AESK,overwrite);
}
```

Figura 59: Método readImpFile cuando coinciden las contraseñas

El método itera por los ArrayList de servicios, y contraseñas usadas del archivo importado de la siguiente forma:

```
for(ServiceType serviceType: loadedFile.getServiceTypeList()){
    if(serviceManager.serviceTypeExist(serviceType.getName())){
        if(overwrite){
            for(ServiceType oldServiceType: serviceManager.getServiceTypes()){
                if(oldServiceType.getName().equals(serviceType.getName())){
                    serviceManager.editServiceType(serviceType, oldServiceType);
                }
            }
        }
    }else{
        serviceManager.addServiceType(serviceType);
    }
}
```

Figura 60: Bucle para iterar los tipos de servicios

Primero determina si el elemento a comprobar existe ya en la aplicación o no, y según la política de escritura en caso de que exista lo sustituye o lo ignora. En caso de que no exista lo añade independientemente de la política.

Para los servicios ya que estos dependen de un tipo de servicio, además de lo anterior se realiza una operación extra iterando por los tipos de servicio existentes para ligarlos y que no exista duplicidad de objetos con las mismas propiedades.

```
for(Service service: loadedFile.getServiceList()){
    if(serviceManager.serviceExist(service.getName())){
        if(overwrite){
            for(ServiceType serviceType: serviceManager.getDefaultServiceTypes()){
                if(serviceType.getName().equals(service.getServiceType().getName())){
                    service.setServiceType(serviceType);
                }
            }
            for(ServiceType serviceType: serviceManager.getServiceTypes()){
                if(serviceType.getName().equals(service.getServiceType().getName())){
                    service.setServiceType(serviceType);
                }
            }
            for(Service oldService: serviceManager.getServiceList()){
                if(oldService.getName().equals(service.getName())){
                    serviceManager.editService(service, oldService);
                }
            }
        }
    }else{
        for(ServiceType serviceType: serviceManager.getDefaultServiceTypes()){
                if(serviceType.getName().equals(service.getServiceType().getName())){
                    service.setServiceType(serviceType);
                }
            }
            for(ServiceType serviceType: serviceManager.getServiceTypes()){
                if(serviceType.getName().equals(service.getServiceType().getName())){
                    service.setServiceType(serviceType);
                }
            }
        serviceManager.addService(service);
    }
}
```

Figura 61: Bucle para iterar los servicios

Para los perfiles el método realiza la misma operación que los servicios, iterando en este caso con los servicios. Como pequeña nota de discordancia, se ha de utilizar la clave aportada para transformar los perfiles codificados, que almacena el archivo de exportación, en perfiles normales.

```
for(Profile profile: loadedFile.getProfileList(AESK)){
```

Figura 62: Decodificación de la lista de perfiles

6.4 Pruebas

6.4.1 Pruebas de caja blanca

Para esta iteración se ha seguido el mismo procedimiento que en la anterior, pero en este caso realizando operaciones de importación y exportación de archivos comprobando que los valores en el modelo eran iguales a los contenidos por los archivos, y a los mostrados en la interfaz trás las operaciones de importación.

6.4.2 Pruebas de caja negra

PCN-34 Prueba de exportación importación	
Objetivo	Comprobar que la aplicación exporta e importa correctamente.
Precondiciones	Usuario identificado, en la pantalla de ajustes, con un servicio, un perfil y un tipo de servicio.
Datos de entrada	Exportar el archivo Eliminar el servicio, el perfil y el tipo de servicio. Importar el archivo
Acción esperada	La información del perfil, el servicio y el tipo de servicio es la misma que antes de borrarlos.
Resultado	Correcto

Tabla 111: Prueba de caja negra PCN-34

PCN-35 Prueba de exportación importación sobrescribir elementos	
Objetivo	Comprobar que la aplicación sobrescribe la información almacenada correctamente.
Precondiciones	Usuario identificado, en la pantalla de ajustes, con un servicio, un perfil y un tipo de servicio.
Datos de entrada	Exportar el archivo Editar el servicio, el perfil y el tipo de servicio. Importar el archivo
Acción esperada	La información del perfil, el servicio y el tipo de servicio es la misma que antes de editarlos.
Resultado	Correcto

Tabla 112: Prueba de caja negra PCN-35

PCN-36 Prueba de exportación importación sin sobrescribir elementos	
Objetivo	Comprobar que la aplicación no sobrescribe la información almacenada.
Precondiciones	Usuario identificado, en la pantalla de ajustes, con un servicio, un perfil y un tipo de servicio.
Datos de entrada	Exportar el archivo Editar el servicio, el perfil y el tipo de servicio. Importar el archivo
Acción esperada	La información del perfil, el servicio y el tipo de servicio es la misma que después de editarlos.
Resultado	Correcto

Tabla 113: Prueba de caja negra PCN-36

PCN-37 Prueba de importación con contraseña distinta	
Objetivo	Comprobar que la aplicación distingue si la contraseña del archivo es distinta, y que permite introducir la contraseña para el archivo correctamente.
Precondiciones	Usuario identificado, en la pantalla de ajustes, con un servicio, un perfil y un tipo de servicio exportados desde un usuario con clave distinta.
Datos de entrada	Importar el archivo Contraseña del archivo correcta
Acción esperada	La información del perfil, el servicio y el tipo de servicio se ha importado correctamente.
Resultado	Correcto

Tabla 114: Prueba de caja negra PCN-37

PCN-38 Prueba de importación con contraseña distinta incorrecta	
Objetivo	Comprobar que la aplicación distingue si la contraseña del archivo es distinta, y que no permite introducir una contraseña incorrecta.
Precondiciones	Usuario identificado, en la pantalla de ajustes, con un servicio, un perfil y un tipo de servicio exportados desde un usuario con clave distinta.
Datos de entrada	Importar el archivo Contraseña del archivo incorrecta
Acción esperada	La aplicación notifica al usuario que la contraseña no es correcta
Resultado	Correcto

Tabla 115: Prueba de caja negra PCN-38

Parte III

Manuales

Capítulo 7

Manuales

7.1 Manual de instalación

Extraiga la carpeta SPM comprimida en zip, que puede encontrar en el CD en la carpeta “Programa”, en la carpeta donde desee almacenar la aplicación.

Una vez extraída ya se puede utilizar la aplicación ejecutando el archivo SPM.jar. Para el perfecto funcionamiento de la aplicación no se debe mover dicho archivo, por lo que se recomienda la creación de un acceso directo para mayor comodidad.

7.2 Manual de usuario

7.2.1 Ventanas principales

A continuación se describen las principales pantallas de la aplicación:

Generador de Contraseñas

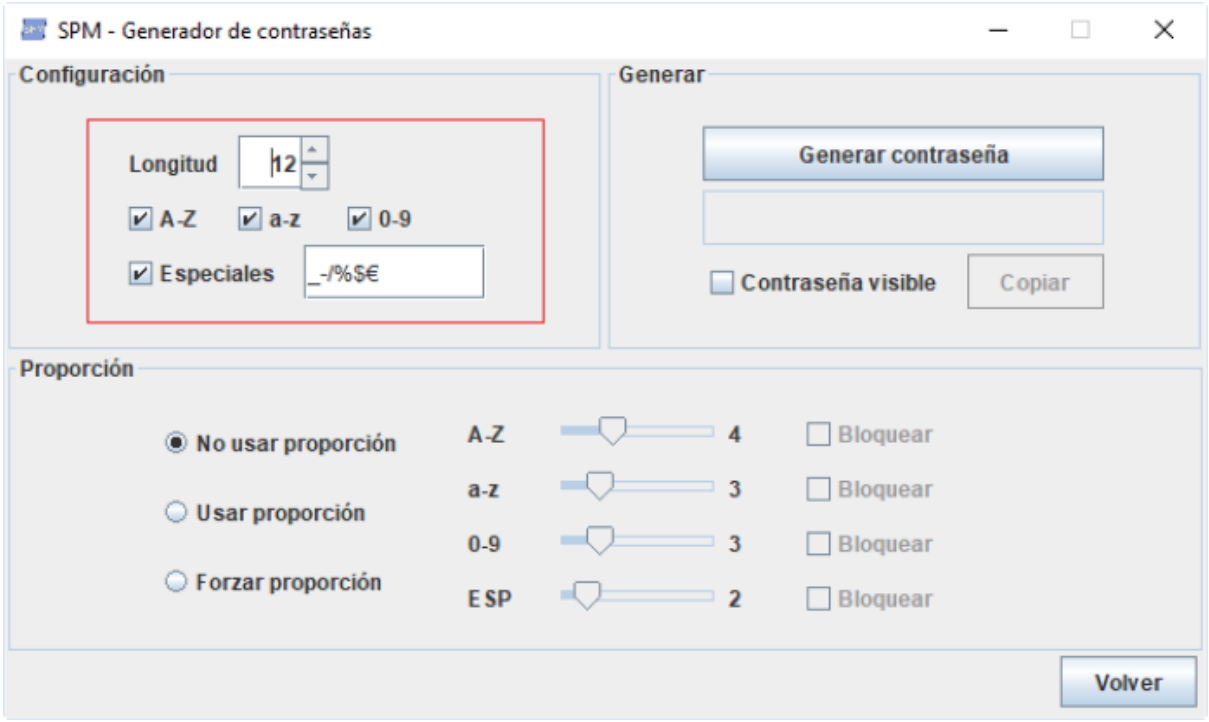


Figura 63: Generador de Contraseñas (1)

En esta sección encontramos 4 checkbox que permiten seleccionar o deseleccionar los diferentes tipos de caracteres posibles (Mayusculas, minusculas, numeros y caracteres especiales), un campo de texto en el cual podemos ver los caracteres especiales que se están usando en este momento, además de poder modificarlos, y por último un spin box que permite modificar la longitud de la contraseña a generar.

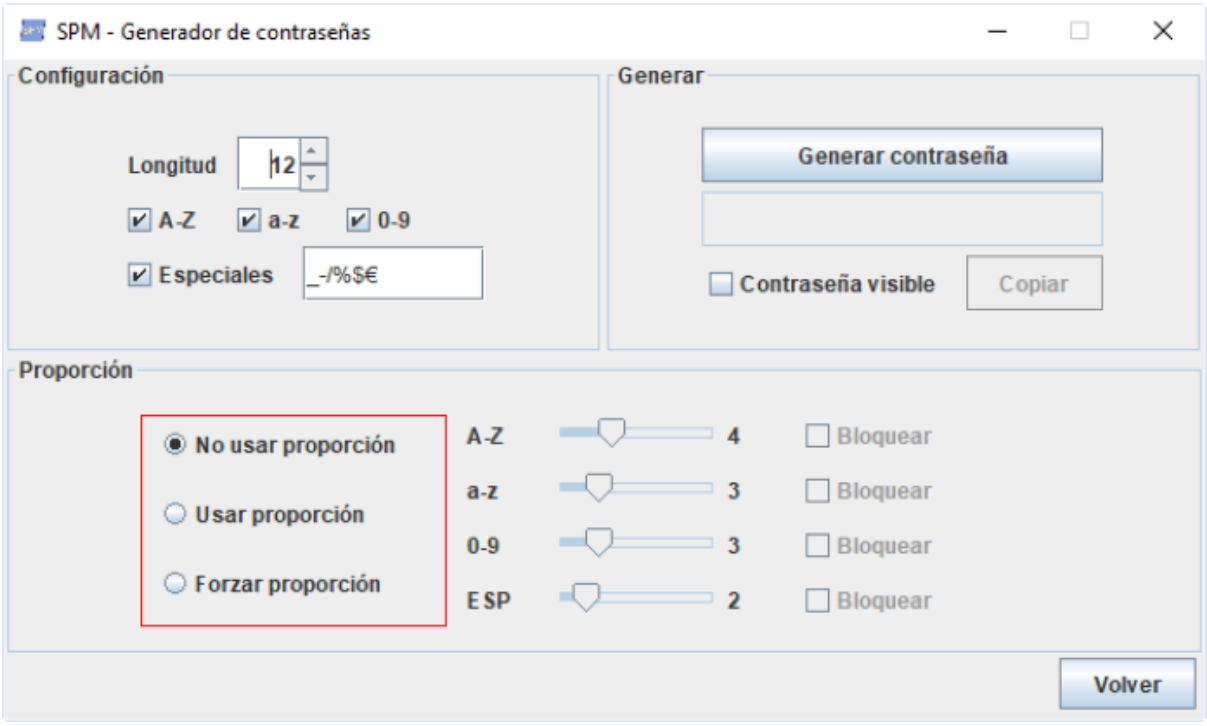


Figura 64: Generador de Contraseñas (2)

Estos tres radio button permiten elegir el método de creación de contraseñas. Con “No usar proporción”, la contraseña será totalmente aleatoria teniendo al menos un carácter de cada uno de los set seleccionados. Con “Usar proporción” se utilizara la proporción seleccionada, de manera que que el resultado más probable sea el que se ajusta a la proporción, pero sin ser seguro. Y por último con “Forzar proporción” la contraseña generada seguirá de manera estricta la proporció

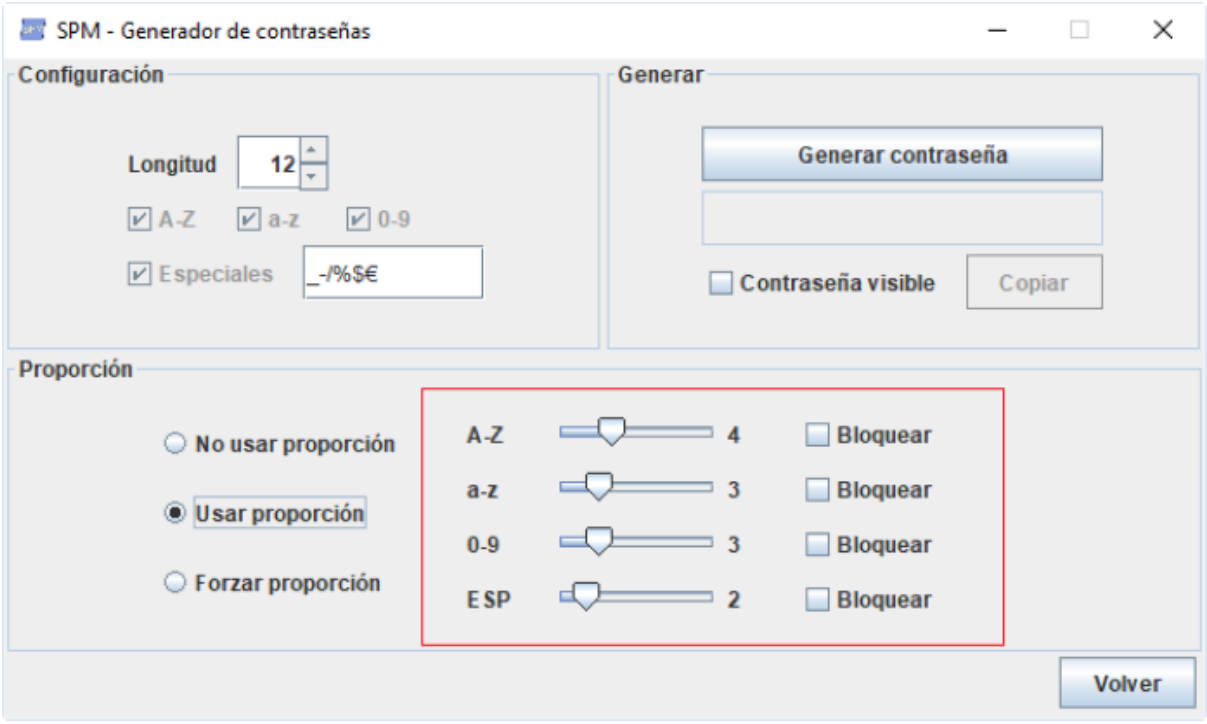


Figura 65: Generador de Contraseñas (3)

Los slider permiten ajustar individualmente la proporción de cada uno de los sets de caracteres.

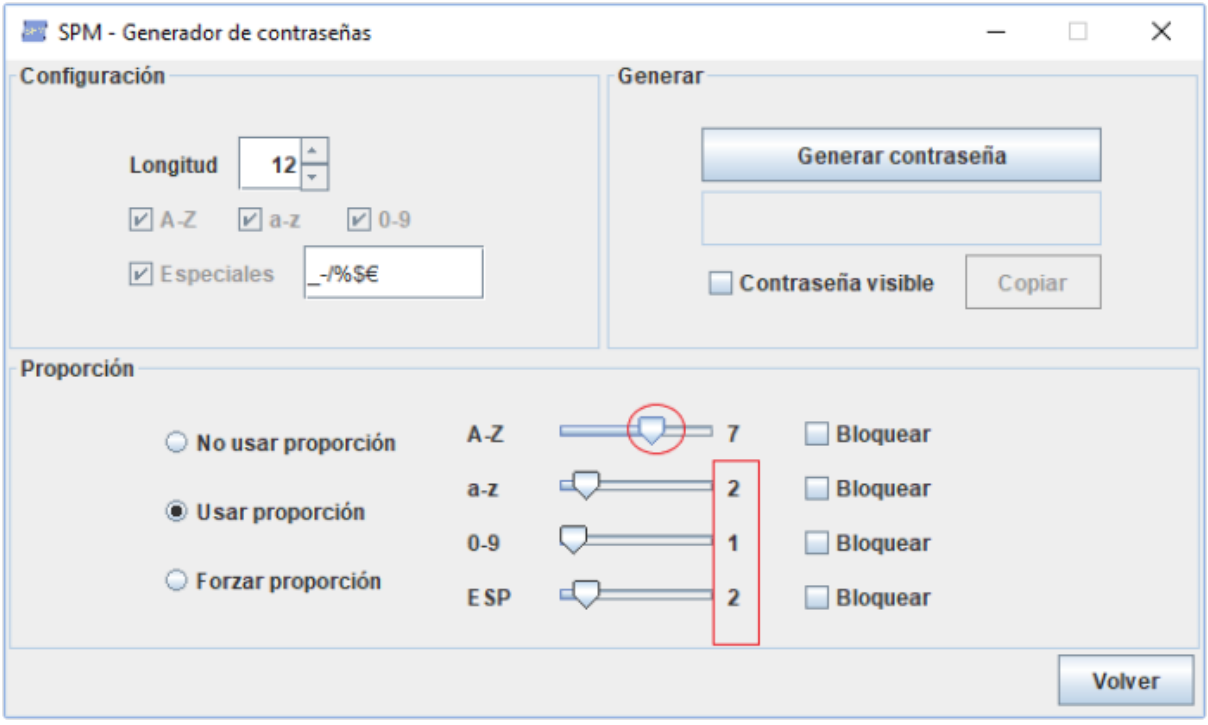


Figura 66: Generador de Contraseñas (4)

El resto de slider se ajustan automáticamente para que la suma de los valores sea igual a la longitud de la contraseña.

