Àlex Soriano Faiges

Sistema de Autenticación Segura en Servicios Online mediante un móvil

TRABAJO DE FIN DE GRADO

Dirigido por el Dr. Jordi Castellà Roca y Cristòfol Daudén Esmel

Grado en Ingeniería Informática

UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2022

Agradecimientos

En este trabajo se han visto involucradas muchas personas que me han apoyado durante el proceso, por tanto, este apartado va para ellos.

En primer lugar, agradecer a mi familia y amigos Adrià, Bernat, Laura y Alex entre otros, por estar ahí mostrando su apoyo durante todo el desarrollo del proyecto, aportando sus opiniones e interesándose por mi trabajo.

Mencionar también a dos grandes grupos de amigos "MDT" y "Alsa" que han estado durante estos últimos años y en este último proyecto de mi grado, haciendo más ameno mi camino aportando su compañía y amistad.

Finalmente, quiero expresar mi más grande y sincero agradecimiento al Dr. Jordi Castellà Roca, principal guía en el desarrollo del proyecto y quien supo motivarme para cumplir los objetivos propuestos, y a Cristòfol Daudén Esmel que siempre se ha dispuesto a ofrecerme su ayuda cuando la he requerido.

Índice

Resumen

El número de servicios online que requieren de autenticación a los usuarios es muy grande, generalmente mediante usuario y contraseña, esto provoca que los usuarios terminen con un elevado número de usuarios y contraseñas que recordar, o utilizando la misma contraseña en muchos sitios. En el primer caso, hace que la gestión sea compleja, y en el segundo, que el compromiso de una contraseña afecte a la seguridad de muchos servicios.

En este trabajo se desarrolla un sistema modular que permite autenticarse contra un servicio online previamente registrado en el sistema de manera robusta empleando el móvil. Se ha efectuado un servicio online para probar la eficacia y eficiencia del sistema, que consta de un servidor para autenticar a los usuarios y un servidor web para realizar las peticiones de autenticación. En el proceso de registro del usuario en un servicio online se usa el DNI v3.0 y un teléfono móvil NFC. Esto permite establecer las claves de autenticación de forma segura y recuperarlas en caso de cualquier incidencia.

1. Introducción

En la actualidad disponemos de un gran nombre de servicios online para facilitar o complementar nuestra vida diaria. Unos ejemplos claros son las redes sociales, las cuentas bancarias, los servicios multimedia como Netflix y HBO, entre muchos otros.

Cuando usamos estos servicios, algunos necesitan del registro y la autenticación del usuario, la cual se basa en comprobar si coincide la información ofrecida por el usuario, con la guardada por el servicio en el registro; de modo que solo el usuario es capaz de administrar su propia información en sus servicios online, por ejemplo, cuando administramos nuestra cuenta bancaria o cuando recibimos recomendaciones específicas de nuestras plataformas multimedia basadas en nuestros gustos particulares.

El hecho de autenticarse provoca que podamos identificarnos, lo cual es muy útil debido a que los usuarios pueden efectuar acciones en el servicio, quedando estas registradas a su identidad. Diferenciar que la identificación es un dato público, mientras que l'autenticación utiliza información compartida entre el usuario y el servicio.

En el campo de la Autenticación tenemos muchos métodos para asegurarnos de que ningún otro sujeto podrá suplantar nuestra identidad, actuar en nuestro nombre o beneficiarse de nuestros privilegios. Uno de los métodos más comunes y conocidos por los usuarios es el uso de contraseñas. [9]

El **uso de contraseñas** constituye que el usuario introduce un nombre y la contraseña correspondiente al nombre dado. El grado de seguridad depende directamente de la complejidad de la contraseña, de forma que también es un gran punto débil. Los expertos recomiendan contraseñas de 12 caracteres, con una cardinalidad de 94, que representa que los caracteres están elegidos de un abanico de 94 posibilidades para cada carácter, y con un nivel de entropía alto. Se calcula la seguridad de la contraseña en bits enfrentando un ataque de fuerza bruta.

> A 12 characters password with 94 cardinality and 78.7 bits entropy will take 55 days to crack using super computers. And using PC it will take 3018 years to crack. [2]
>
> (A Review Of Authentication Methods, 2016: ISSN 2277-8616)

Lo ideal seria que cada usuario usara una contraseña distinta para cada servicio, pero debido a que cada usuario de media posee unas 25 cuentas que tendrían que tener su respectiva contraseña [3], y que cada contraseña para que sea segura tiene que tener una cierta complejidad, acaba provocando que la creación y memorización sea una tarea complicada.

Debido al gran número de contraseñas que tienen que manejar los propios usuarios acaban realizando malas prácticas, la media de contraseñas que consta que un usuario promedio tiene es de 7 contraseñas compartidas, es decir, se reúsan contraseñas entre los diferentes servicios online, provocando un gran riesgo en el caso de que un atacante logre averiguar una. [3]

El hecho de tener que recordar la contraseña también influye en que estas no sean lo suficientemente largas y variadas, resultando ser simples, y por tanto, más fáciles de averiguar con ataques de fuerza bruta, donde el atacante puede usar una herramienta para intentar cada combinación de letras y números, esperando en algún momento hallar la contraseña.

Se puede ver en el listado de contraseñas más utilizadas en el 2021, que todas las posiciones iniciales son también las más vulnerables [5], aumentando así la facilidad de averiguar contraseñas con ataques de diccionario, donde el atacante prueba una lista de posibles contraseñas de uso común para poder hallar la que usa el usuario.

Las contraseñas también son susceptibles a los ataques de Ingeniería social (phishing), que intentan estafar al usuario para obtener datos privados del mismo, especialmente para acceder a sus cuentas o datos bancarios.

Aparte de las responsabilidades del usuario al administrar sus contraseñas, también existe un compromiso por parte de los servidores a la hora de guardar las contraseñas, para que estas no estén de forma visible para un atacante.

1.1. Objetivos

Se quiere diseñar un sistema de autenticación robusto, modular, fácil de usar y seguro, que permita eliminar el uso de las contraseñas como método de autenticación en los servicios online.

En lo que a robusto se refiere, se quiere proporcionar al usuario la soberanía de su seguridad, es decir, otorgándole únicamente a este la posesión de sus datos de autenticación, sin que tenga que memorizar estos datos.

Una implementación modular permitiría añadir nuevos servicios online al sistema sin apenas modificar la estructura, tratando así a los servicios online como módulos fáciles de administrar. La implementación del sistema de autenticación está enfocada en un entorno móvil, y también se tendrá que diseñar un módulo de ejemplo para autenticarse en él, verificando su correcto funcionamiento.

Para una adaptación a la actualidad, el sistema ha de ser fácil de utilizar a la vez que de aprender, con una interfaz clara para el usuario e incluso alguna explicación breve de su uso.

Las funciones del sistema permitirán registrarse, autenticarse y recuperar las credenciales de autenticación en caso de perdida para cada servicio online registrado.

1.2. Motivación

Este sistema viene inspirado para aportar a todos los usuarios un punto de unión para gestionar todas las cuentas de sus diferentes servicios online de forma centralizada, en el dispositivo móvil que el usuario elija, y sin que este tenga que recordar las contraseñas para cada servicio, denegando el uso de contraseñas débiles o repetitivas. Además de llegar a superar la seguridad que ofrece el modelo de contraseñas actual.

Debido a que actualmente la mayoría de la población dispone de acceso a internet [4], y por tanto, cada vez es más necesaria la capacidad de autenticarse, este sistema es útil para todo tipo de usuario que quiera poder gestionar sus servicios online.

También provee de una mayor comodidad y rapidez a la hora de registrarse en un nuevo servicio online por parte del usuario, reduciendo el tiempo y el esfuerzo dedicado para obtener una cuenta.

En la actualidad están aumentando el número de ataques a las contraseñas convencionales, motivando a cambiar de sistema de autenticación.

Los ataques de fuerza bruta fueron el principal vector de intrusión a redes informáticas, con el 53 % de las detecciones en el segundo cuatrimestre de 2021, además el crecimiento de estos ataques en 2020 aumento un 768 %. [6]

Los ataques de phishing aumentaron un 29 % en 2021 en comparación con 2020. [8]

1.3. Organización de la Memoria

En la memoria se tratará de explicar el proceso de creación del software, para ello se ha dividido en seis grandes apartados:

- Sección 1: En esta sección se establece un contexto para el marco del trabajo fijando unos objetivos y presentando las motivaciones.
- Sección 2: Se encuentra una breve descripción de las tecnologías utilizadas para realización del trabajo.
- Sección 3: Contiene la explicación del diseño del sistema junto con los requisitos que este debe cumplir y se explican las funcionalidades que se ofrecen al usuario.
- Sección 4: En este apartado se explican brevemente las composiciones y las funcionalidades que ofrecen cada una de las estructuras definida en el diseño.
- Sección 5: En esta sección se evalúa el correcto funcionamiento del sistema en sus funcionalidades junto con imágenes de los resultados obtenidos.
- Sección 6: Finalmente, en la última sección se relatan las conclusiones obtenidas en la realización del trabajo junto a las sensaciones derivadas durante su realización.

2. Tecnologías Utilizadas

2.1. NFC

La tecnología NFC (Near Field Communication), permite realizar un intercambio de datos entre dos dispositivos de forma inalámbrica a corta distancia y a una alta velocidad (424 Kbits/s), lo que significa que está diseñado para enviar pequeñas cantidades de información a una alta frecuencia. [7]

La corta distancia que se debe emplear es una ventaja para evitar la lectura de la transmisión de datos por parte de un usuario externo, pero no se puede descartar la copia de los códigos de nuestro chip para un uso fraudulento.

Esta tecnología nos ofrece una gran variedad de utilidades como identificarnos, pagar con el teléfono móvil y sincronizar instantáneamente dispositivos, entre muchas más.