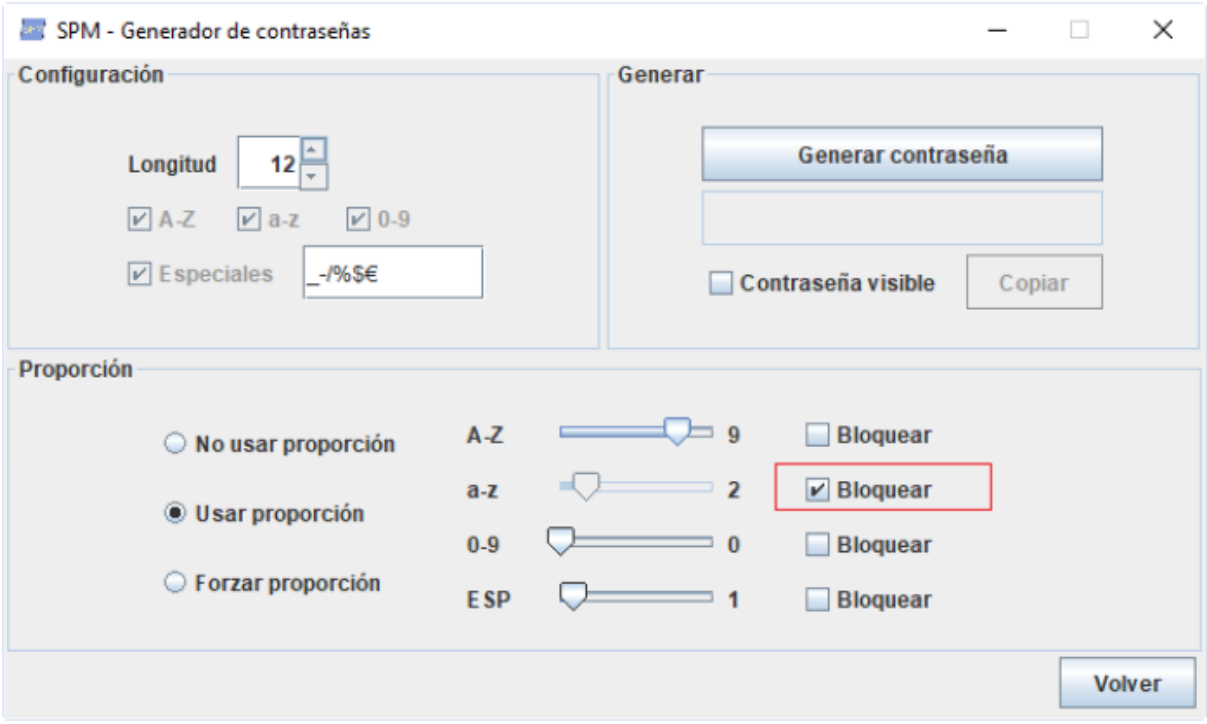


Figura 67: Generador de Contraseñas (5)

Para poder seleccionar un valor exacto se puede bloquear un slider, para que el valor no se modifique automáticamente al modificar otro valor. Se pueden bloquear hasta un máximo de dos sliders.

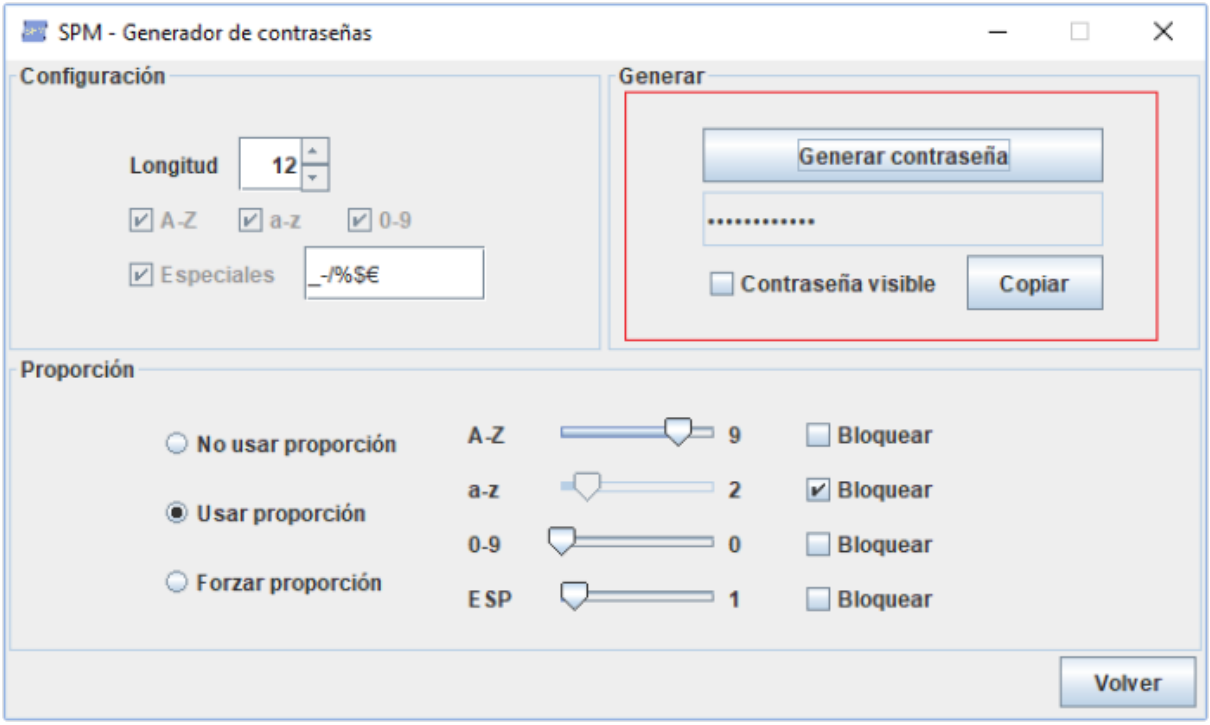


Figura 68: Generador de Contraseñas (6)

Una vez escogida la configuración se puede generar una contraseña con el botón “Generar contraseña”, y una vez generada mostrarla actualizando el checkbox “contraseña visible” o copiarla al portapapeles con el botón “Copiar”.

Menú principal (Ver perfiles)

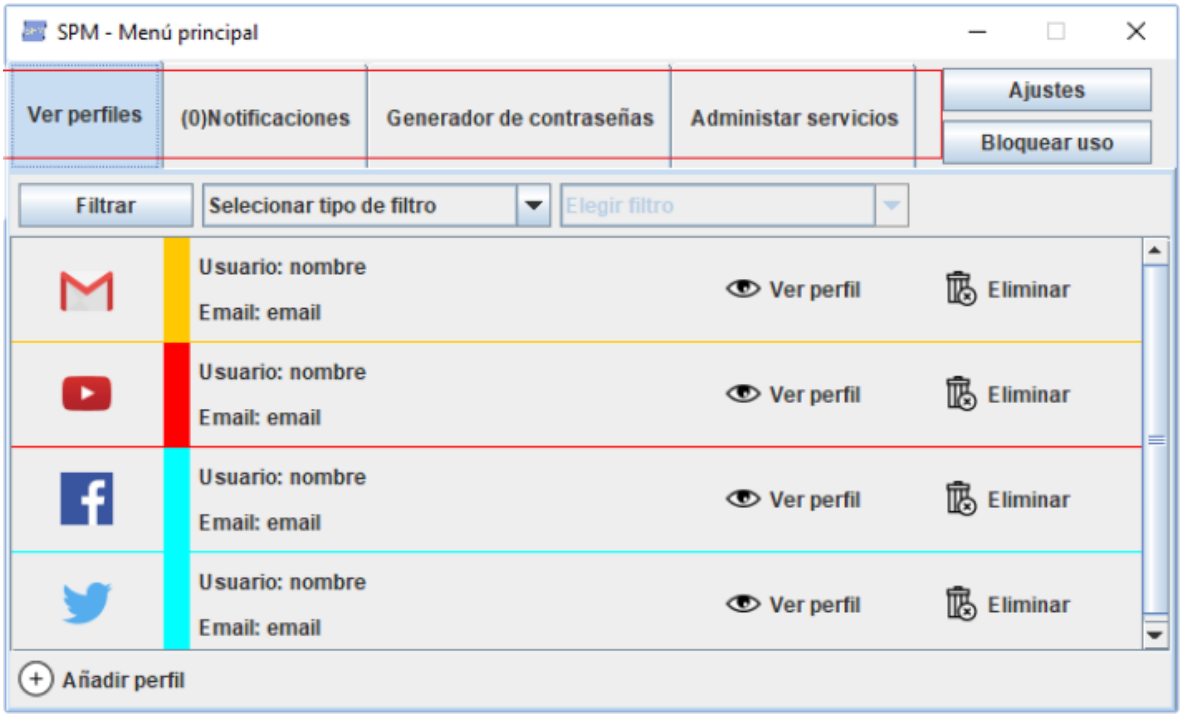


Figura 69: Menú principal, ver perfiles (1)

Cada una de las pestañas nos permite movernos por el menú principal.

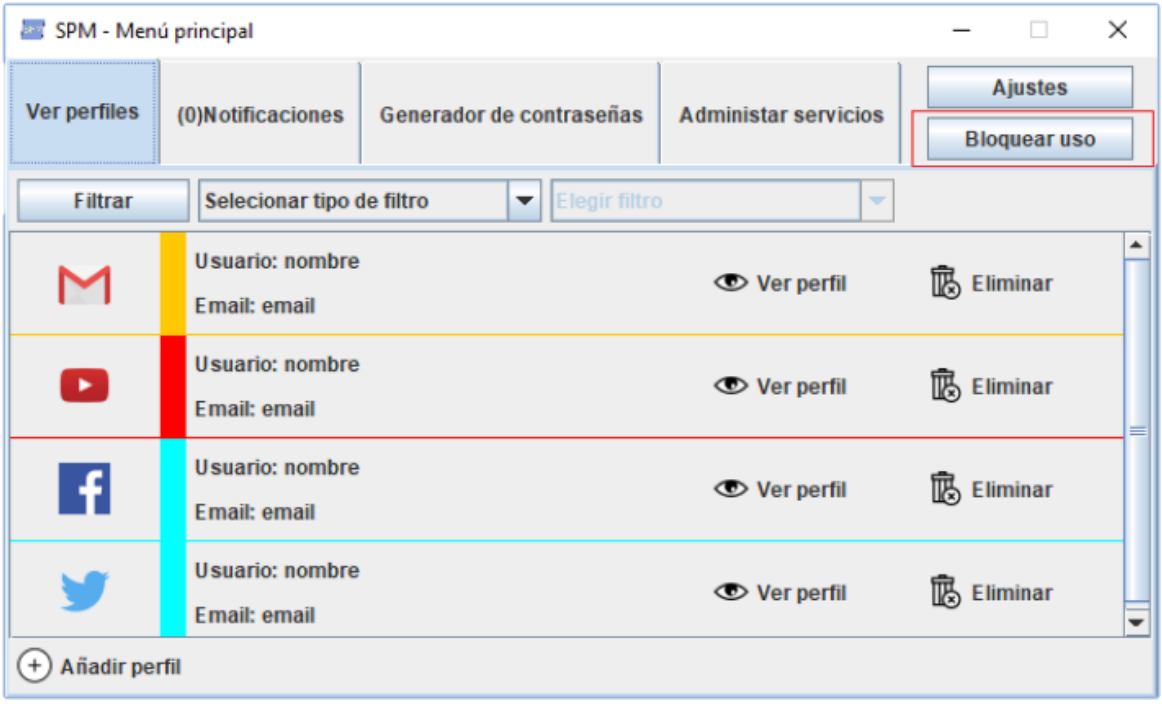


Figura 70: Menú principal, ver perfiles (2)

El botón “Bloquear uso” permite bloquear el uso eliminado la información de la memoria y volviendo al menú de inicio.

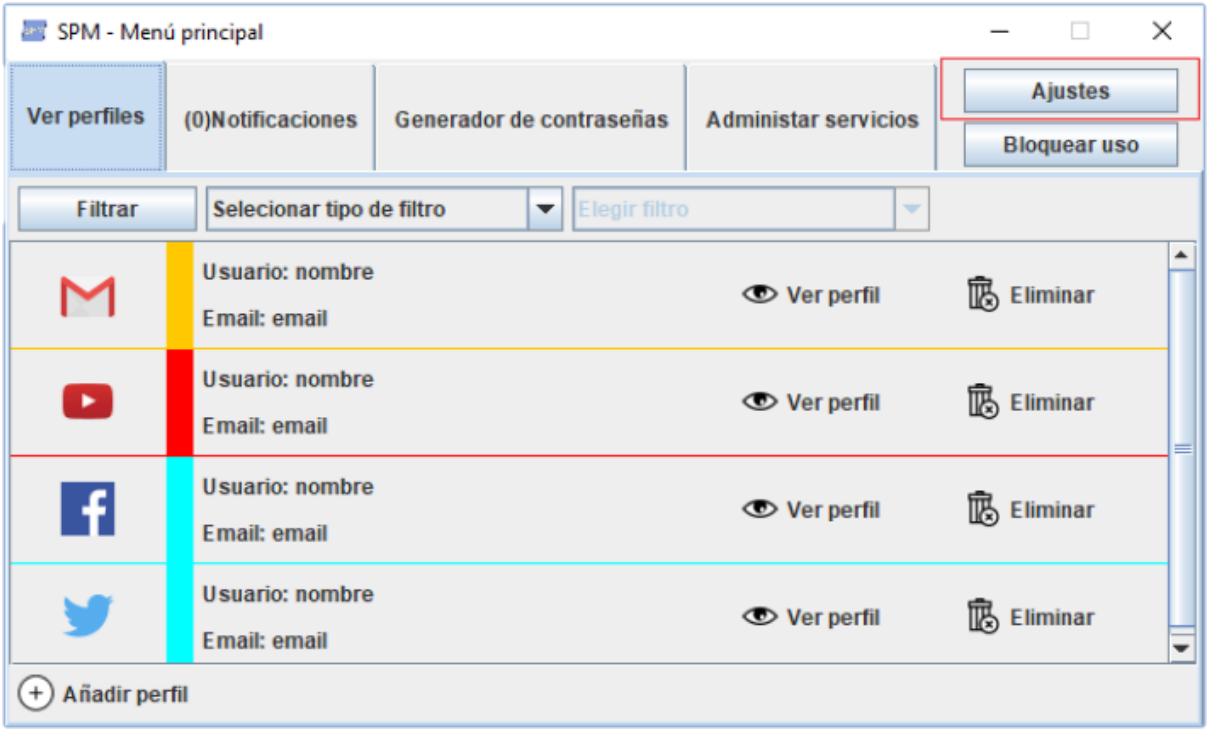


Figura 71: Menú principal, ver perfiles (3)

El botón “Ajustes” permite abrir la pantalla de ajustes.

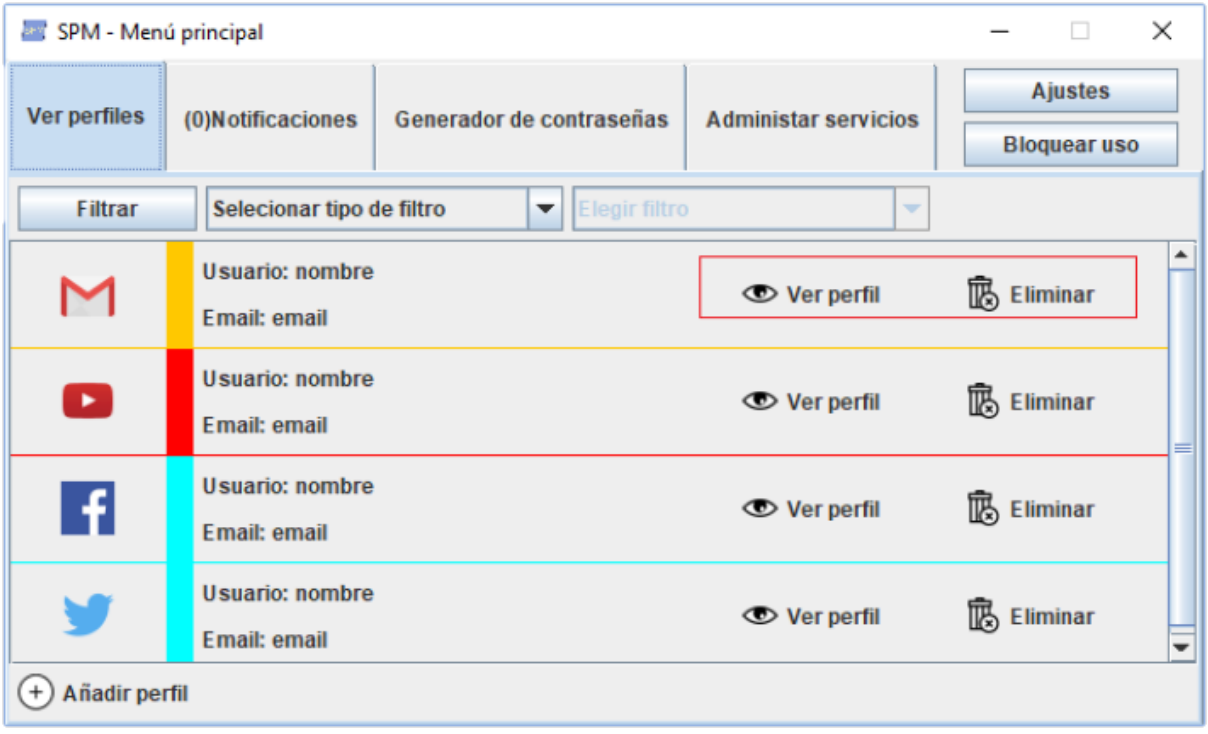


Figura 72: Menú principal, ver perfiles (4)

Los botones “Ver perfil” y “Eliminar” permiten ver en detalle o eliminar el perfil correspondiente.

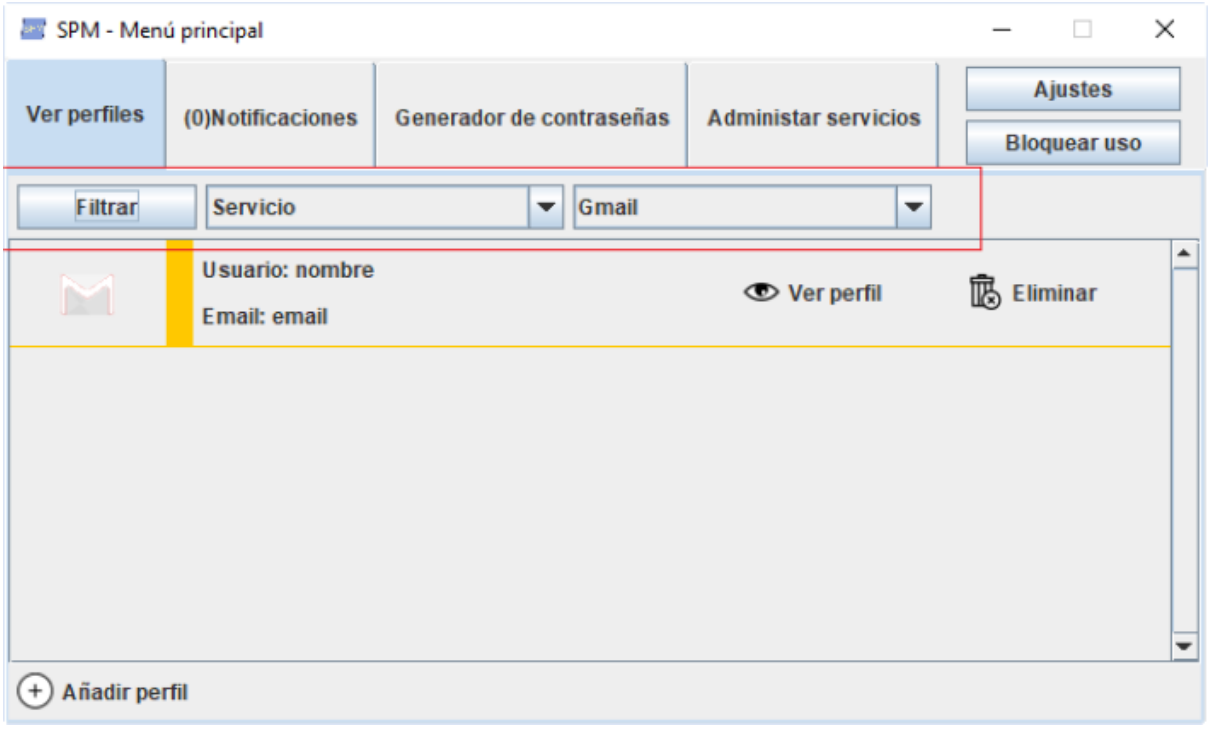


Figura 73: Menú principal, ver perfiles (5)

Con los dropbox podemos seleccionar un filtro eligiendo entre filtrar por servicio o por tipo de servicio.

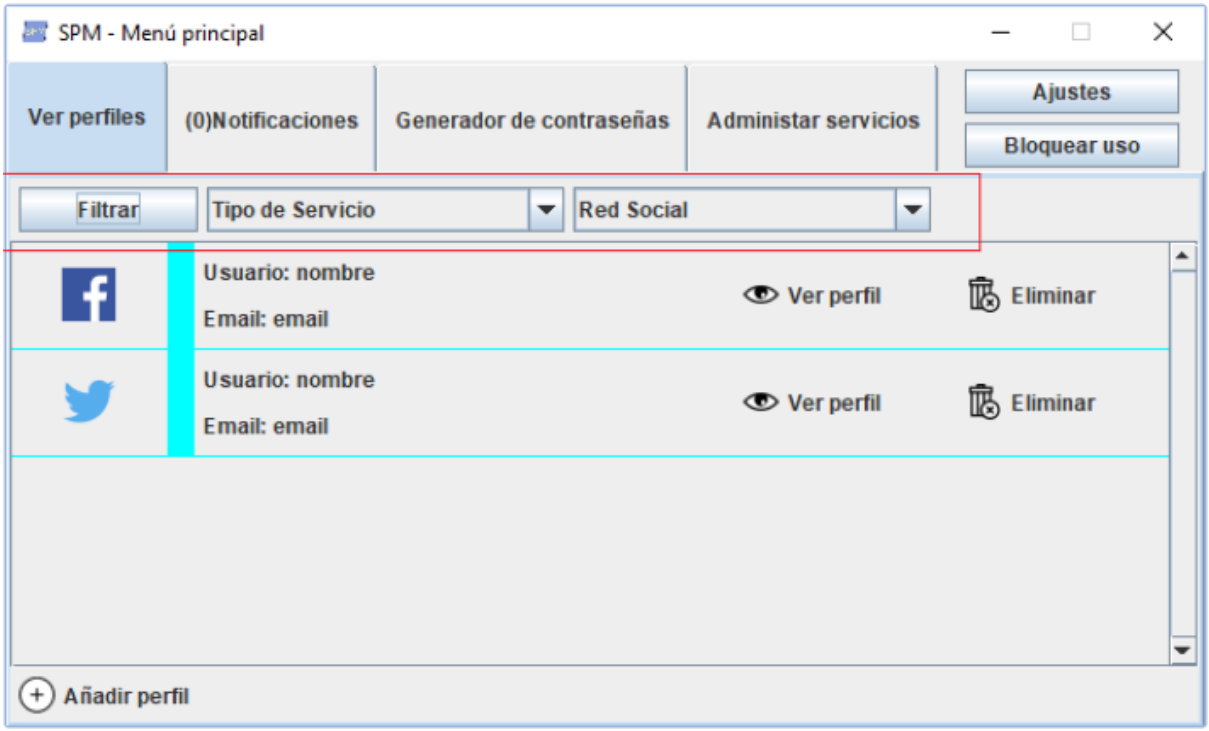


Figura 74: Menú principal, ver perfiles (6)

Para volver a ver la lista completa de debe filtrar con seleccionar tipo de filtro, como tipo de filtro.

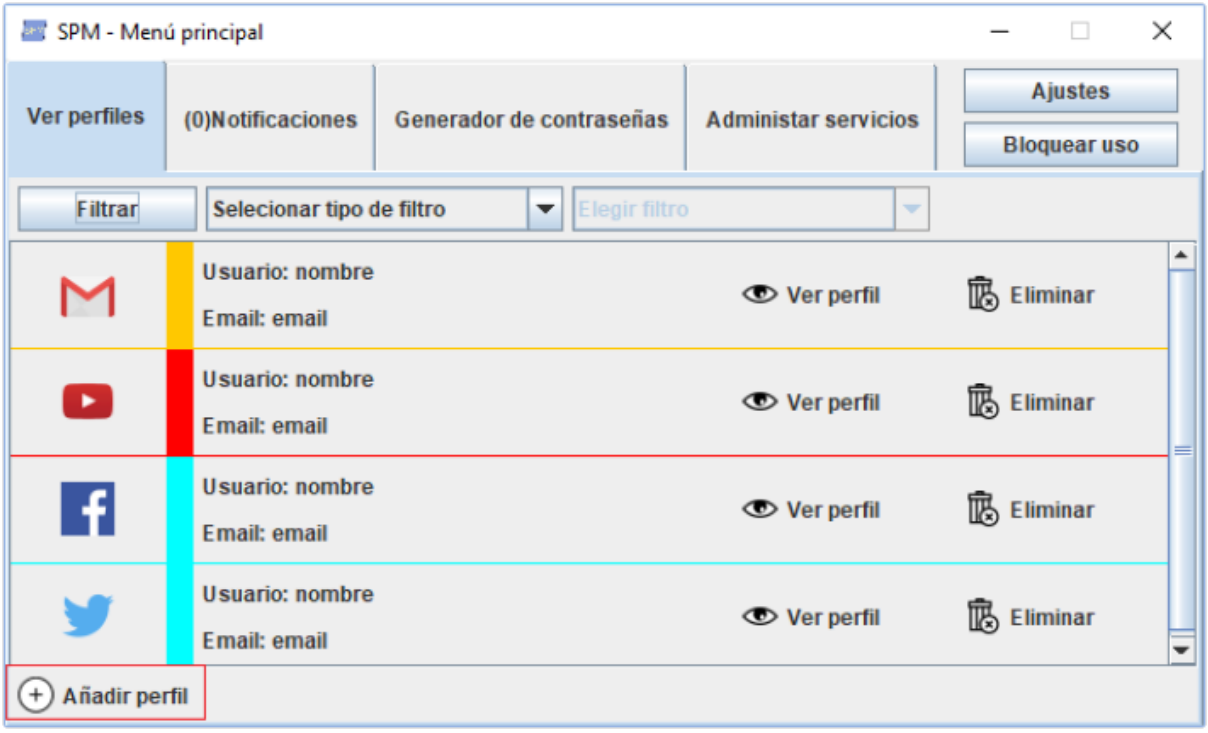


Figura 75: Menú principal, ver perfiles (7)

Para añadir un nuevo perfil se puede utilizar el botón “Añadir perfil” situado en la esquina inferior izquierda.

Menú principal (Administrar servicios)

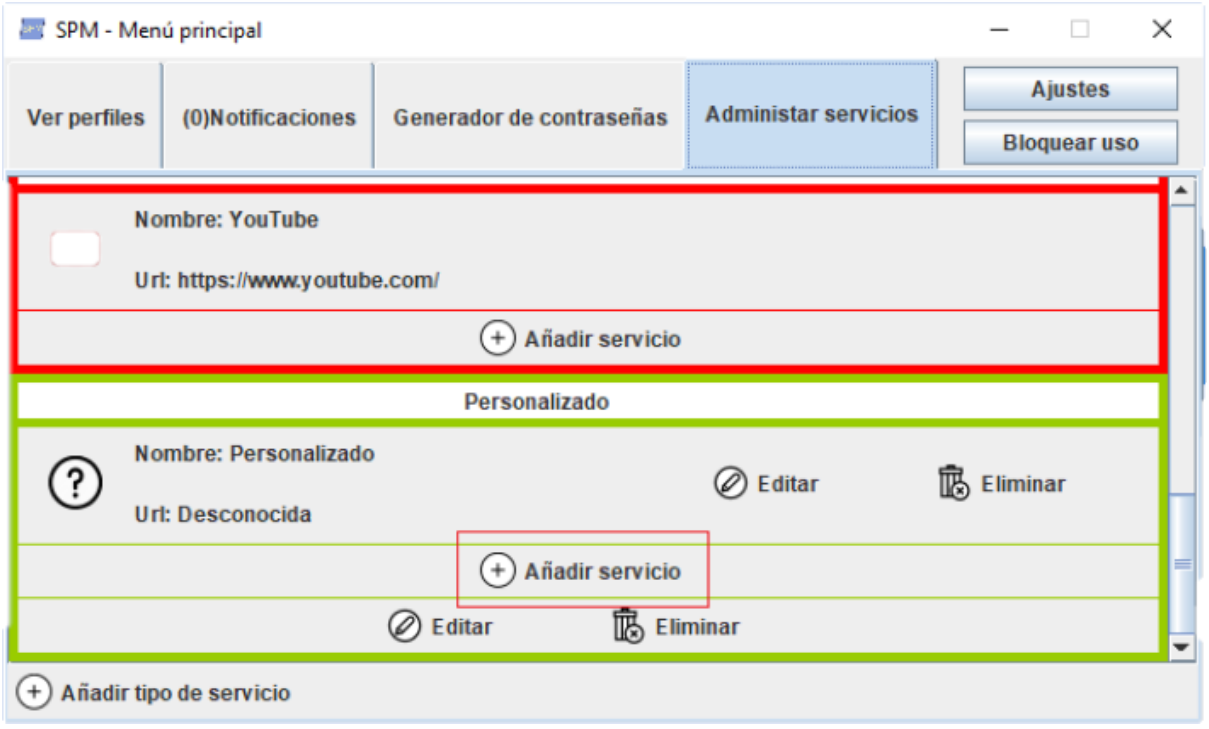


Figura 76: Menú principal, administrar perfiles (1)

El botón “Añadir sevicio” abre la pantalla de creación de servicios, con el tipo de servicio seleccionado al tipo de servicio cuyo botón “Añadir servicio ha sido usado.

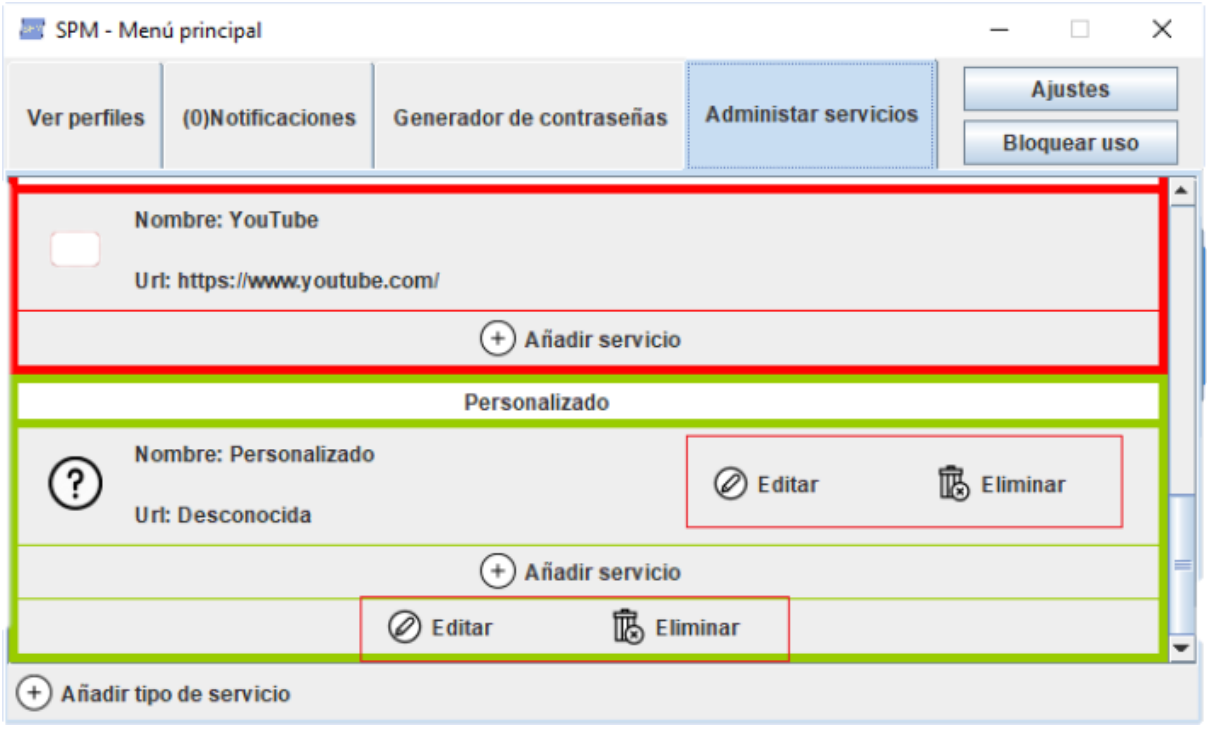


Figura 77: Menú principal, administrar perfiles (2)

De manera similar los botones “Editar” y “Eliminar” permite editar o eliminar el servicio o tipo de servicio con el que están relacionados. Los tipos de servicio y servicios por defecto no se pueden eliminar o modificar.

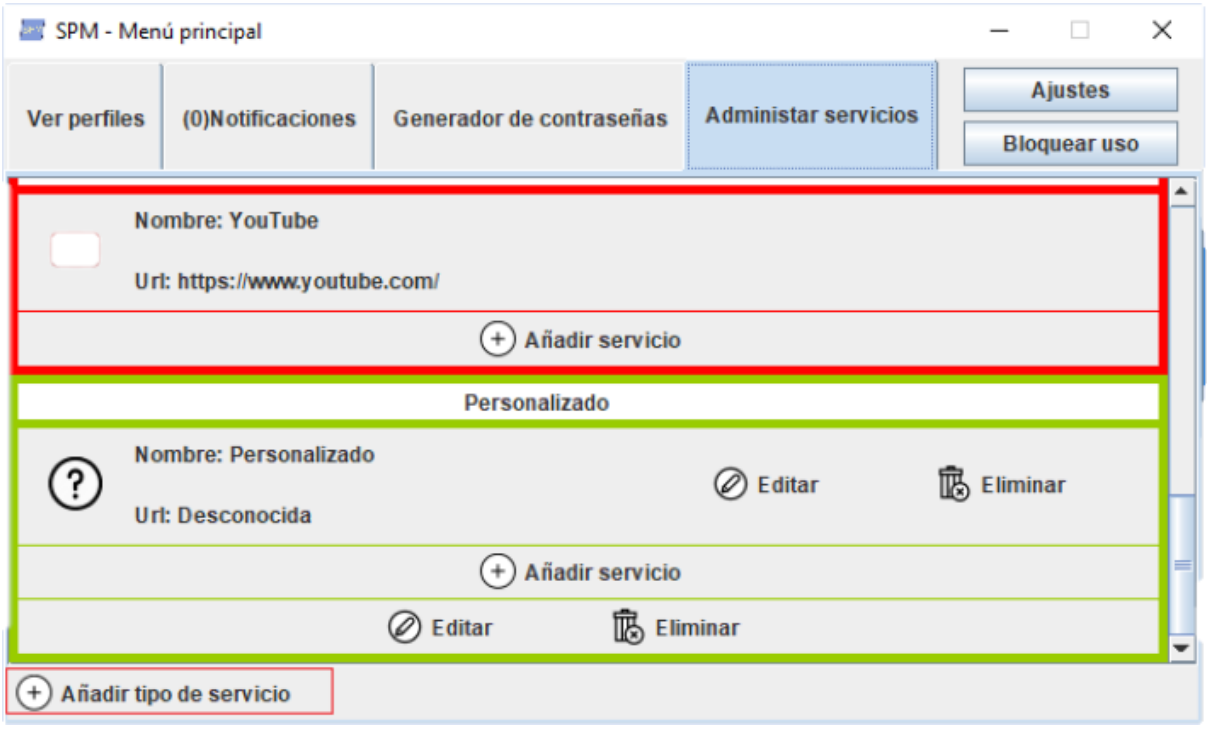


Figura 78: Menú principal, administrar perfiles (3)

Por último para añadir un nuevo tipo de servicio se ha de usar el botón “Añadir tipo de servicio” situado en la esquina inferior izquierda.