2.2. Dni 3.0

El Documento Nacional de Identidad (DNI), es básico en nuestra sociedad y permite identificar a los ciudadanos del estado español. En la actualidad, desde 2015 se está utilizando su versión 3.0, que consta de una compatibilidad con la tecnología NFC y aumenta su seguridad respecto la versión anterior del DNI.

El DNI incluye un chip donde se encuentran los datos personales, la fotografía del titular del documento, el patrón de la huella dactilar y los certificados:

- Certificado de Autenticación: Usado para demostrar que es el mismo usuario propietario del DNI el que está haciendo uso de él.
- Certificado de Firma: Usado para firmar trámites o documentos a través de Internet.

2.2.1. DNIeDroid v2.3

Es un kit de desarrollo del cuerpo nacional de policía, creado para facilitar el uso de la tecnología del DNIe 3.0 y versiones posteriores, a los usuarios que quieran diseñar aplicaciones móviles en Android utilizando la tecnología NFC.

Se trata de un middleware encargado de gestionar la conexión entre el móvil y el DNIe, ofreciendo una API básica y transparente al usuario de las operaciones que se pueden efectuar,

sin que el programador tenga que conocer al detalle el funcionamiento de la tarjeta, y pueda aplicar dichas funcionalidades a sus propios proyectos. [1]

2.3. PKI

Una Infraestructura de clave pública (PKI) es una combinación de hardware y software aplicándose a la seguridad digital. Se basa en integrar los certificados digitales, con las diferentes entidades de certificación, junto con la criptografía de clave pública.

Su finalidad es aportar seguridad y garantías a operaciones de identificación, cifrado, autenticación y firma.

Se utilizan algoritmos de cifrado accesibles por todos, por tanto, la seguridad depende vitalmente de la privacidad de la llamada clave privada.

2.4. Spring Boot

Es un módulo que fue creado para simplificar el desarrollo de aplicaciones con Spring Framework. Nos brinda una infraestructura para el desarrollo de aplicaciones en una plataforma Java de código abierto, permitiendo olvidarnos de la arquitectura y enfocarnos en el desarrollo.

Utiliza internamente un servidor de aplicaciones, por defecto Tomcat.

2.5. Android Studio

Es el IDE (Entorno de desarrollo integrado) oficial para desarrollar apps en Android, basado en IntelliJ IDEA.

3. Arquitectura y Diseño

La arquitectura de este sistema de autenticación se puede dividir en 4 unidades que se comunican entre ellas (representadas en la figura 1): i) la aplicación móvil en el dispositivo del usuario, ii) los servidores de autenticación pertenecientes a los diferentes servicios online, iii) los servidores web de estos servicios y, iv) los dispositivos que se conectan a estos servidores web. Destacar que el sistema está pensado para ser modular, disponiendo de más de un servicio online en la aplicación con su respectivo servidor y servidor web, aunque para explicar el funcionamiento del sistema nos centraremos en la utilización de un solo servicio que tendrá su servidor de autenticación y su servidor web.

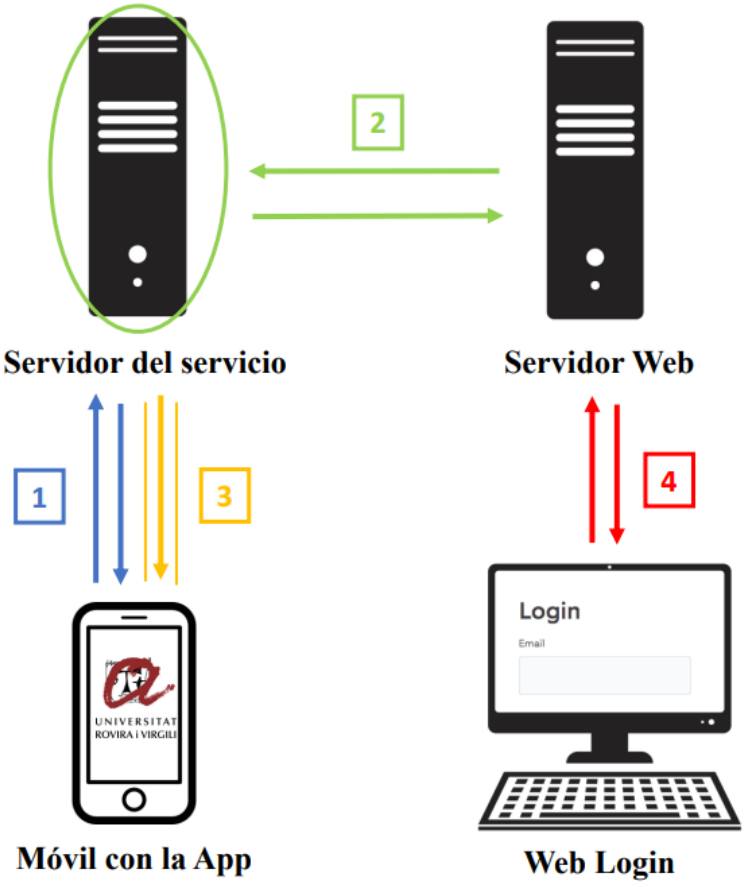


Figura 1: Arquitectura básica del funcionamiento del Sistema de registro y autenticación.

3.1. Estructuras de la arquitectura

En este apartado se mencionan las diferentes tareas y funcionalidades que ofrecerán las distintas estructuras del sistema.

Las funcionalidades de la aplicación móvil se basan en poder contactar con los diferentes servidores de autenticación de los servicios online que están registrados en la aplicación, con el uso de una interfaz intuitiva y precisa. Pudiendo así estos usuarios registrarse en cualquier servidor, recibir solicitudes de autenticación de los servicios donde el usuario esté registrado y ofrecer la posibilidad de rehacer las credenciales de autenticación para contactar con dicho servidor en caso de perdida de estas.

Sobre el servidor recae la obligación de administrar todos los usuarios de su servicio, y por tanto dispone de una base de datos relacional SQL, donde se almacenaran todos los datos de identidad de los usuarios como los datos para su correcta autenticación. El servidor se encarga también de ser el intermediario entre el dispositivo móvil y el servidor web del servicio, realizando la validación de las credenciales para el correcto funcionamiento del sistema. Se ve involucrado en las operaciones de registro y recuperación que efectúa el usuario desde su dispositivo móvil, también en la funcionalidad de login que inicia el usuario al contactar con el servidor web del servicio, donde el servidor comunica al dispositivo móvil correspondiente la petición, este acepta o deniega, y el servidor de autenticación redirige la respuesta al servidor web y este al dispositivo del usuario.

Por la parte del servidor web, servidor donde se conecta el usuario para autenticarse en un servicio. Se le brinda una interfaz al usuario donde debe insertar su dirección de correo electrónico correspondiente a una cuenta ya registrada, y posteriormente el servidor web contacta con el servidor de autenticación del servicio para validar o no la identidad, según el resultado permitirá o denegara al usuario el acceso al servicio con la identificación demandada.

3.1.1. Comunicaciones en la arquitectura

Existen diferentes vías de comunicación entre estas estructuras para poder intercambiarse información; estas comunicaciones emplean HTTP con SSL dentro del sistema para reforzar la seguridad, por lo cual se ha creado una entidad de certificación propia, que será la que respaldara los certificados creados para el servidor de autenticación y el servidor web en las comunicaciones.

Como se puede observar en la figura 1, la comunicación representada con las flechas azules(1), se efectúa sobre HTTPS y es iniciada por el dispositivo móvil hacia el servidor de autenticación obteniendo una respuesta de este.

Las flechas verdes(2) representan que el servidor web ha recibido una petición por parte de un dispositivo para autenticar a un usuario, en consecuencia inicia la comunicación con el servidor de autenticación del mismo servicio y recibe una respuesta, esta comunicación también se establece utilizando HTTPS.

Siendo la primera comunicación que se establece, tenemos la comunicación representada en amarillo(3). Cuando el dispositivo móvil arranca la aplicación, esta crea una conexión abierta o canal como cliente con el servidor de autenticación del servicio, para permitir que el servidor pueda comunicarse con él, sin que previamente el cliente le realice una petición. Como podemos observar en la figura 1, el envío de una notificación push (del servidor al cliente), no exige una contestación por parte del cliente y en este caso se establece únicamente sobre HTTP.

Por último, la común conexión(4) que se efectúa entre un dispositivo y un servidor web con HTTPS.

3.2. Requisitos de funcionamiento

El sistema debe cumplir con estos requisitos para realizar correctamente su funcionalidad:

- La aplicación tiene que mostrar una interfaz clara, fácil de entender, sencilla de usar y informar al usuario de todos los resultados obtenidos en sus funcionalidades.
- La aplicación para registrar a un usuario en un servicio online hace uso del certificado de firma del DNI 3.0 del usuario.
- La aplicación guarda los datos sensibles para la autenticación del usuario en su dispositivo móvil y, para usar-los se requiere superar una autenticación biométrica.
- Solo el usuario propietario del dispositivo es capaz de realizar las funciones de la aplicación.
- Solo se permite el acceso a un servicio online, a un usuario registrado y que haya aceptado la petición de login en su dispositivo.
- La notificación recibida al dispositivo del usuario cuando se intente autenticar en un servicio online ha de ser rápida.
- El servidor del servicio podría ser escalable, pudiendo tener varios de ellos repetidos para no crear cuellos de botella.
- La administración de los servicios online por parte de la aplicación será modular, pudiendo administrarlos fácilmente.
- Las comunicaciones en el sistema tienen que ser seguras, verificando el emisor y que el contenido no ha sido modificado.
- El sistema debe funcionar para varios usuarios simultáneamente.

3.3. Funcionalidades del Usuario

Tenemos diferentes acciones que puede realizar el usuario dentro del sistema, como se puede observar en la Figura 2 el usuario solo interacciona con la aplicación móvil o con la web que le provee el servidor web del servicio.