Ajustes

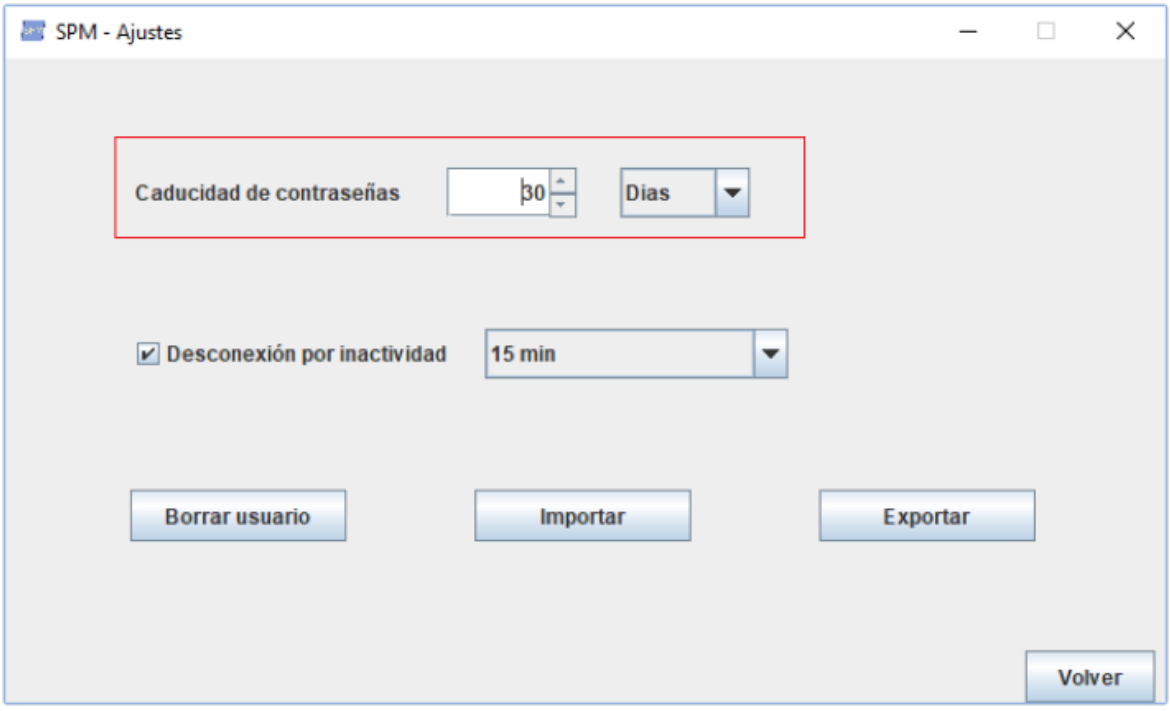


Figura 79: Ajustes (1)

La caducidad de contraseñas establece el tiempo en el cual caduca una contraseña y por lo tanto la aplicación notificara a un usuario que está debe ser cambiada, desde que se crea un perfil con esa contraseña, o se cambia la contraseña de un perfil ya existente.

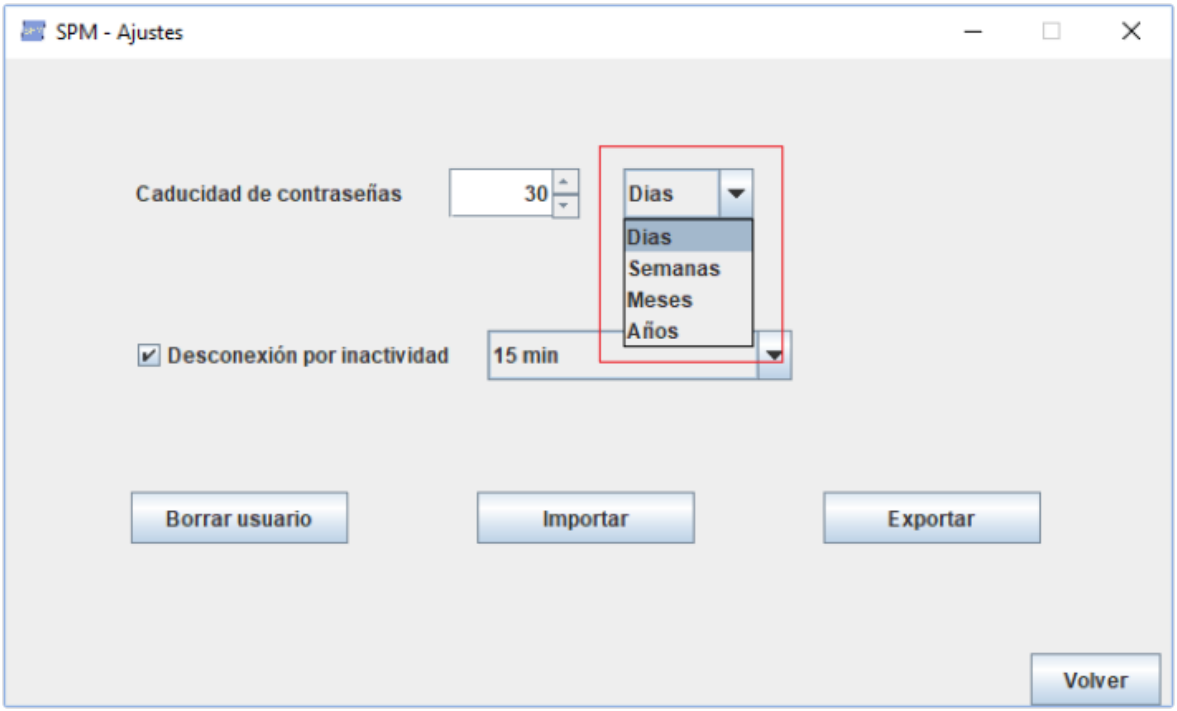


Figura 80: Ajustes (2)

Se puede modificar la cantidad y la unidad a utilizar.

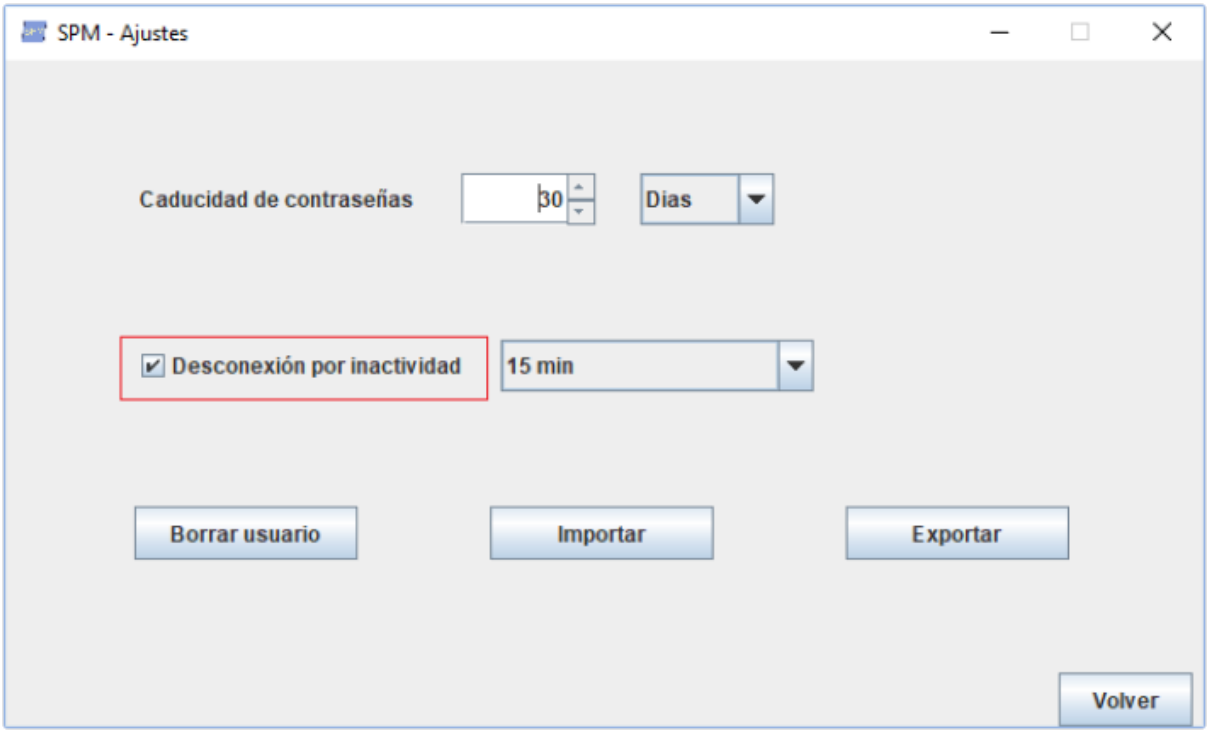


Figura 81: Ajustes (3)

El checkbox permite activar y desactivar la desconexión por inactividad.

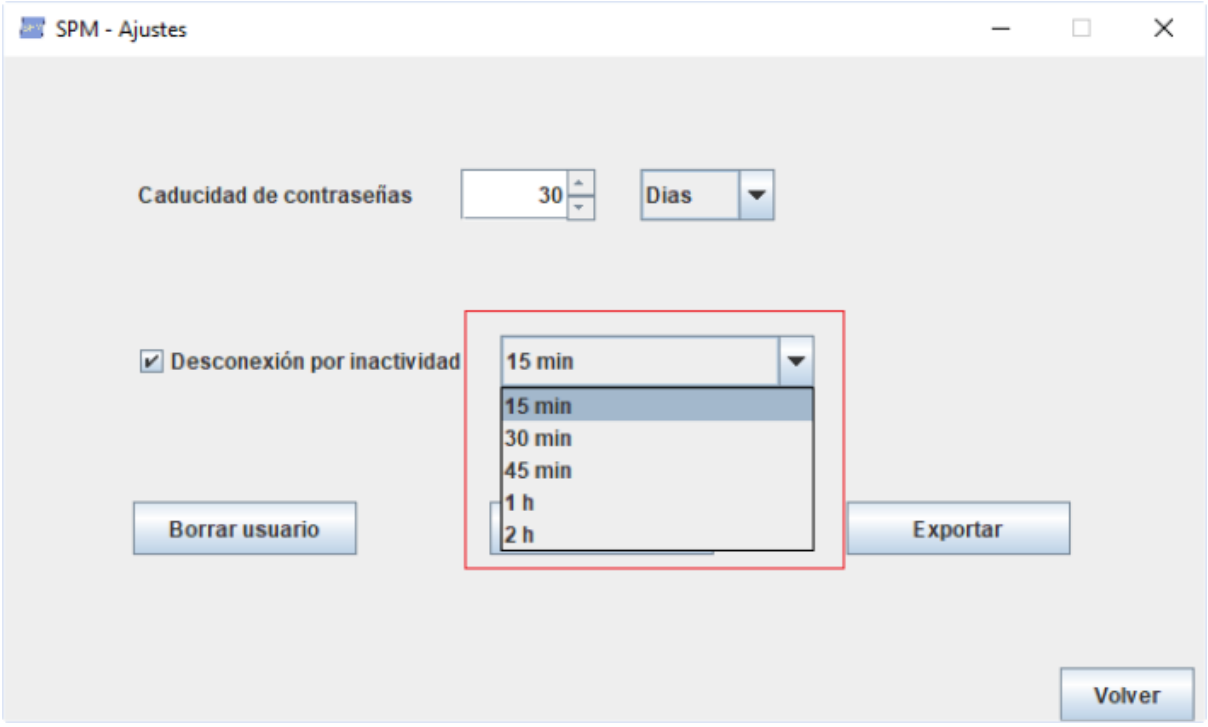


Figura 82: Ajustes (4)

Por su lado el dropbox adyacente permite elegir la cantidad de tiempo tras la cual se producirá la desconexión.

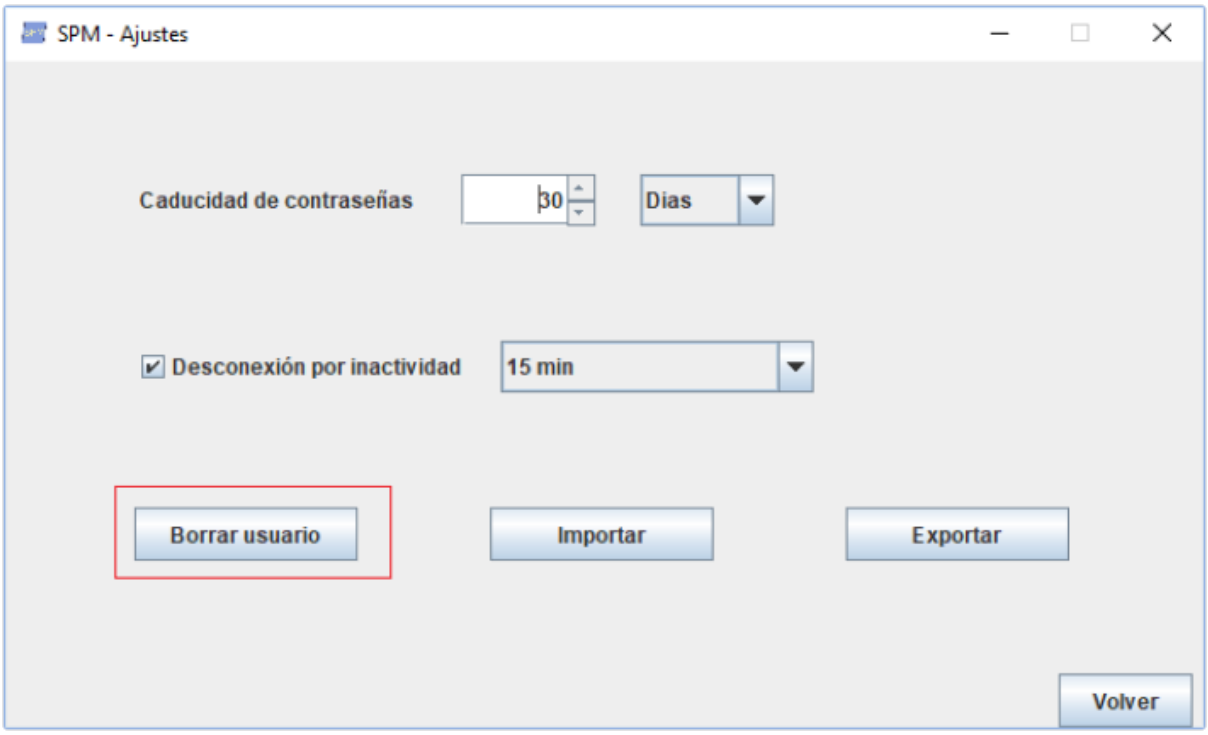


Figura 83: Ajustes (5)

El botón borrar usuario borrara la información almacenada en la aplicación de manera irreversible, antes de completar la operación mostrará un diálogo de confirmación.

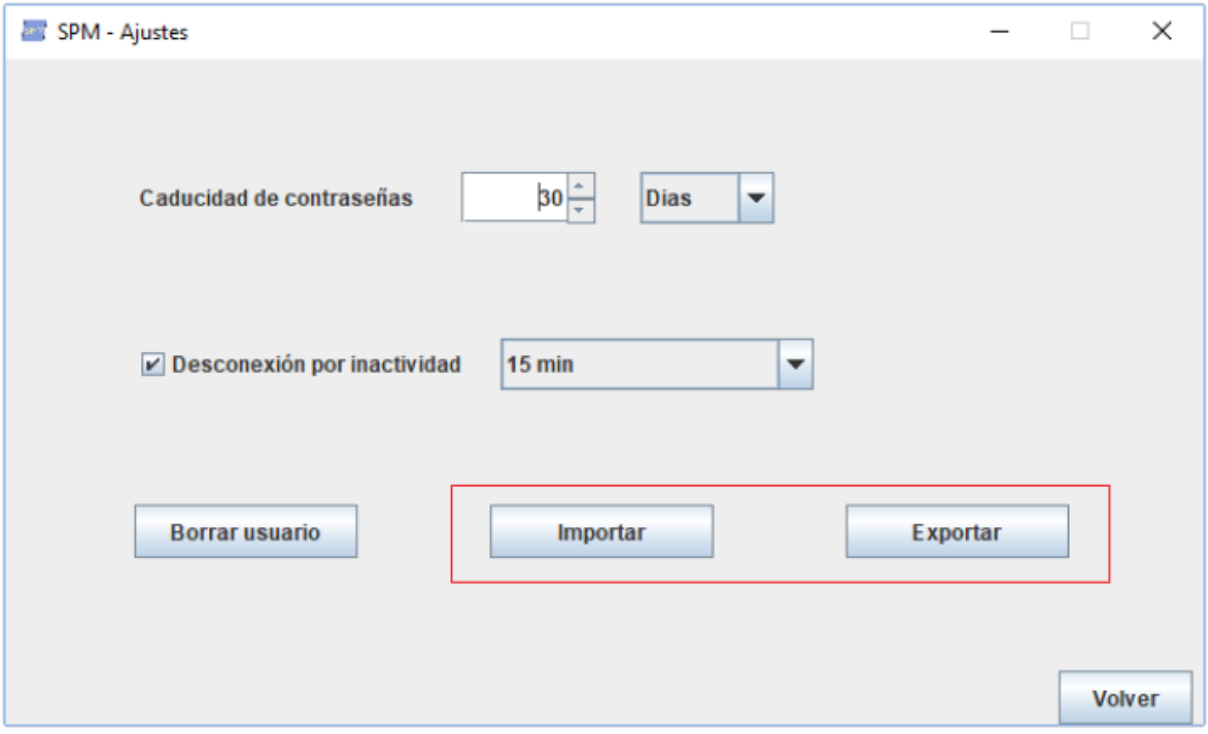


Figura 84: Ajustes (6)

Los botones “Importar” y “exportar” dan acceso a las funcionalidades homónimas de la aplicación.