Las Acciones de la aplicación móvil son:

- ***Añadir Cuenta:*** Sirve para obtener los datos del usuario que se usaran posteriormente como cuenta. Muestra una ventana con una serie de campos por introducir para que el usuario los complete, y al finalizar, si todos los datos están bien introducidos, se guardaran como sesión del usuario, para posteriormente utilizarlos en las demás funcionalidades. Esta funcionalidad se muestra automáticamente al iniciar la aplicación en caso de que esta no disponga de datos de sesión previamente guardados y puede ser usada para cambiar de usuario.

- ***Registrarse:*** Se utiliza para registrar al usuario en el servidor del servicio online, generando y enviando un mensaje para dicho servidor con los datos para crear una cuenta y para posteriormente establecer una autenticación y comunicación seguras. Esta funcionalidad tiene como prerrequisito añadir antes una cuenta, para saber que datos tiene que usar para registrar al usuario.

- ***Recuperar Llaves:*** Sirve para regenerar los datos de autenticación del usuario en un servicio específico tanto en el lado del servidor como en la aplicación móvil del usuario. Para ello se pide al usuario el uso de sus datos de autenticación utilizados en el registro, para poder verificar su identidad en el lado del servidor de autenticación, y confirmar que es el mismo usuario que realizo el registro, a continuación se le muestra la lista de servidores a los cuales está registrado, y al seleccionar uno tendrá que emplear su huella dactilar para autorizar el registro, confirmando que es propietario del dispositivo. Esta funcionalidad tiene como prerrequisito añadir antes una cuenta, para poder comunicar al servidor de que usuario se quiere recuperar las llaves.

- ***Aceptar o denegar la autenticación:*** Utilidad que únicamente se muestra si anteriormente el usuario ha efectuado la acción de introducir correo y nombre en la web del servicio online, con los datos que relaciona de su cuenta. Sirve para aceptar o denegar el inicio de sesión a ese servicio. Se muestra en el dispositivo móvil del usuario una ventana que muestra un botón para cancelar, para denegar así el acceso, e indica que para aceptar se tendrá que emplear la huella dactilar. Como requisito para que esta acción se complete correctamente, en el dispositivo móvil se tiene que hallar las credenciales de autenticación en ese servidor generadas anteriormente con un registro o una recuperación de llaves.

La Acción en la Web del servicio online es:

- ***Login*** El usuario introducirá su correo, que servirá como identificador de la cuenta a la que se quiere acceder en ese servicio online, y esperara a una respuesta del servidor que solicitara el acceso al teléfono móvil correspondiente a ese identificador.

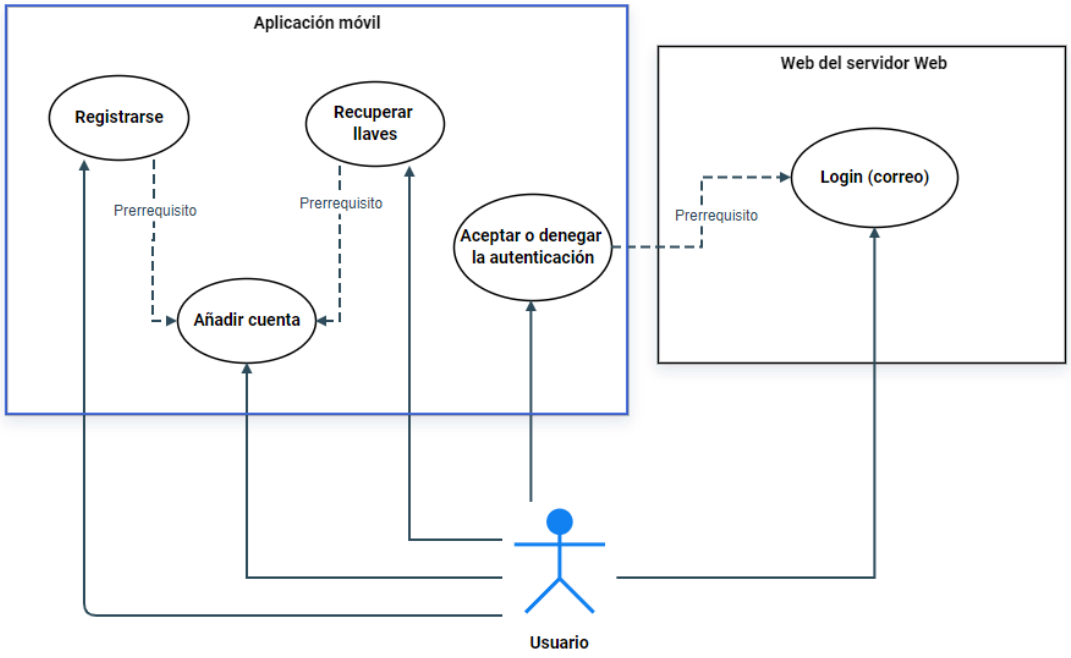


Figura 2: Diagrama de casos de uso del Sistema

3.3.1. Añadir cuenta

Es la acción más sencilla que ofrece la aplicación y esta descrito su funcionamiento en la figura 3. Su funcionamiento como hemos mencionado anteriormente es mostrar al usuario una ventana con campos para introducir, el primero se trata de una dirección de correo electrónico que se utilizará como identificador de esa cuenta en los servicios online, en segundo lugar tenemos un Alias que usaran los servicios, y por último tenemos el número CAN que será necesario para poder realizar la lectura del DNI vía NFC. Una vez todos los datos están introducidos, la aplicación verificara que sean correctos y notificara si la operación se ha efectuado correctamente o no al usuario.

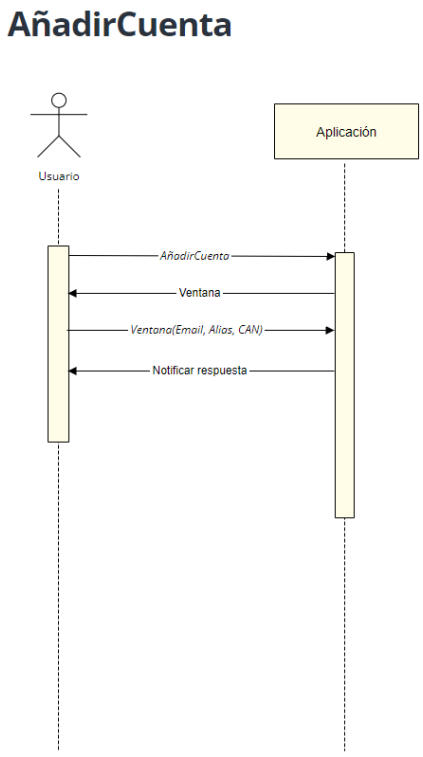


Figura 3: Diagrama de secuencia de la acción de Añadir Cuenta en el Sistema

3.3.2. Registro

Se detalla la secuencia de sucesos que se pueden observar en la figura 4 durante la acción de registro de un usuario en un servicio online. El usuario selecciona la opción de registro en la aplicación, habiendo antes añadido una cuenta, la aplicación le pide que introduzca su DNI utilizando NFC, para firmar con el certificado de firma, demostrando así su identidad y siendo utilizada después para otras funcionalidades, seguidamente muestra al usuario la lista de servidores a los cuales se puede registrar para obtener una cuenta, el usuario al seleccionar uno, tendrá que emplear su huella dactilar para autorizar el registro, confirmando que es propietario del dispositivo. Se generan entonces un conjunto de llaves, donde la llave privada se quedara guardada en el secure element del teléfono móvil del usuario específicamente para ese usuario en ese servicio online, y la llave publica se le otorgara al servidor.

Seguidamente se envían al servidor los datos del usuario junto con datos de control, el mismo conjunto firmado con el DNI y los datos del servidor. Los datos de control se basan en el certificado obtenido del DNI, la fecha, la llave publica generada, una palabra aleatoria firmada con la llave privada del usuario, y la misma palabra sin firmar, para que después el servidor pueda comprobar la validez de la llave publica que se le entrega.

El Servidor al recibir el mensaje comprueba la validez de la llave publica que se le otorga, verifica los datos firmados usando el certificado del DNI y también comprueba la fecha. Si todas las comprobaciones son correctas y el usuario no existe en la base de datos, lo crea, devolviendo una respuesta a la aplicación con el resultado de la operación, la fecha, un NONCE y un HASH del mensaje recibido para informar que mensaje ha recibido y por tanto a cual responde, todos estos datos son firmados con el certificado del servidor. En caso de algún error, el servidor también responde a la aplicación informando del error sucedido. Cuando la aplicación recibe la respuesta del servidor verifica su origen, verificando la firma del servidor y si el HASH que ha recibido coincide con el HASH que este ha enviado, y traslada el resultado de la operación al usuario.