7.2.2 Paso a paso

A continuación se explican paso a paso las principales funcionalidades de la aplicación.

Crear usuario

Figura 85: Crear Usuario (1)

Desde el menú de seleccione el botón “Crear Usuario”.

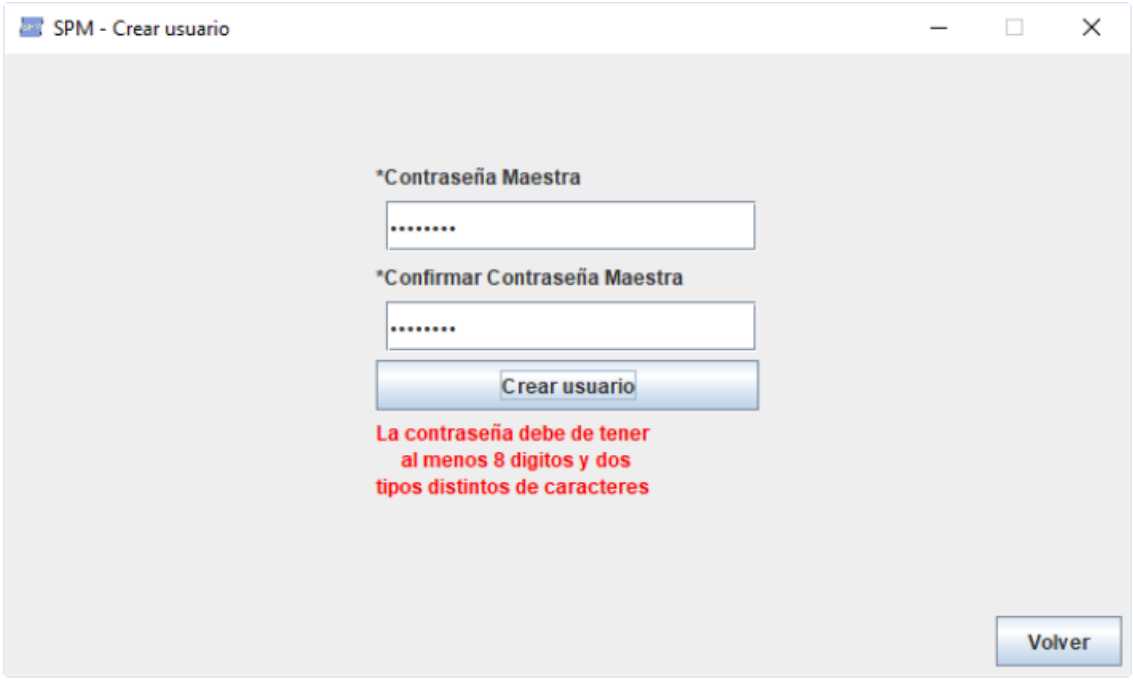


Figura 86: Crear Usuario (2)

Una vez introducida una contraseña válida (al menos 8 dígitos y dos tipos de caracteres distintos entre, mayusculas, minusculas, numeros o caracteres especiales), pulse el botón “Crear usuario”.

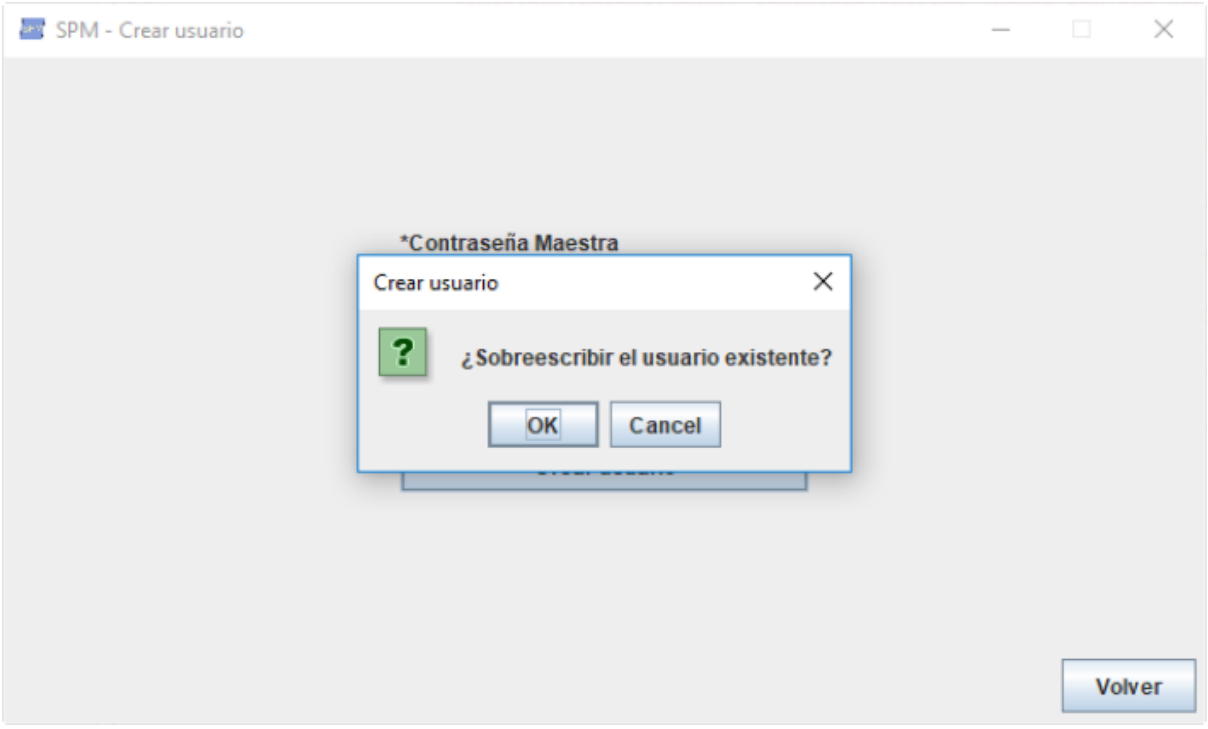


Figura 87 Crear Usuario (3)

En caso de ya existir un usuario se mostrará un diálogo para confirmar la eliminación del usuario anterior y toda su información.

Ha creado un usuario con éxito.

Crear perfil

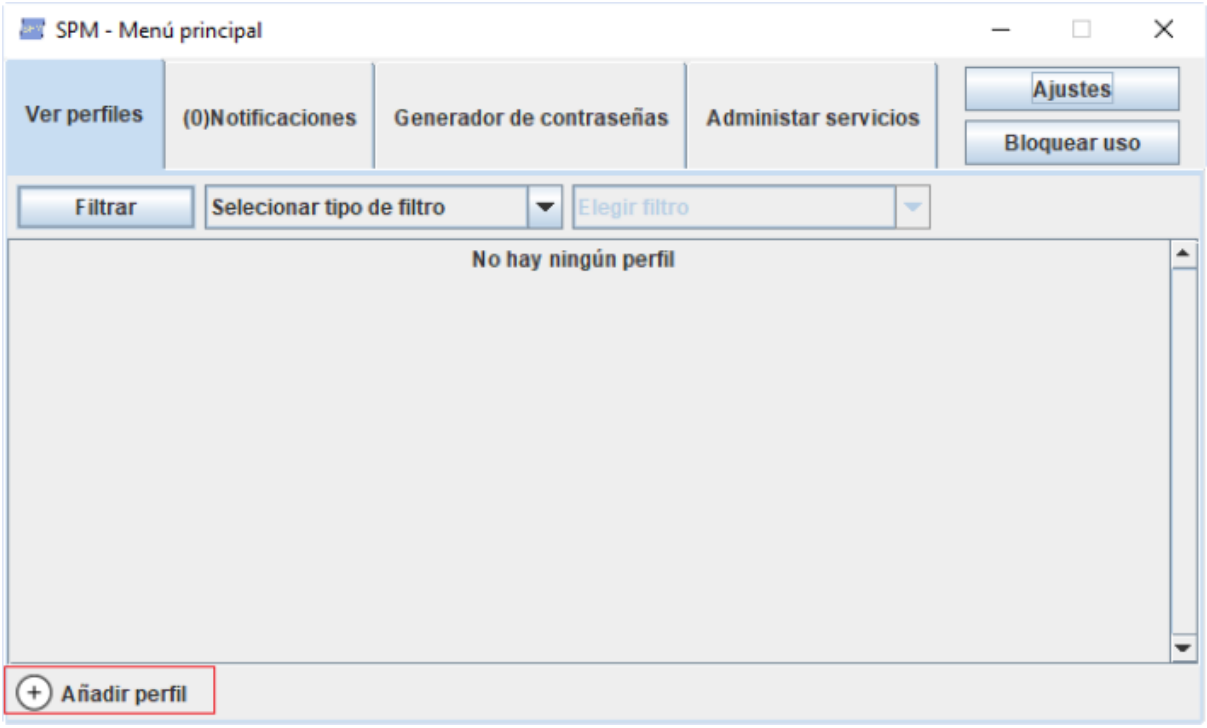


Figura 88: Crear Perfil (1)

Como usuario identificado desde el menú principal, en la pestaña Ver perfiles, pulse el botón Añadir perfil.

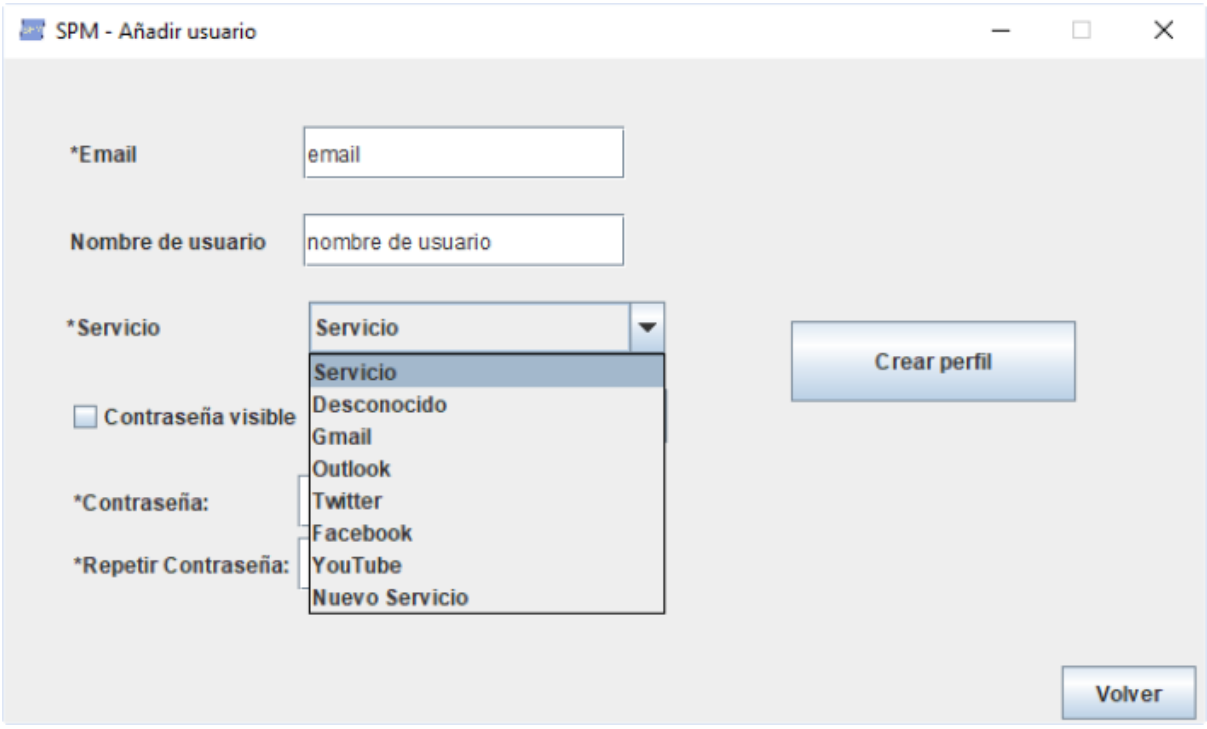


Figura 89: Crear Perfil (2)

Rellene los campos con la información del nuevo perfil, si selecciona como servicio la opción Nuevo servicio, se iniciara el proceso de creación de servicio tras pulsar al botón crear perfil.

Figura 90: Crear Perfil (3)

Puede generar una contraseña automáticamente mediante el botón Generar con la configuración actual del generador de contraseñas. Puede configurar la configuración de este pulsando al botón Generador de Contraseñas, para abrir el Generador de contraseñas completo.

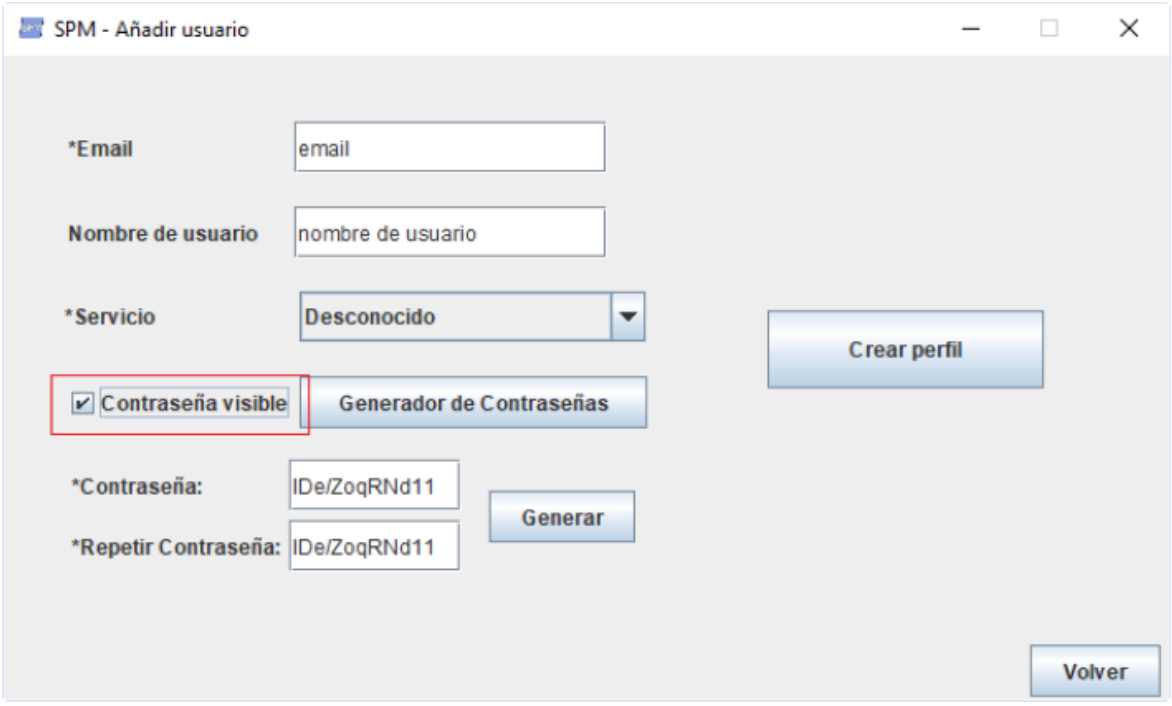


Figura 91: Crear Perfil (4)

El checkbox permite mostrar/ocultar las contraseñas.

Editar perfil

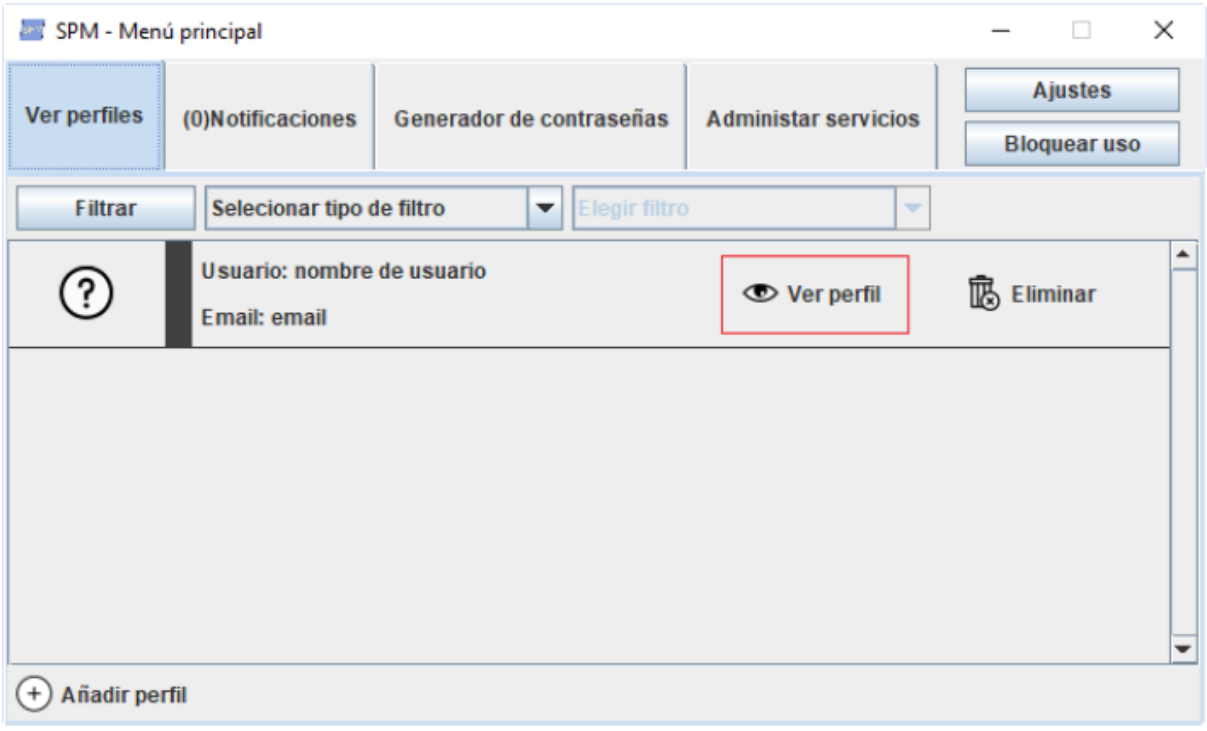


Figura 92: Editar perfil(1)

Desde el menú principal como usuario identificado pulse el botón ver perfil de un perfil existente.