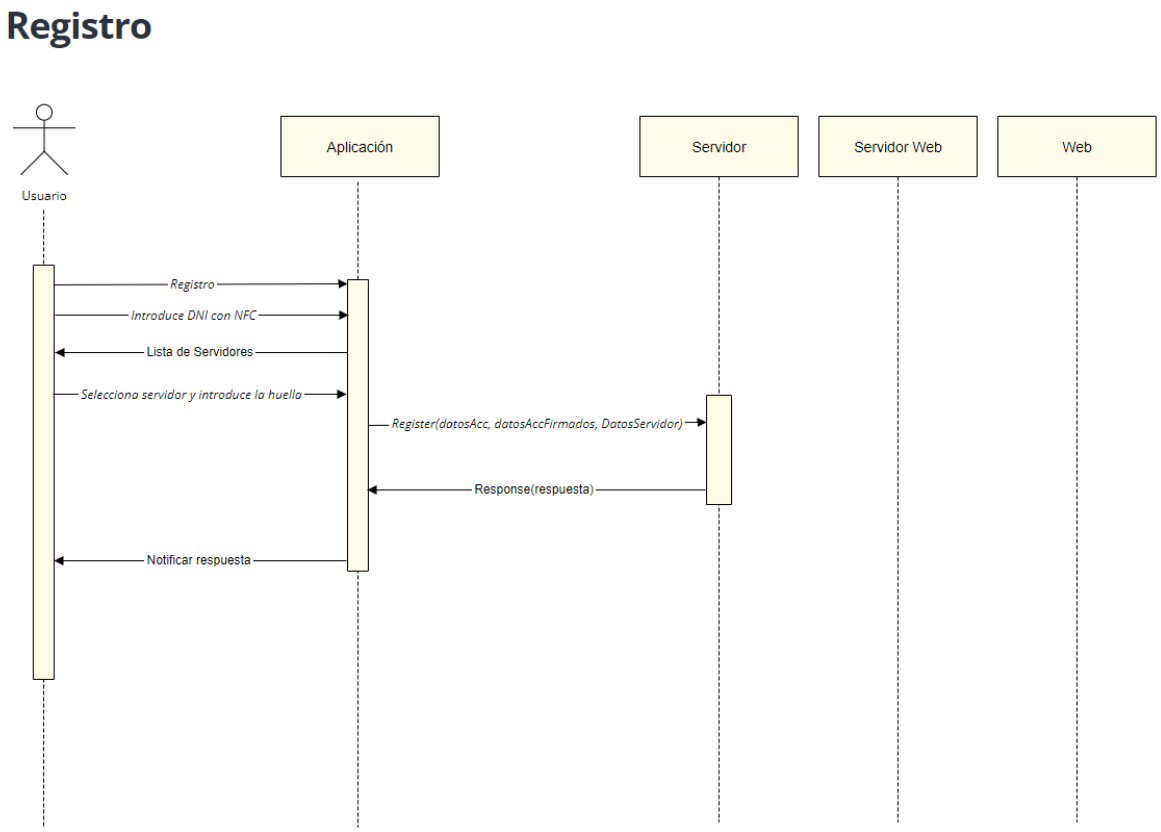


Figura 4: Diagrama de secuencia de la acción de Registro en el Sistema

3.3.3. Login

Se detalla la secuencia de sucesos que se pueden observar en la figura 5 durante la acción de autenticación de un usuario en un servicio web. El usuario una vez ha accedido a la página web del servicio online procede a introducir su correo electrónico que será utilizado como identificador en el servicio online.

El servidor web al recibir el correo lo envía a su servidor de autenticación pertinente, donde posteriormente se comprobara que exista dicho usuario en el servicio, que no exista una petición de acceso de ese usuario en proceso, y por último si se dispone de una conexión con el dispositivo móvil pertinente a ese identificador. Una vez efectuadas todas las comprobaciones, se creará un mensaje para el dispositivo móvil firmado con su clave pública guardada en el servidor, que contendrá la fecha, el nombre del servicio, el correo y un número NONCE, dicho mensaje se enviara como push usando el canal abierto que se ha mencionado anteriormente en el apartado 3.1.1, entre el servidor y el dispositivo móvil, además el servidor creara un hilo para esa petición que esperara 30 segundos y si en ese tiempo no se ha obtenido una respuesta del móvil del cliente, se responderá al cliente web que se ha excedido el tiempo de la petición.

La aplicación al recibir el mensaje verificara con su llave privada para ese servicio la autoría del servidor y revisara la fecha en la que se envió el mensaje, posteriormente pedirá una autorización biométrica para aceptar la petición a la que el usuario deberá responder, a continuación la aplicación generara un mensaje con la fecha, la id del usuario, el resultado de la petición, otro NONCE y un HASH del anterior mensaje recibido para señalar exactamente al servidor el mensaje al cual está contestando, y se firmara con la clave privada de ese usuario en ese servicio antes de ser enviado de vuelta al servidor de autenticación. Al recibirlo, el servidor comprueba la identidad del usuario,la firma efectuada, si hay alguna petición de login en proceso, la fecha, y por último el HASH del mensaje para confirmar que coincide con el que el envío como petición. Una vez hechas las comprobaciones finaliza comunicando a la aplicación si la comunicación se ha efectuado correctamente o no, y contestando al servidor web con el resultado obtenido del dispositivo móvil del usuario.

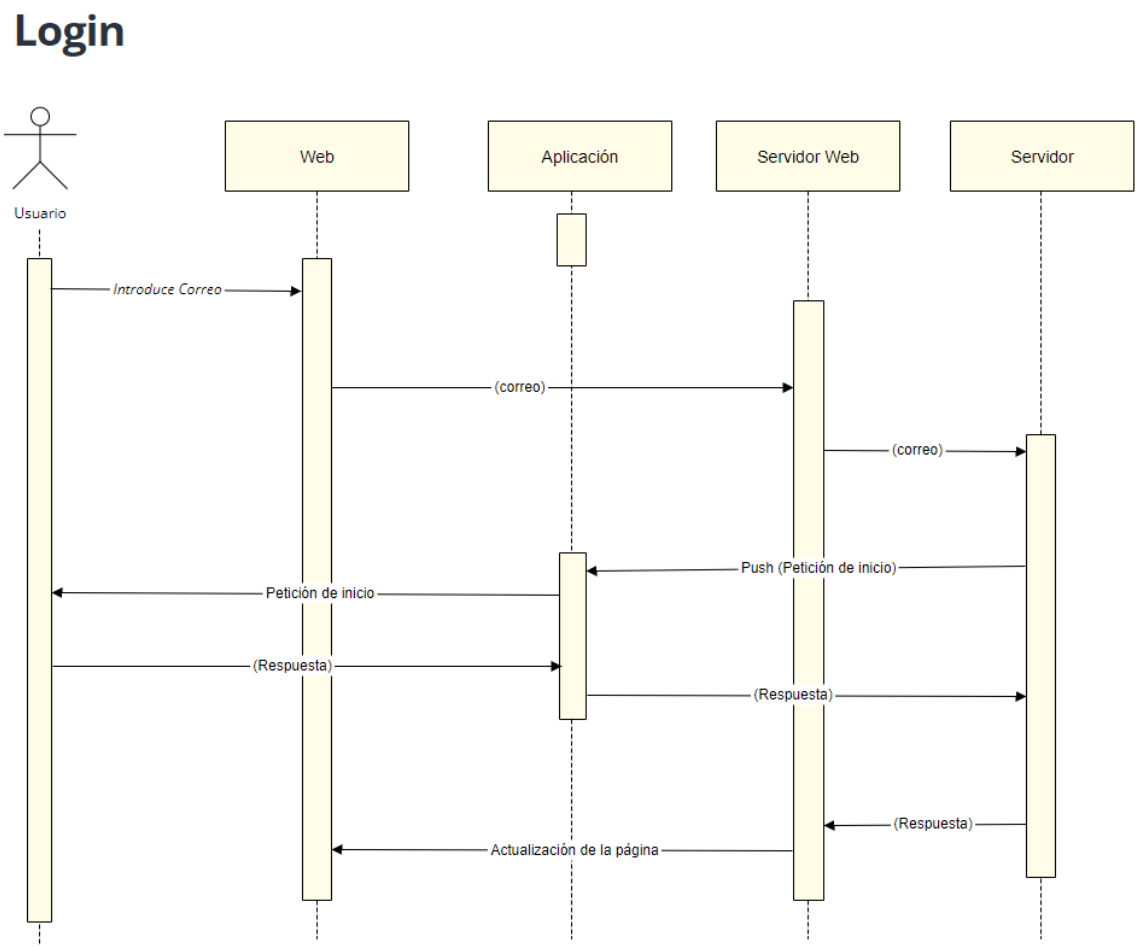


Figura 5: Diagrama de secuencia de la acción de Login en el Sistema

3.3.4. Recuperación de llaves

Se detalla la secuencia de sucesos que se pueden observar en la figura 6 durante la acción de recuperación de llaves de un usuario en un servicio online. Funcionalidad muy parecida a la acción de registro. Cuando el usuario accede a la acción de recuperación, la aplicación le muestra una lista de servidores, cuando el usuario selecciona un servidor e introduce la huella dactilar, se generan otro par de llaves y se le pide la utilización del DNI con NFC para poder firmar el futuro mensaje que se enviara al servidor, compuesto del correo (obtenido de la sesión), la nueva llave publica, la fecha, una palabra firmada con la llave pública y la misma palabra sin firmar.

El servidor, al recibir el mensaje, verifica la llave pública, el mensaje firmado con la llave privada del DNI,la fecha y la existencia del usuario. Si todas las comprobaciones son correctas, actualiza la llave pública de ese usuario, si no devuelve error. El mensaje de respuesta del servidor contiene un HASH del mensaje recibido, el resultado de la operación y la fecha, todo firmado con el certificado del servidor.

La aplicación, al recibir el mensaje, verifica la firma, el HASH recibido, la fecha y procede a mostrar el resultado al usuario.

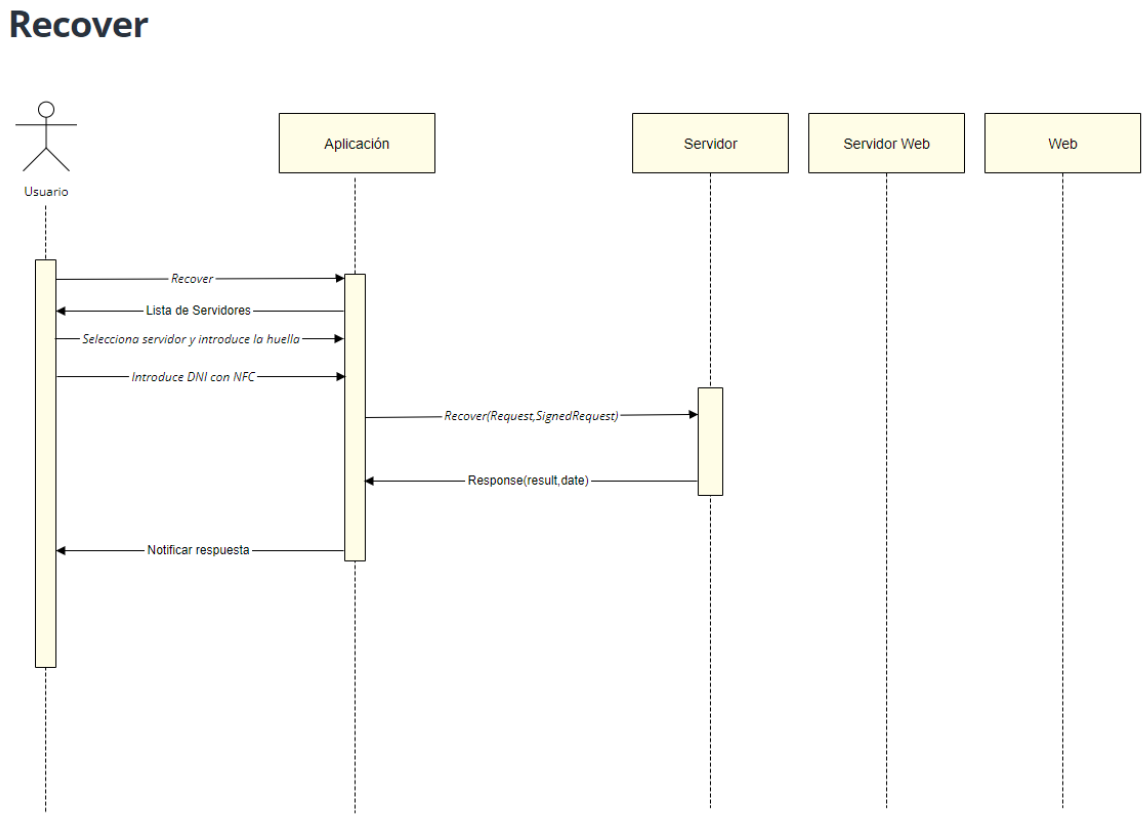


Figura 6: Diagrama de secuencia de la acción de Recuperación de llaves en el Sistema

4. Sección de implementación

En este apartado se estructuran las diferentes clases, métodos y datos que utiliza cada estructura del sistema y como estas se comunican.

4.1. Implementación de la aplicación

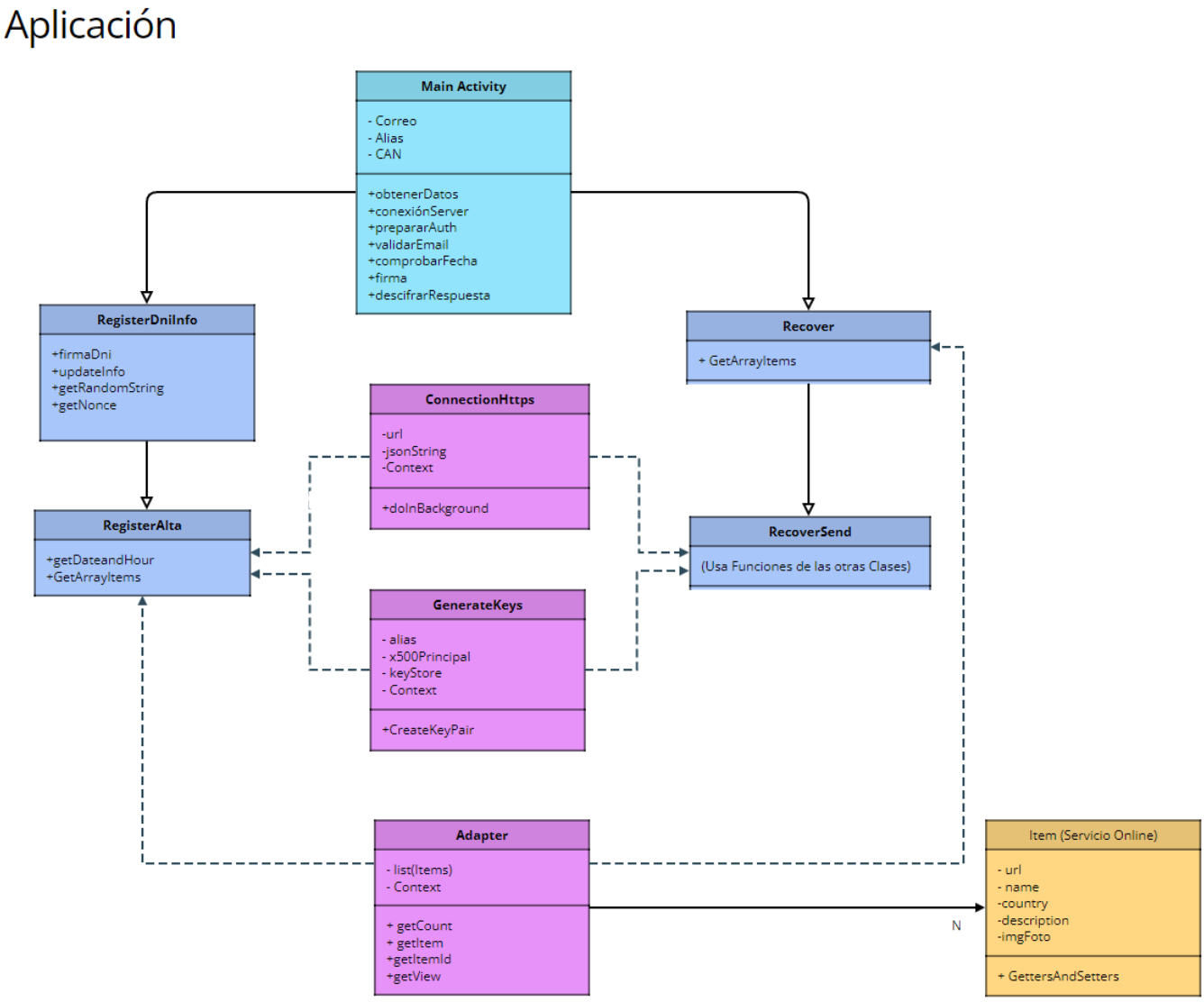


Figura 7: Diagrama de clases de la aplicación

El desarrollo se ha llevado a cabo utilizando Android Studio, y se explicarán sus componentes basándonos en la representación de la figura 7, todas las clases en color azul son actividades de la aplicación que son los estados que va representando la aplicación según las acciones del usuario, las de color púrpura se definen a clases de utilidad que son utilizadas por las actividades que marcan, y la clase amarilla (**Item**) es la estructura de los datos que guarda la aplicación sobre cada Servicio online.

La **clase adapter** es la que gestiona la lista de los diferentes Ítems, ofreciendo funciones para administrarla, y es la clase encargada de adaptar la lista para crear una representación visual para el usuario cuando esta sea necesaria.

Empezando por la clase **Main Activity** la cual tiene los objetivos de:

- Mostrar al usuario las diferentes funcionalidades que posee, registro, recuperación de llaves y añadir una cuenta.
- Crear los canales para recibir las notificaciones push de los servidores con el método conexionServer.
- Obtener una sesión antes de permitir el acceso a las otras funcionalidades.

La funcionalidad de añadir una cuenta, usa el método obtenerDatos explicada en la sección 3.3.1, y es indispensable para realizar cualquier otra de las funcionalidades, si no se halla una sesión se le denegara el acceso y se le indicara al usuario que debe añadir una cuenta.

El método conexionServer necesita de un correo, que servirá de identificador, de un nombre, que será el nombre del servidor, y de una URL. Se establecerá el canal abierto entre ese dispositivo y el servidor al cual pertenezca ese URL, identificando la conexión por parte del servidor empleando el correo, además de estar establecido el comportamiento de la aplicación al recibir una notificación push definido en la sección 3.3.3.

La clase de utilidad **ConnectionHttps** crea un nuevo hilo para no entorpecer la ejecución de la aplicación y realiza una comunicación HTTPS con el método POST a la URL asignada, enviando la información en forma de String dentro del objeto jsonString, y devuelve el resultado recibido de la comunicación.

La clase de utilidad **GenerateKeys** utilizando RSA genera un par de claves, que guarda en la keyStore que se le pasara por parámetro, con el alias que se le pasa por parámetro, y devuelve true si se han generado correctamente, en caso contrario devuelve false.

La actividad **RegisterDniInfo**, tiene como finalidad obtener la información necesaria del DNI utilizando NFC, notificando al usuario los pasos a seguir usando diferentes vistas. Una vez obtiene los datos del DNI, los combina junto a los de sesión y añade más datos detallados en la sección 3.3.2, para crear un mensaje, que será firmado con el certificado de firma del DNI, esta actividad finalizara enviando el mensaje original y el firmado a la actividad RegisterAlta.

La actividad **RegisterAlta**, muestra al usuario la lista de Servicios online utilizando la clase Adapter, a los que se puede registrar, cuando se selecciona un ítem la actividad pide una autenticación biométrica, el resultado se le notifica al usuario por pantalla. Al superarse la autenticación se comprueba que el usuario no esté registrado ya con en el servidor revisando la keyStore y se generan las nuevas llaves usando la clase GenerateKeys, se procede a crear un mensaje con los datos recibidos de la actividad RegisterDniInfo junto con los datos específicos de ese servidor, y se envía el mensaje resultante empleando la clase ConnectionHttps al servidor especificado, recibiendo una respuesta validándola y notificando al usuario el resultado por pantalla.

La actividad **Recover** muestra al usuario la lista de Servicios online utilizando la clase Adapter, en los que puede recuperar sus llaves, cuando se selecciona un ítem la actividad pide una autenticación biométrica, el resultado se le notifica al usuario por pantalla. Al superarse la autenticación se envía la información del ítem seleccionado a la actividad RecoverSend.

La actividad **RecoverSend** es la encargada de recopilar los datos del usuario con los datos de sesión junto con los datos de su DNI obtenidos vía NFC, se generan también un nuevo par de llaves, y se crea un mensaje con los datos obtenidos y la llave pública generada. Se envía un mensaje al servidor especificado utilizando la clase ConnectionHttps que contiene el mensaje en claro y el mensaje firmado con el DNI del usuario, seguidamente la actividad validara la respuesta y notificara al usuario el resultado por pantalla.