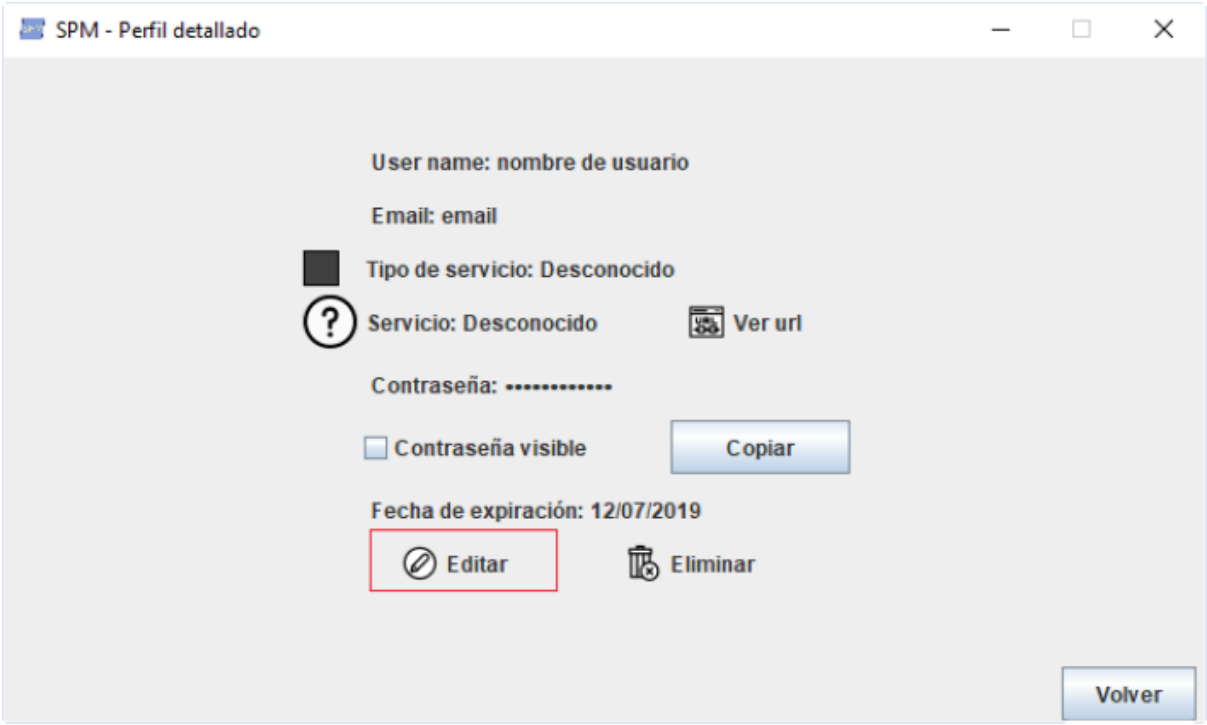


Figura 93: Editar perfil(2)

Desde la visualización del perfil en detalle pulse el botón Editar.

Figura 94: Editar perfil(3)

Puede modificar todos los campos de manera directa excepto la contraseña, incluido la fecha de caducidad mediante un datepicker.

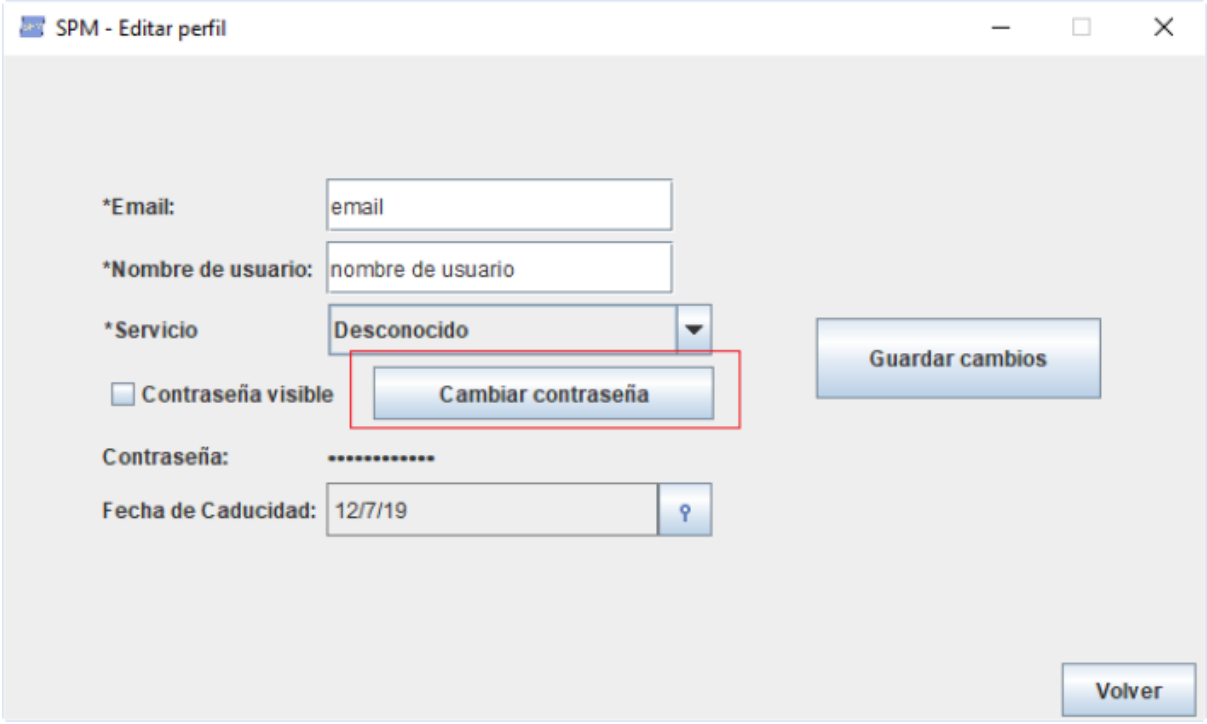


Figura 95: Editar perfil(4)

Para cambiar la contraseña es necesario acceder a una ventana adicional con el botón cambiar contraseña.

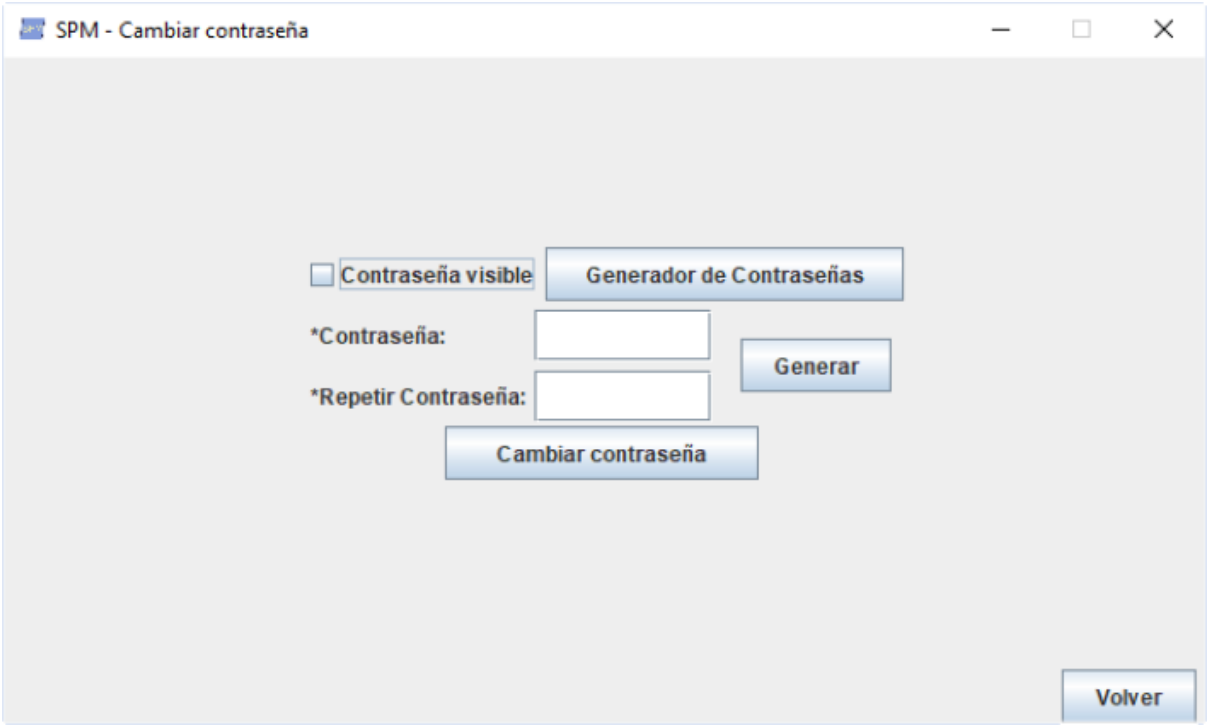


Figura 96: Editar perfil(5)

Desde la pantalla cambiar contraseña puede cambiar la contraseña con un funcionamiento similar a los componentes para la creación de perfiles.

Figura 97: Editar perfil(6)

Una vez realizadas las modificaciones pulse al botón Guardar cambios para guardar los cambios. En caso de pulsar a volver se descartaran todos los cambios realizados.

Crear servicio

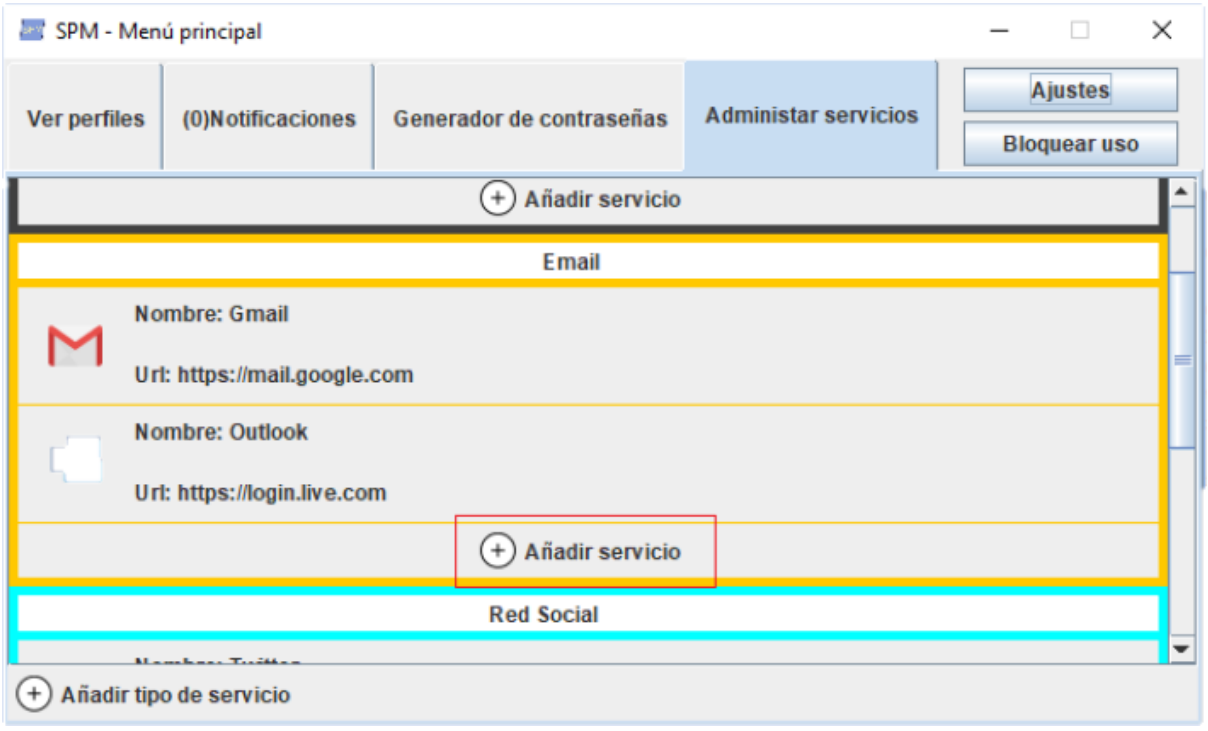


Figura 98: Crear servicio(1)

Desde el menú principal, como usuario identificado en la pestaña Administrar Servicios, pulse Añadir servicio en el tipo de servicio correpondiente.

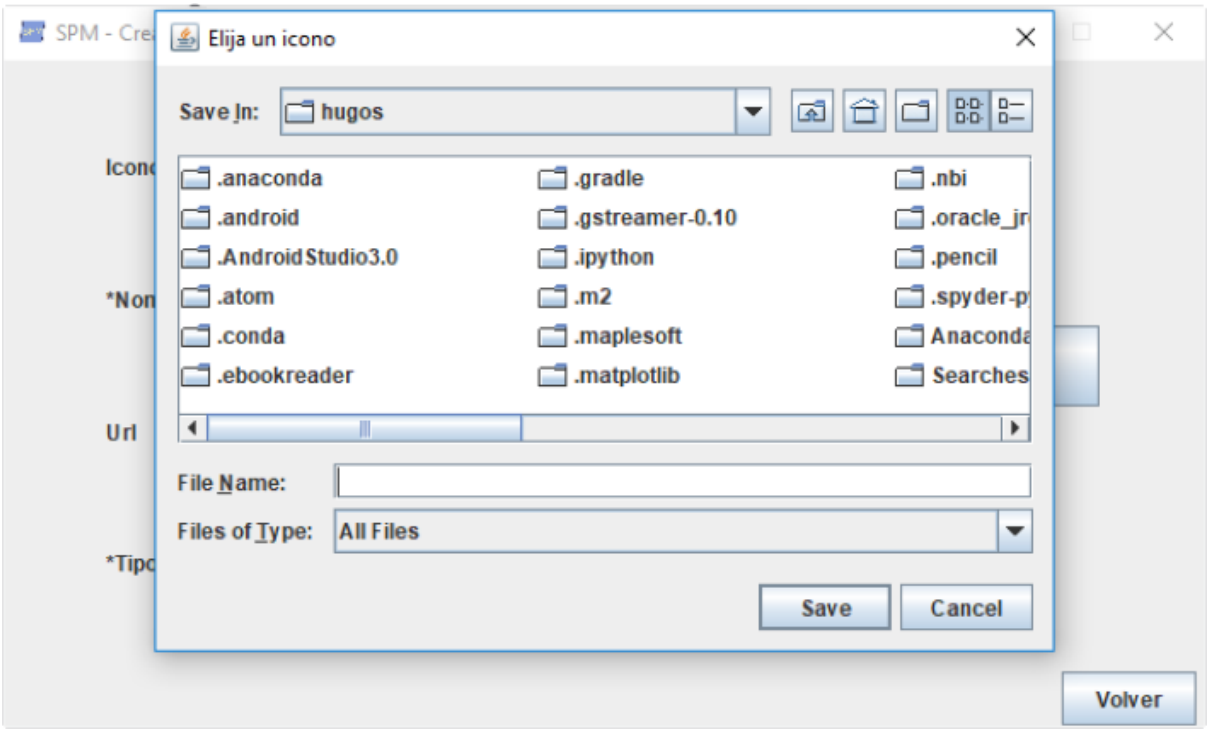


Figura 99: Crear servicio(2)

Puede utilizar el botón cargar icono para escoger un icono para el nuevo tipo de servicio. En caso de no elegir uno se utilizará el icono por defecto.

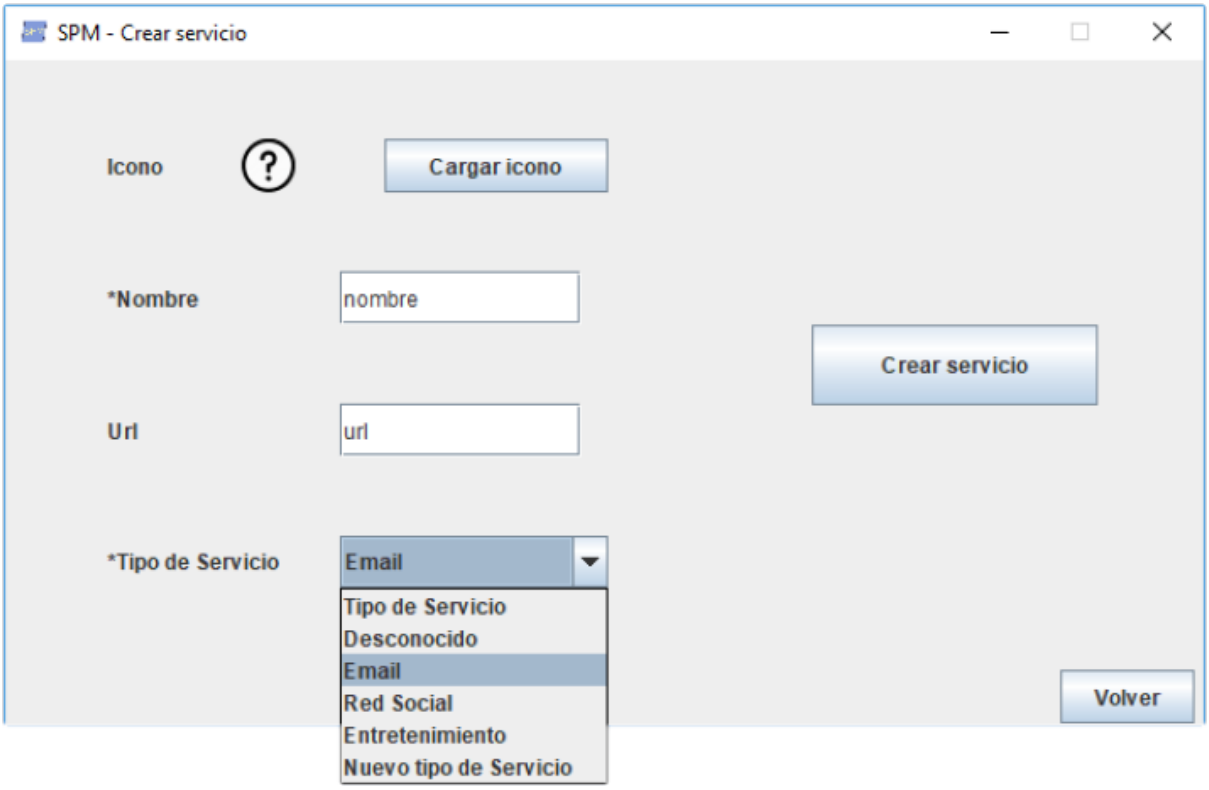


Figura 100: Crear servicio(3)

Utilice el dropdown para elegir el tipo de servicio. La opción nuevo tipo de servicio abrirá la pantalla de creación de tipos de servicio antes de completar la creación del servicio.

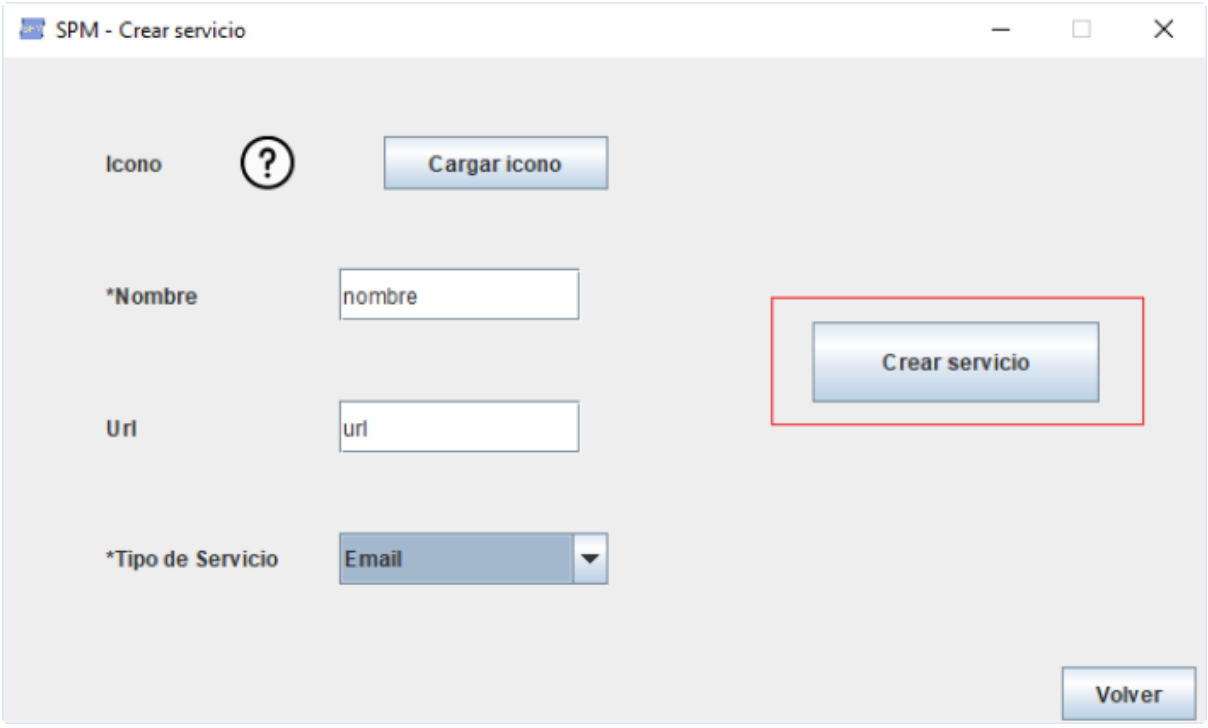


Figura 101: Crear servicio(4)

Una vez rellenados todos los campos obligatorios puede crear el nuevo servicio con el botón Crear servicio.

Crear tipo de servicio

Figura 102: Crear tipo de servicio(1)

Desde el menú principal, como usuario identificado en la pestaña Administrar servicios, pulse el botón añadir tipo de servicio.

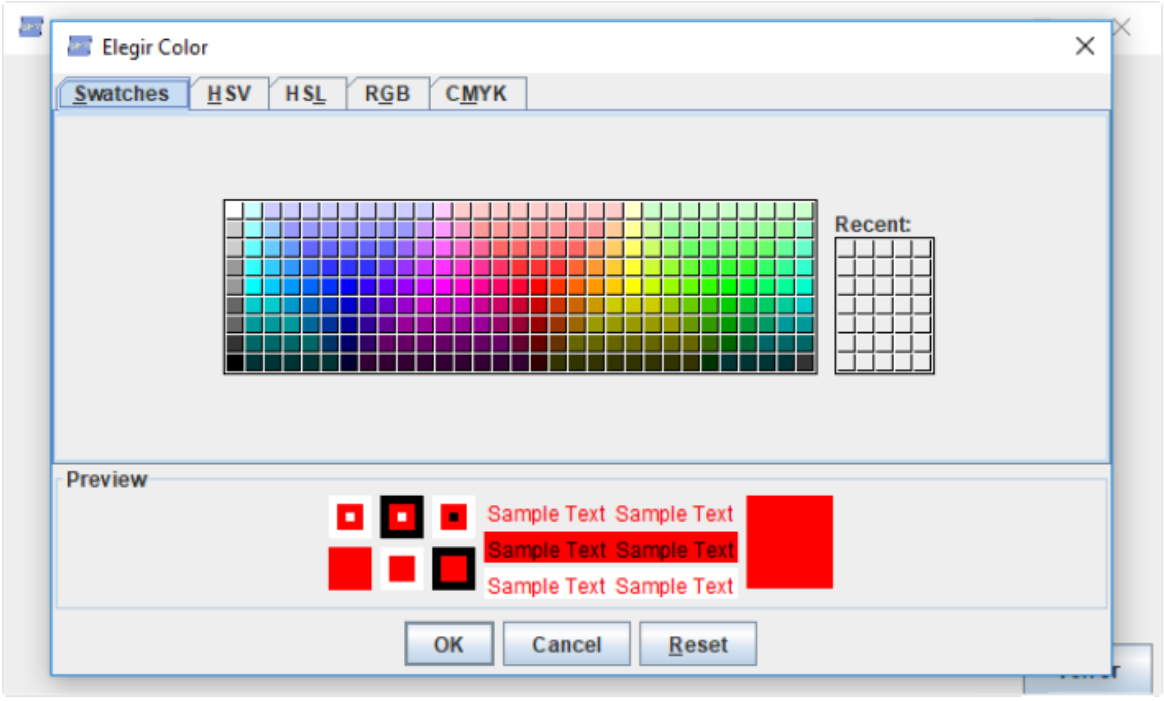


Figura 103: Crear tipo de servicio(2)

Puede utilizar el botón elegir color para elegir el color del tipo de servicio.

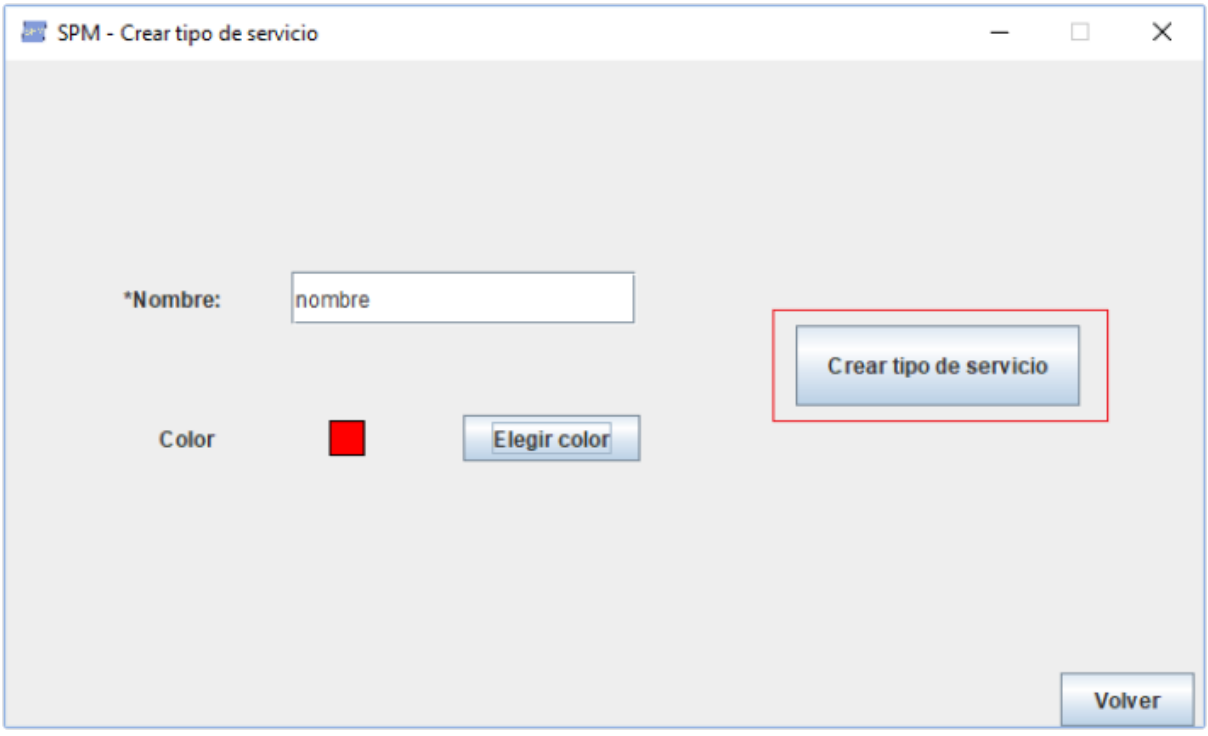


Figura 104: Crear tipo de servicio(3)

Una vez rellenados los campos se puede crear el tipo de servicio pulsando el botón Crear tipo de servicio.

Importar

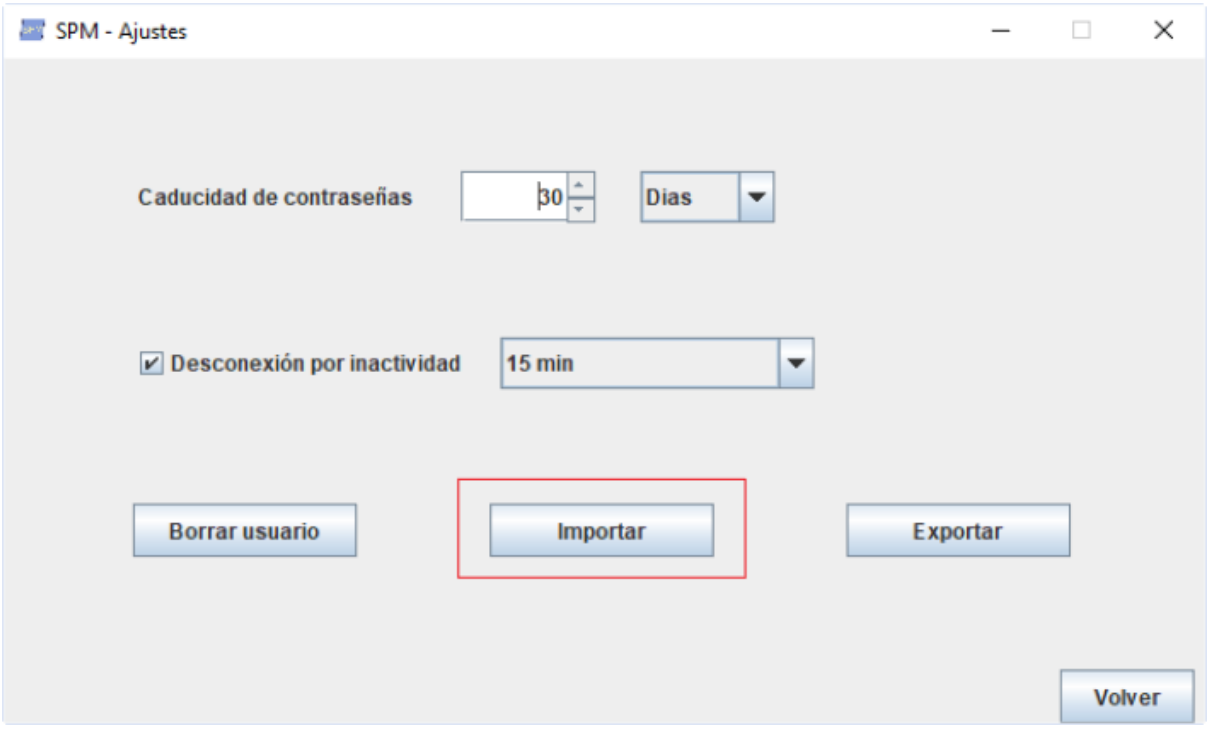


Figura 105: Importar (1)

Desde la ventana de ajustes, pulsa el botón importar.

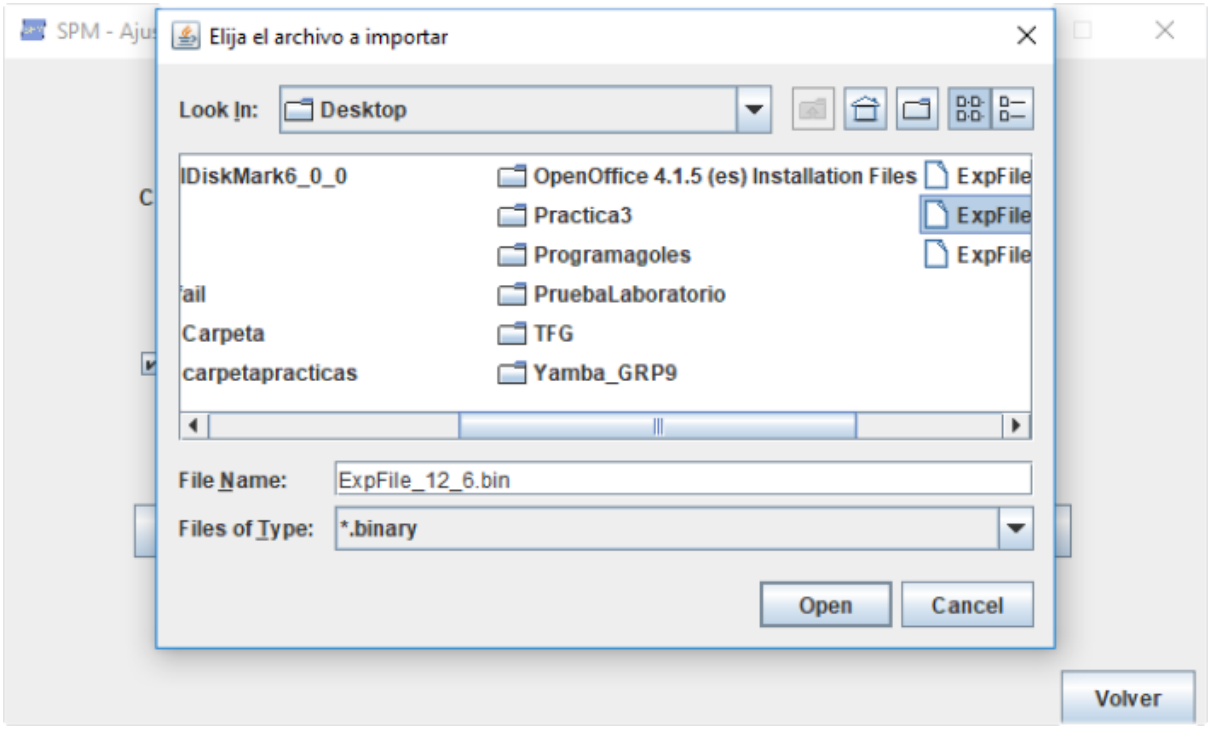


Figura 106: Importar (2)

Selecciona el archivo a importar.

Figura 107: Importar (3)

El campo contraseña está inhabilitado si la contraseña del archivo y la de la aplicación coinciden, si no lo está introduzca la contraseña del archivo. Puede utilizar el checkbox para elegir la política de escritura.

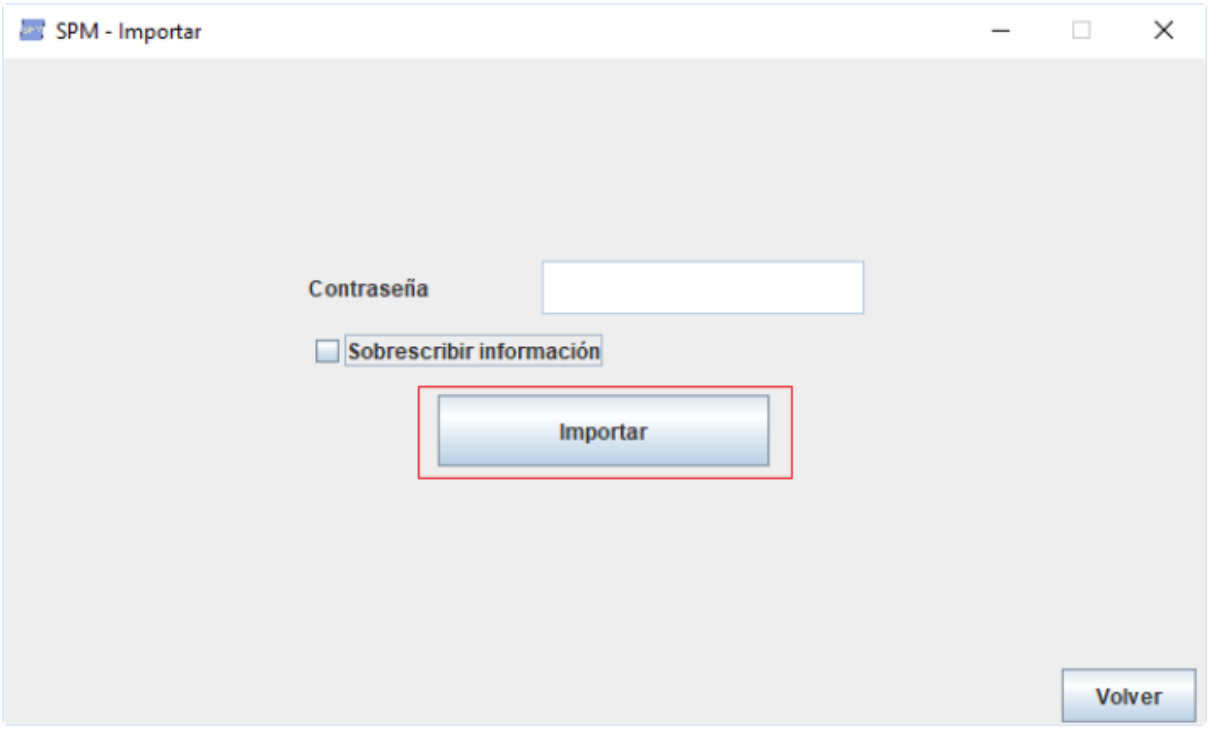


Figura 108: Importar (4)

Una vez elegida la política he introducido la contraseña de ser necesario, pulse el botón Importar para completar el proceso.