4.2. Implementación del Servidor

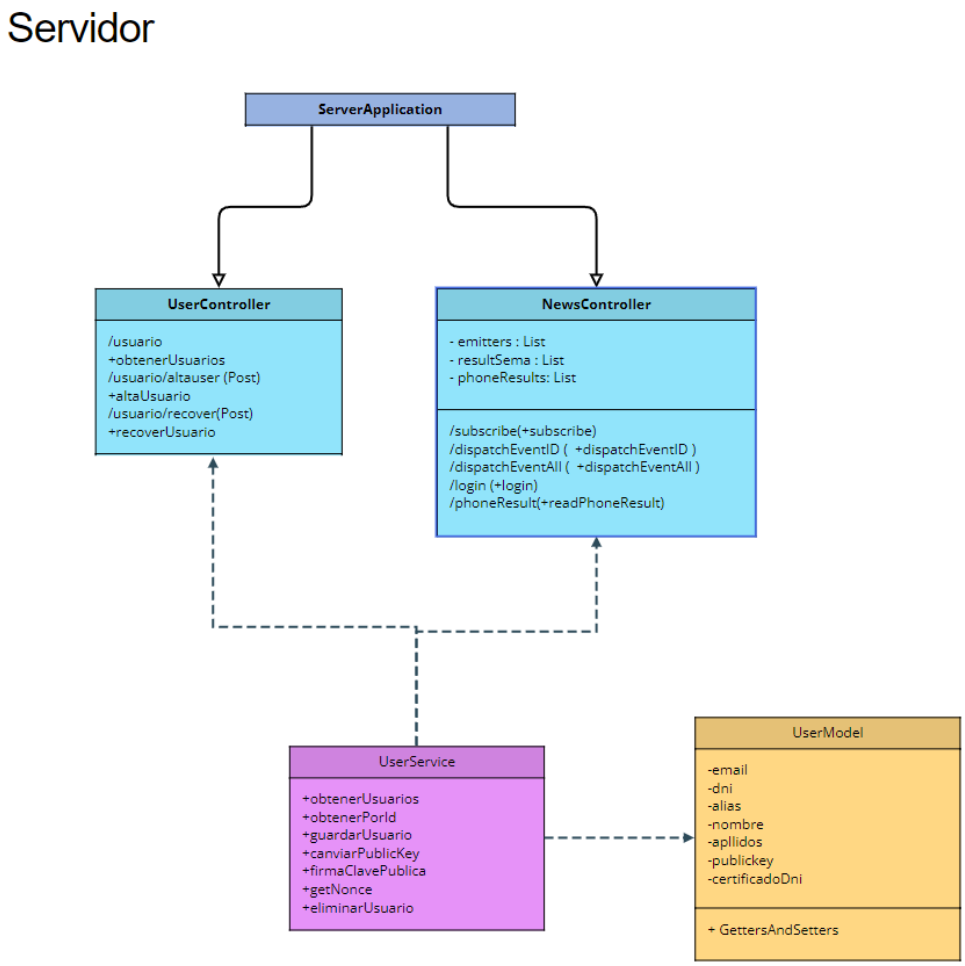


Figura 8: Diagrama de clases del Servidor

El servidor está creado usado la tecnología Spring Boot, y dispone de una base de datos que utilizara en sus operaciones, esta se crea automáticamente si no existe cuando se inicia el servidor, creando una tabla con los atributos definidos en la clase **UserModel** utilizando el correo como llave principal de cada registro.

La clase **UserService** contiene los métodos que luego se emplearan en las clases de control (color azul en la figura 8), en su mayoría se usa para administrar eficazmente la base de datos.

En la clase **ServerApplication**, se encuentra la configuración de puertos que empleara el servidor para recibir comunicaciones, siendo el 8080 para HTTP y el 8443 para HTTPS.

La clase **UserController** como dice su nombre, es un controlador que tiene definidos unos subdominios del servidor y las funciones que se tienen que ejecutar cuando estos reciban una petición. Cuando un mensaje se recibe en el subdominio /usuario/altauser, la función altaUser creada para registrar un usuario en el servicio, lo procesa esperando recibir un objeto JsonString para convertirlo en un objeto Json, y procesarlo tal y como hemos definido en la sección 3.3.2. Comparte el mismo funcionamiento con el subdominio /usuario/recover empleado en la acción de recuperación de llaves descrita en la sección 3.3.4. Cabe mencionar que estos dos dominios solo están disponibles usando el método de petición HTTP POST, usando el método GET solo funciona el subdominio /usuarios el cual muestra por pantalla la lista de usuarios registrados en el servidor utilizando la función obtenerUsuarios.

La clase **NewsController** es un controlador donde existen diferentes subdominios enfocados en el proceso de autenticación:

- El subdominio /subscribe origina un canal abierto con el dispositivo que le realiza una petición utilizando la id del usuario para identificarla, se guarda en la lista emitters.
- El subdominio /login recibe la petición de autenticación por parte del servidor web, crea un mensaje que se enviara por el canal abierto en forma de push al dispositivo correspondiente, crea un hilo para esa petición en específico para controlar el tiempo máximo de la petición y se bloqueará esperando la respuesta del dispositivo móvil del usuario.
- El subdominio /phoneResult recibe la respuesta del dispositivo a la petición de login, verificando los datos, desbloqueando la petición de login correspondiente a ese usuario y respondiendo al dispositivo móvil que se ha recibido el mensaje correctamente por parte del servidor.

4.3. Implementación del Servidor Web

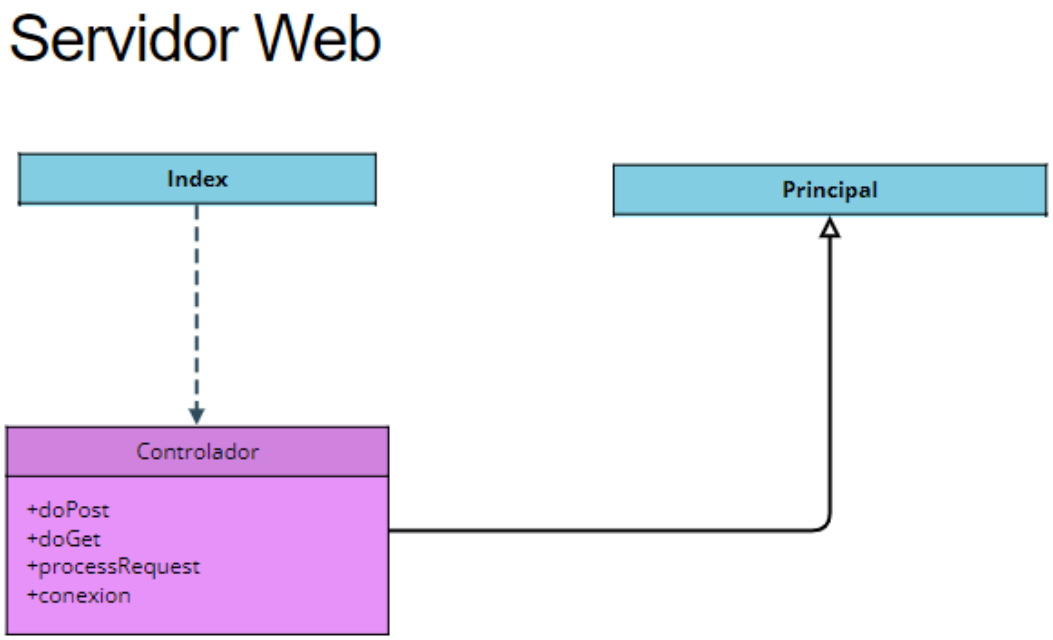


Figura 9: Diagrama de clases del Servidor Web

El servidor web, representado en la figura 9, consta de dos ficheros HTML, junto a un fichero JAVA llamado controlador.

El fichero **index** contiene el código HTML de la página de login del servidor web que se le mostrara al usuario cuando quiera autenticarse en ese servicio online, cuando presione el botón de ingresar se enviaran los datos introducidos en el login al **controlador** el cual realizara una petición al servidor de autenticación al subdominio login y esperara la respuesta, si se verifica correctamente la identidad del usuario se le mostrará al usuario el contenido del fichero **principal** que simularía la entrada al servicio web por parte del usuario, por el contrario, mostraría el mensaje de error recibido o si se le ha denegado acceso.

5. Juego de pruebas

En la figura 10 se puede observar que la aplicación que recibe el nombre provisional de ConnectApp está instalada y lista para su uso.

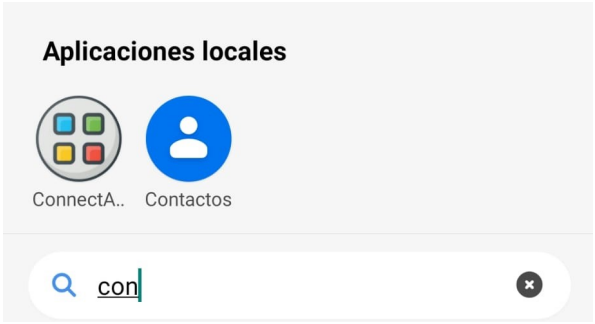
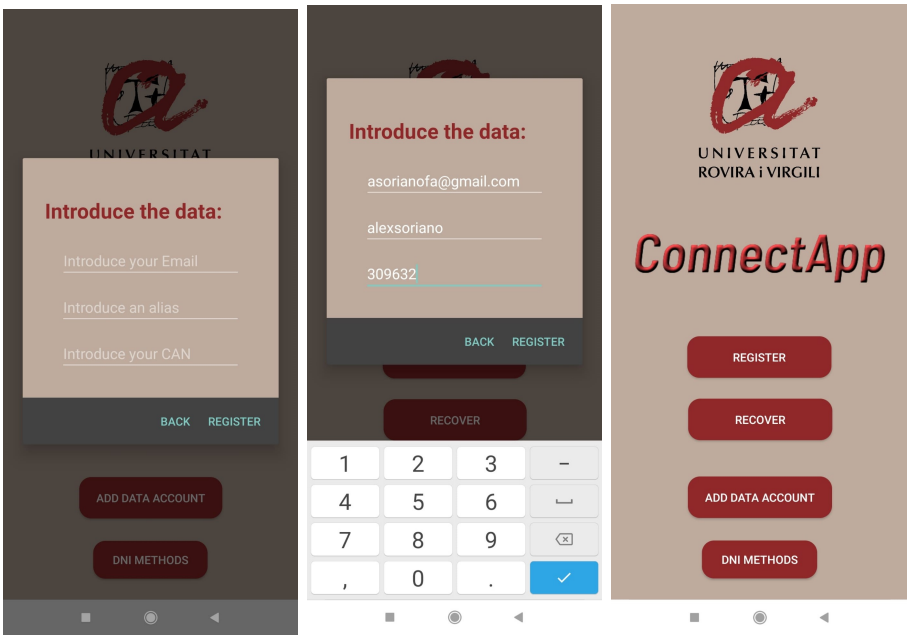