Exportar

Figura 109: Exportar(1)

Desde el menú de ajustes, pulse el botón Exportar.

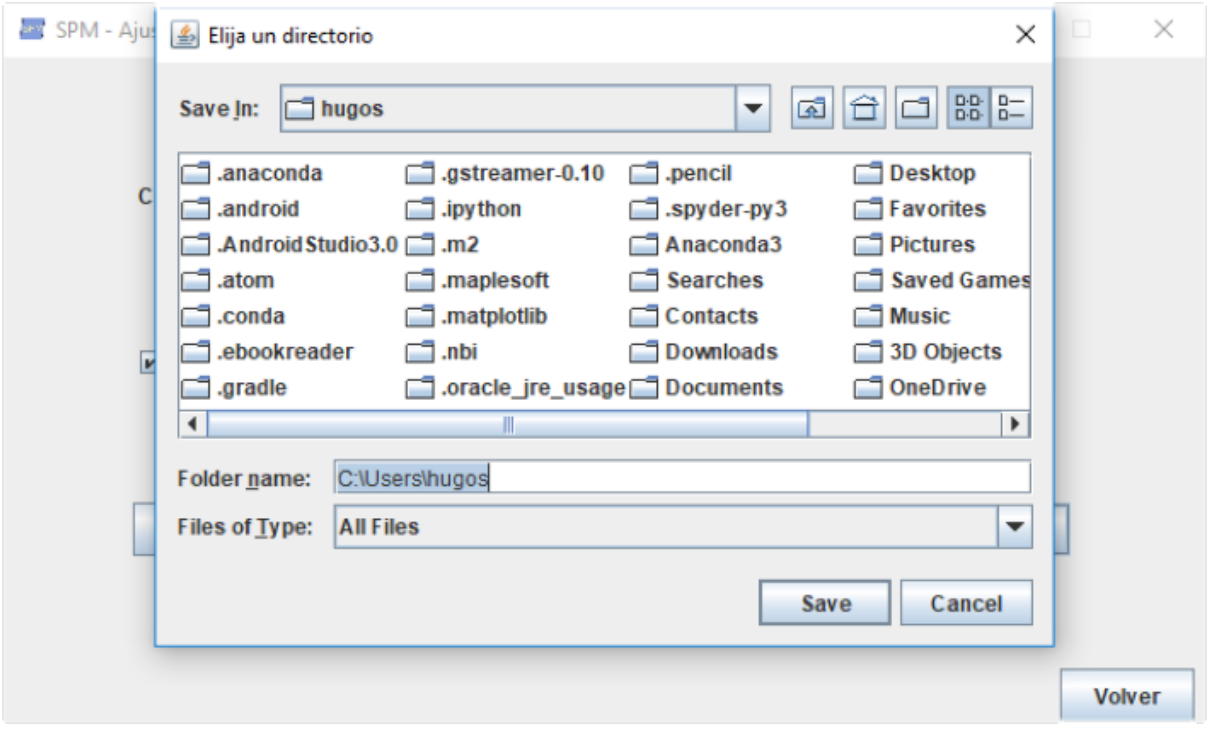


Figura 110: Exportar(2)

Seleccione el directorio donde guardar el archivo para completar el proceso.

Parte IV

Conclusiones

Capítulo 8

Conclusiones

8.1 Conclusiones

El resultado de este proyecto es una aplicación funcional que permite la gestión de contraseñas de manera suficiente, pero que dista mucho de la idea original que motivó este proyecto. El contexto del trabajo limitado a las 300 horas ha supuesto un esfuerzo de contención para el que han sido necesarias una estimación y planificación adecuadas unidas a un control autónomo de las horas de trabajo, además de un importante trabajo de priorización de objetivos para no extender el alcance fuera de los límites viables dentro del tiempo establecido.

Como elemento académico, este TFG ha supuesto una unión de los conocimientos adquiridos en la carrera de manera directo o indirecta. Siendo destacable el caso de los contenidos relacionados con la ingeniería de software que por primera vez se han realizado junto con una implementación completa de los elementos modelados, realzando su necesidad e importancia.

Como aspectos a mejorar para futuras ocasiones cobra relevancia la gestión preventiva de riesgos, dado que aunque ha sido posible solventar los problemas acaecidos durante el proyecto de manera reactiva, en proyectos de mayor envergadura o con un límite de tiempo más estricto es poco probable que sea posible solventarlos de esta manera.

Por último me parece importante comentar el salto de perspectiva con respecto al resto de trabajos realizados hasta ahora, al ser este TFG un proyecto orientado a un público más general, alejándose de un público especializado al que se presenta un trabajo con unas cotas predefinidas.

8.2 Futuras mejoras

En primer lugar sería necesario continuar con el desarrollo de la idea original de este proyecto y completar las iteraciones planteadas en el momento inicial y que quedaron fuera por el límite que impone el contexto de este proyecto.

Yendo a puntos más concretos con respecto a lo implementado, hay principalmente dos aspectos a mejorar. Siendo el primero de estos aspectos la exportación importación de elementos más acotados como solo los Servicios, o solo los Perfiles, en lugar de la información completa, siendo la evolución completa el poder elegir los elementos a exportar

de manera singular, perfil a perfil. Y el segundo la adaptación de la interfaz implementada a un diseño responsive que permitiera ajustarse al tamaño del dispositivo.

Como añadido en lugar de como mejora, sería interesante habilitar al usuario de una manera para cambiar la contraseña maestra de manera directa. Así como permitir que las notificaciones puedan ser lanzadas desde el programa en segundo plano ligadas al sistema operativo para que no fuera necesario acceder a la aplicación para recibir esas notificaciones.

Parte V

Anexos

A. Glosario de términos y métodos

■Actor: toda entidad humana o no que tiene relación con la aplicación o sistema y hace uso de alguna de sus funcionalidades.

■AES: sistema de cifrado simétrico (que utiliza la misma clave para cifrar y descifrar) por bloques (que cifra un número limitado información, y repite el mismo algoritmo para el resto de la información en fragmentos de la longitud máxima posible, aunque se suelen utilizar cuando el número de bloques de información a codificar no es muy grande, como contraseñas en lugar de mensajes completos)

■Árbol de características: diagrama en forma de árbol que muestra las características principales de un programa o sistema agrupandolas en ramas relacionadas.

■Arquitectura lógica: modelado a nivel conceptual de una aplicación o sistema.

■Arquitectura física: elementos de hardware que utiliza la aplicación o sistema para llevar a cabo su función.

■Ataque de fuerza bruta: ataque criptográfico basado en probar todas las combinaciones posibles.

■Chacha 20: sistema de cifrado simétrico en flujo (que cifra un número ilimitado de información usando generalmente el cifrado de los primeros elementos para cifrar los siguientes.

■CRUD: es el acrónimo de "Crear, Leer, Actualizar y Borrar" (Create, Read, Update and Delete), que se usa para referirse a las funciones básicas en bases de datos o la capa de persistencia en un software.

■Diagrama de casos de uso: es un diagrama UML que define las acciones que pueden realizar los actores y la relación entre las acciones.

Los “monigotes” representas a los actores que realizan los casos de uso, los óvalos representan a los casos de uso que describen con su texto, y las flechas determinan el tipo de relación.

Las flechas directas entre actores y casos de uso significan que un actor puede realizar ese caso de uso. Existen dos tipos de flechas entre casos de uso:Include y extends. Include implica que siempre que sucede un caso de uso en el extremo de la fecha, sucede el caso de uso que hay al final de la flecha. Extens significa que el caso de uso del que parte la flecha puede suceder o no si sucede el caso de uso al que apunta la flecha.

■Diagrama de clases: es un diagrama UML que define los objetos que conforman una aplicación y la relación entre ellos. Cada objeto está separado en atributos (“características”) y métodos (“acciones”). El símbolo delante de objetos y métodos define la visibilidad pudiendo ser esta publica (+ , todos los objetos pueden verlo), privada (-,ningún objeto puede verlo) o protegida (#, solo objetos dentro de la jerarquía de herencia pueden verlo).
Las relaciones entre objetos pueden ser de tres tipos, asociación (existe algún tipo de relación), agregación (un objeto forma parte de otro, la parte puede existir de forma independiente) o composición(la parte no puede existir de forma independiente).

■Diagrama de Gantt:es un diagrama que expone las actividades a realizar en un proyecto frente al tiempo, de manera que la barra de tiempo representa un periodo de tiempo continuo y las tareas se muestran de manera cronológica sobre esa barra mostrando cuando empiezan y cuando terminan, y que acciones se están realizando simultáneamente.

■Java: lenguaje de programación orientado a objetos. Su punto fuerte es que los programas en java corren sobre una máquina virtual, lo que permite que se pueden usar en diferentes dispositivos independientemente del sistema operativo.

■Pruebas de caja blanca: tipo de prueba ligada al código, destinada a comprobar el flujo de ejecución de un programa cerciorándose de que el valor de las distintas variables a lo largo de la ejecución del programa para distintas entradas sea el correcto.

■Pruebas de caja negra: tipo de prueba destinado a comprobar que la respuesta de un programa frente a una entrada es la salida, sin importar que pase en el programa para obtener dicha salida.

■SHA: algoritmo de HASH. Los algoritmos de HASH o de resumen obtienen una cadena única a partir de una entrada de manera que la misma entrada produce la misma salida, pero es imposible obtener la entrada a partir de la salida.

■Requisitos de usuario: mediante lenguaje natural o diagramas, los requisitos de usuario describen las operaciones que un sistema debe poder realizar.

■Requisitos no funcionales: requisitos de usuario que describen restricciones en lugar de acciones directas.

B. Requisitos denegados

En este apartado se recogen diversos requisitos o versiones de requisitos que han sido modificados o retirados y que no aparecen en la documentación técnica.

Requisito
NFS-02
Descripción
Se deben almacenar los 128 primeros bits de la salida al algoritmo SHA-384 de la contraseña maestra para identificar la contraseña maestra.
Motivo retirada
Sustituido por NSF-05, sustituido para evitar la posibilidad de colisiones no detectadas al utilizar todos los bits de la salida en lugar de solo un fragmento.

Tabla 116:Requisito denegado NFS-02

Requisito
US-01

Descripción		
US-01	Establecer contraseña Maestra	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-01	
Requisitos asociados	RU-01	
Descripción	El sistema solicita al usuario una contraseña a partir de la cual se generará la clave para codificar la información.	
Precondición	No está establecida una contraseña maestra.	
Secuencia normal	Paso	Acción
	1	El usuario accede a la aplicación por primera vez.
	2	La aplicación notifica que la pérdida de la contraseña maestra supone la pérdida de la información codificada.
	3	La aplicación solicita una contraseña maestra.
	4	El usuario introduce una contraseña.
	5	La aplicación aplica el algoritmo SHA-384 sobre la contraseña introducida.
	6	La aplicación guarda los 128 primeros bits de la salida del algoritmo SHA-384 como medio para validar la contraseña maestra en futuras ocasiones.
	7	La aplicación carga en la memoria temporal del programa los 256 últimos bits de la salida de SHA-384 para usarlos como clave en la codificación con AES-256.
	8	La aplicación notifica al usuario que el proceso se ha completado.
	9	La aplicación muestral el menú principal.
	10	El caso de uso finaliza con éxito.
Postcondición	La aplicación está en el menú principal, lista para codificar información.	
Excepciones	Paso	Acción
Comentarios		

Motivo retirada

Actualizado a la versión 1.1, actualizado para evitar la posibilidad de colisiones no detectadas al utilizar todos los bits de la salida en lugar de solo un fragmento.

Tabla 117:Requisito denegado US-01

Requisito
US-02

Descripción		
US-02	Validar la contraseña maestra	
Versión	1.0	
Autor	Hugo Gilarranz	
Objetivos asociados	OBJ-01	
Requisitos asociados	RU-02	
Descripción	Un usuario podrá introducir la contraseña maestra para verificar que es un usuario legítimo.	
Precondición	Se ha establecido una contraseña maestra y el usuario se encuentra en el menú de inicio.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la opción introducir contraseña.
	2	La aplicación solicita la contraseña.
	3	El usuario introduce la contraseña.
	4	La aplicación aplica el algoritmo SHA-384 a la contraseña.
	5	La aplicación utiliza los 128 primeros bits para validar la contraseña.
	6	La aplicación carga en memoria los 256 últimos bits para usarlos como clave en la codificación con AES-256.
	7	La aplicación muestral el menú principal.
	8	El caso de uso finaliza con éxito.
Postcondición	La aplicación está en el menú principal, lista para codificar información.	
Excepciones	Paso	Acción
	6 b	La aplicación notifica al usuario de que la contraseña es incorrecta y se vuelve al paso 2.
Comentarios		

Motivo retirada
Actualizado a la versión 1.1, actualizado para evitar la posibilidad de colisiones no detectadas al utilizar todos los bits de la salida en lugar de solo un fragmento.

Tabla 118:Requisito denegado US-02

C. Contenido del CD

El CD contiene los siguientes ficheros:

► **Documentación:** fichero que contiene la memoria en formato PDF.

► **Código:** fichero que contiene el proyecto de NetBeans de la aplicación

► **Programa:** fichero que contiene lo necesario para instalar la aplicación siguiendo el manual de instalación.

Parte VI

Webgrafía

Webgrafía

A continuación aparece un listado de las fuentes utilizadas para el desarrollo de este proyecto:

-01 Sobre la estimación del consumo de un PC, visitado el 24/02/2019, disponible en "https://hardzone.es/2015/03/31/cuanto-cuesta-la-electricidad-que-consume-tu-pc/"

-02 Sobre el sueldo medio de un Analista, visitado el 24/02/2019, disponible en "https://www.indeed.es/salaries/Analista-programador/a-Salaries"

-03 Sobre el sueldo medio de un Programador Junior, visitado el 24/02/2019, disponible en " https://www.indeed.es/salaries/Programador/a-junior-Salaries"

-04 Sobre el sueldo medio de un Tester, visitado el 24/02/2019, disponible en "https://www.indeed.es/salaries/Tester/a-Salaries"

-05 Sobre la seguridad de AES, visitado el 26/02/2019 , disponible en "https://crypto.stackexchange.com/questions/50605/how-likely-is-a-aes-kdf-bypass"

-06 Sobre SHA-384 y ejemplos para verificar el correcto funcionamiento, visitado el 27/02/2019, disponible en "http://www.iwar.org.uk/comsec/resources/cipher/sha256-384-512.pdf"

-07 Sobre los tipos de Datos para almacenar fechas en java, visitado el 13/03/2019, disponible en "https://www.journaldev.com/2800/java-8-date-localdate-localdatetime-instant"

-08 Sobre la generación de números aleatorios seguros en java, visitado el 16/03/2019, disponible en "https://es.stackoverflow.com/questions/5390/como-generar-n%C3%BAmeros-aleatorios-dentro-de-un-rango-de-valores"

-09 Sobre la presencia de caracteres en strings en java, visitado el 16/03/2019, disponible en: "https://stackoverflow.com/questions/14278170/how-to-check-whether-a-string-contains-at-least-one-alphabet-in-java"

-10 Sobre la conversión de hexadecimal a ASCII en java, visitado el 12/03/2019, disponible en:"https://www.mkyong.com/java/how-to-convert-hex-to-ascii-in-java/"

-11 Sobre la conversión de hexadecimal a string binario en java, visitado el 12/03/2019, disponible en:
"https://stackoverflow.com/questions/8640803/convert-hex-string-to-binary-string"

-12 Sobre la obtención de las dimensiones de la pantalla del dispositivo, visitado el 19/05/2019, disponible en:
"https://alvinalexander.com/blog/post/jfc-swing/how-determine-get-screen-size-java-swing-app"

-13 Sobre la edición de la altura del título de las pestañas en un JTabbedPanel, visitado el 28/05/2019, disponible en:
"http://www.java2s.com/Tutorials/Java/Swing_How_to/JTabbedPane/Handle_the_height_of_the_tab_title_in_JTabbedPane.htm"

-14 Sobre la adaptación de un JSpinbox para la recogida de números, visitado el 28/05/2019, disponible en:
"http://www.java2s.com/Tutorials/Java/Swing_How_to/JSpinner/Create_JSpinner_for_number_with_number_editor.htm"

- 15 Sobre la edición de un JButton para solo mostrar el icono que contiene, visitado el 31/05/2019, disponible en:
"https://stackoverflow.com/questions/20566772/java-jbutton-only-image"

-16 Sobre redimensionar iconos dentro de componentes, visitado el 31/05/2019, disponible en:
"https://stackoverflow.com/questions/2856480/resizing-a-imageicon-in-a-jbutton"

-17 Sobre hacer transparente el fondo de un componente, visitado el 02/06/2019, disponible en:
"https://stackoverflow.com/questions/30435186/how-to-make-jtextarea-transparent-background"

-18 Sobre hacer visible el contenido de un JPasswordfield, visitado el 02/06/2019, disponible en:
"https://stackoverflow.com/questions/20812857/how-to-display-characters-in-jpasswordfield-rather-than-sign-in-java"

-19 Sobre el acceso al portapapeles desde java, visitado el 03/06/2019 disponible en:
"http://chuwiki.chuidiang.org/index.php?title=Uso_del_Clipboard_del_sistema"

-20 Sobre detectar la inactividad del usuario, visitado el 03/06/2019, disponible en :"http://www.java2s.com/Tutorials/Java/Swing_How_to/JFrame/Close_JFrame_after_user_inactivity.htm"

-21 Sobre la lectura de recursos en un archivo jar, visitado el 12/06/2019, disponible en "https://stackoverflow.com/questions/31127/java-swing-displaying-images-from-within-a-jar"