Figura 10: Icono de la aplicación en un dispositivo Android

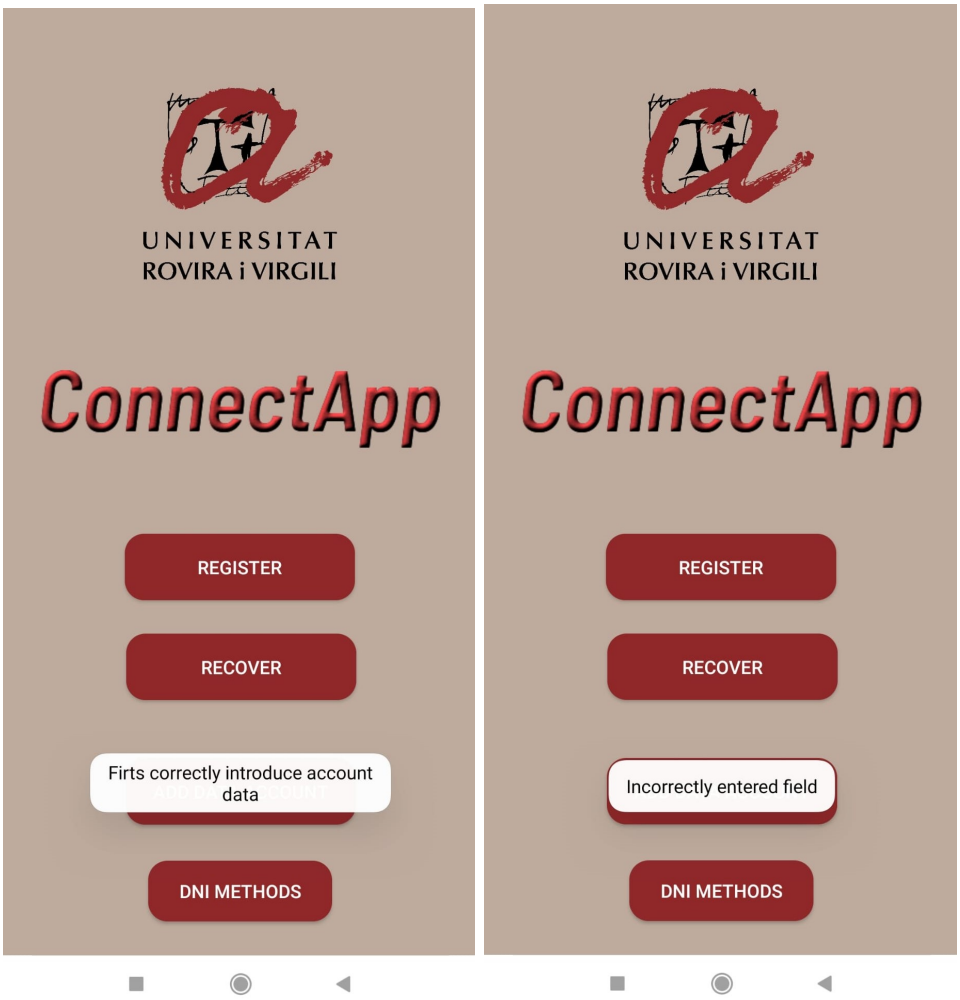
Al iniciar la aplicación por primera vez nos pedirá que ingresemos los datos de sesión (figura 11(a)), rellenaremos los campos y se nos mostrara la pantalla principal de la aplicación (figura 11(c)).



((a)) ((b)) ((c))

Figura 11: Proceso de Añadir Cuenta

En caso de no haber introducido los datos de sesión no se nos permitirá el acceso a otras funcionalidades como se puede observar en la figura 12(a), y si los datos de sesión presentan un formato incorrecto o están vacíos también se notificará (figura 12(b)).



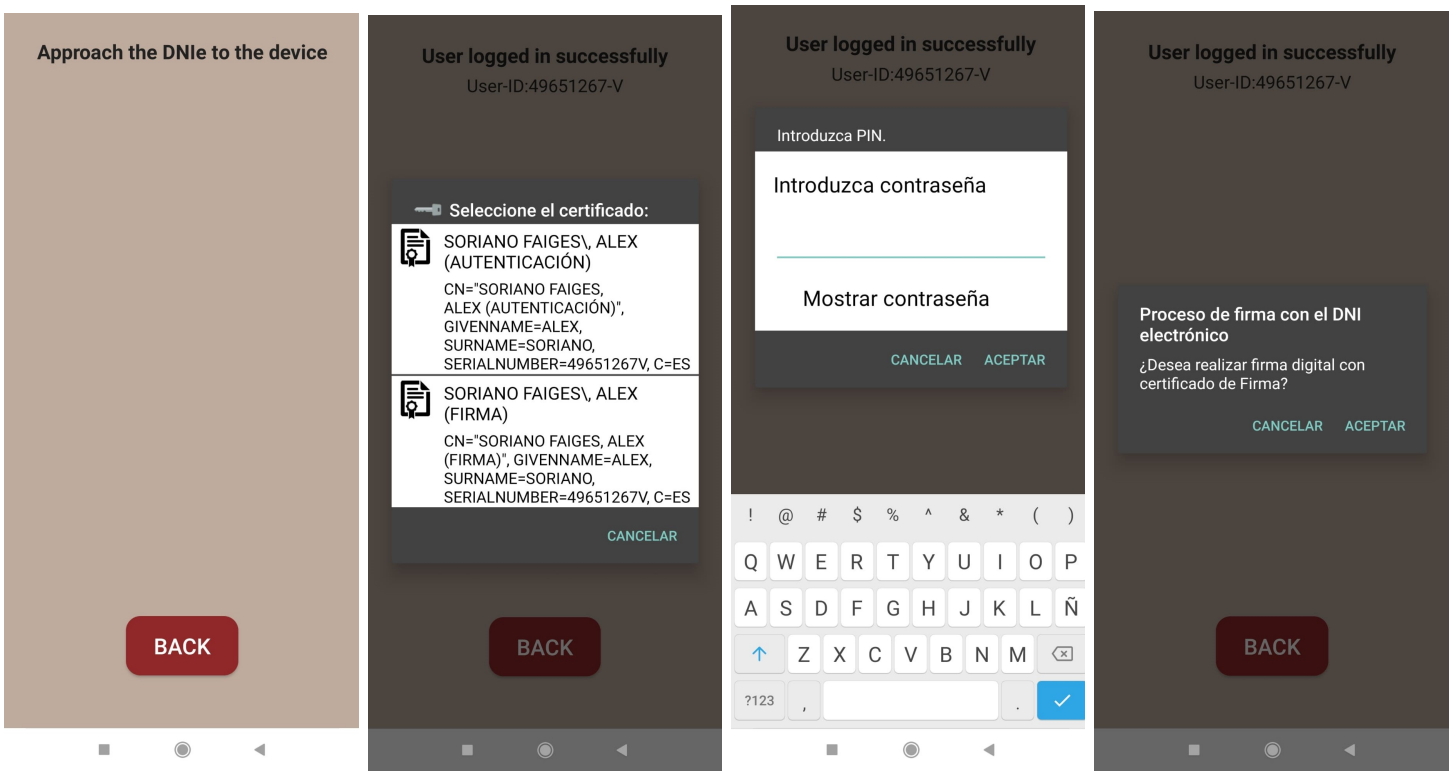
((a)) ((b))

Figura 12: Notificaciones de errores a la hora de añadir una cuenta

5.1. Prueba de Registro en un servicio online

Una vez añadida la sesión pasamos a probar el registro en un servicio online, lo primero que nos encontramos es una pantalla que nos indica que acerquemos el DNI al dispositivo para poder realizar la lectura con NFC (figura 13(a)), cuando el usuario acerca el DNI se cambiara el texto por "Leyendo Datos"'indicando al usuario el correcto funcionamiento y si hay algún error durante la lectura se cambiará el texto por el mensaje de error en cuestión.

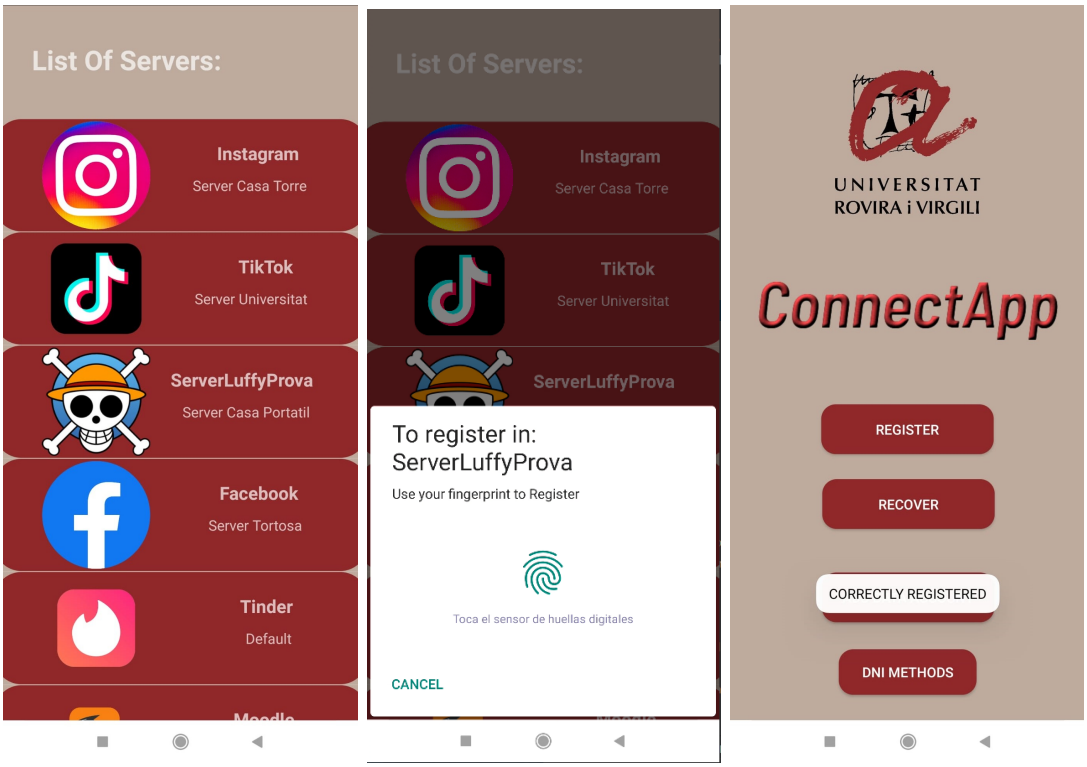
En la figura 13(b) se ha completado la lectura mostrando el numero de DNI, y el usuario tiene que elegir el certificado que va a utilizar para autenticarse en el servidor, una vez elegido se le pedirá la contraseña del certificado (figura 13(c)) y seguidamente se le notificara si quiere realizar una firma con su certificado (figura 13(d)).



((a)) ((b)) ((c)) ((d))

Figura 13: Proceso de lectura del certificado del DNI

A continuación se le muestra al usuario una lista de los servicios online en los que se puede registrar (figura 14(a)), el usuario al elegir un servicio se le pedirá superar una autenticación biométrica (figura 14(b)), al superarse si no hay errores se cambiara a la pantalla inicial de la aplicación con el mensaje de "Correctamente registrado" (figura 14(c)).



((a)) ((b)) ((c))

Figura 14: Proceso de Registro en un servicio

Como resultado podemos observar en la figura 15 que los datos del usuario están correctamente guardados en la base de datos del servicio online.

email	alias	apellidos	certificat_dni	dni	nombre	publickey
asorianofa@gmail.com	alexsoriano	SORIANO FAIGES	MIIHyTCCBbGgAwIBAgIEVUdTYTANBgkqhkiG9w...	49651267-V	ALEX	MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBC...
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 15: Imagen de la base de datos del servidor

También se controlan los errores en el proceso de registro como se puede observar en la figura 16, en la figura 16(a) ocurre cuando en el dispositivo móvil ya tenemos los datos de autenticación guardados en el secure element para autentificar-nos en ese servicio con nuestro usuario, la figura 16(b) ocurre cuando el servidor responde que este usuario ya está registrado, y por último la figura 16(c) cuando no se consigue obtener conexión con el servidor del servicio online.

((a)) ((b)) ((c))

Figura 16: Ejemplos de errores en el proceso de registro

5.2. Prueba de Login en un servicio online

Ya registrados en el servicio vamos a probar a autenticarnos en él utilizando la aplicación, se puede observar en la figura 17 la página web del servicio de ejemplo que utilizaremos, denominado Luffy.

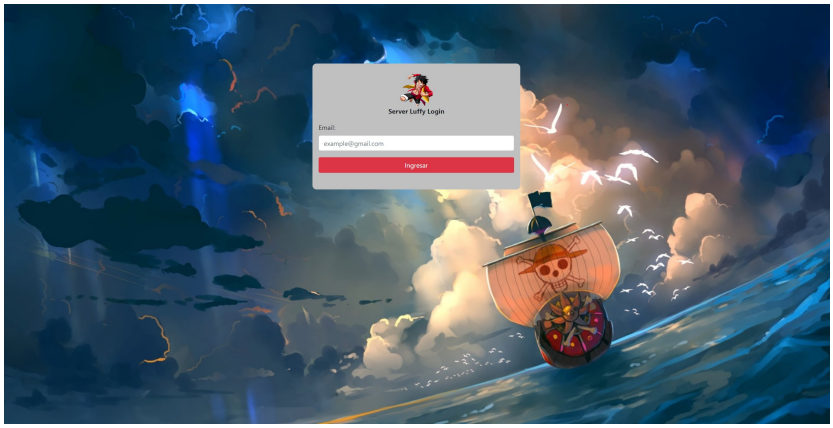
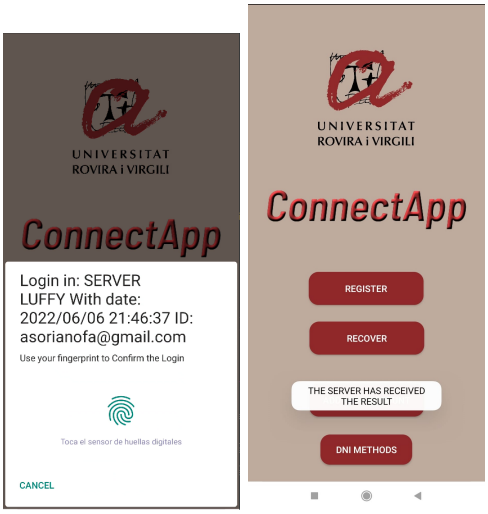


Figura 17: Web del servicio online

Introduciremos el correo con el cual queremos autenticarnos, y en el dispositivo móvil correspondiente obtendremos la petición como se puede observar en la figura 18(a) que una vez contestada nos indicara que ha sido correctamente recibida por el servidor (figura 18(b)).



((a)) ((b))

Figura 18: Petición de Login en la aplicación

Al introducir la huella correctamente, si el dispositivo posee de las credenciales de autenticación correctas habremos logrado autenticarnos correctamente obteniendo el resultado mostrado en la figura 19, se puede observar el alias con el que nos registramos anteriormente en el servidor desde la aplicación.

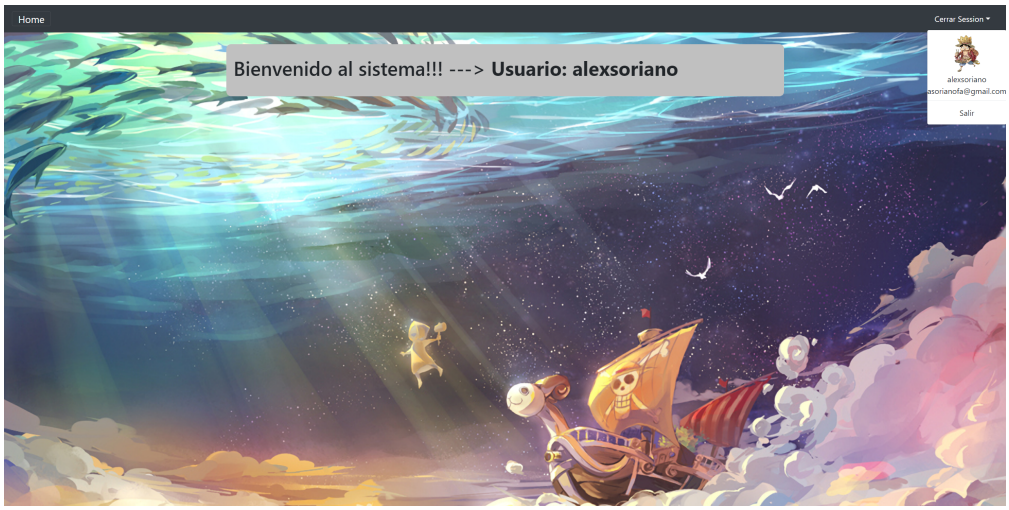


Figura 19: Web del servicio online una vez autenticado

Por el contrario si el usuario rechaza la petición obtendremos el resultado de la figura 20.

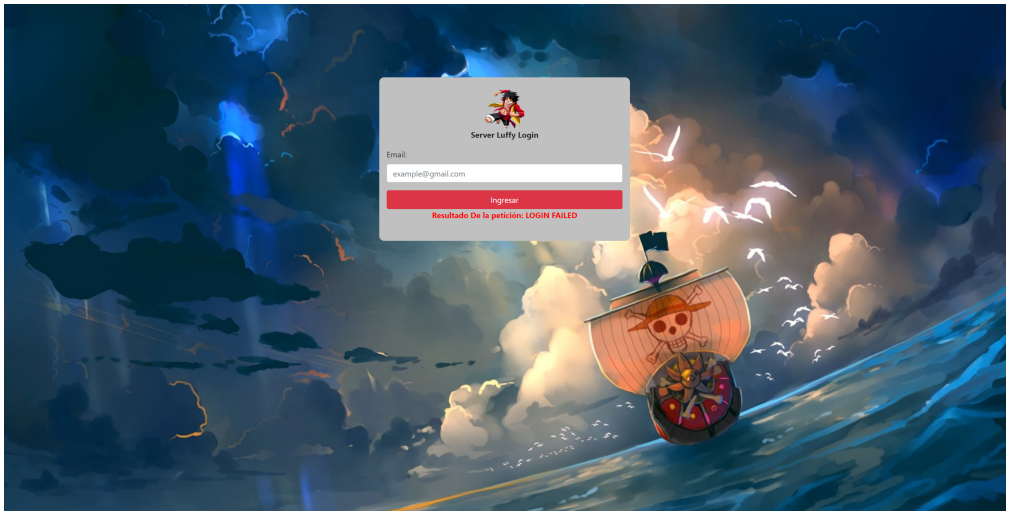
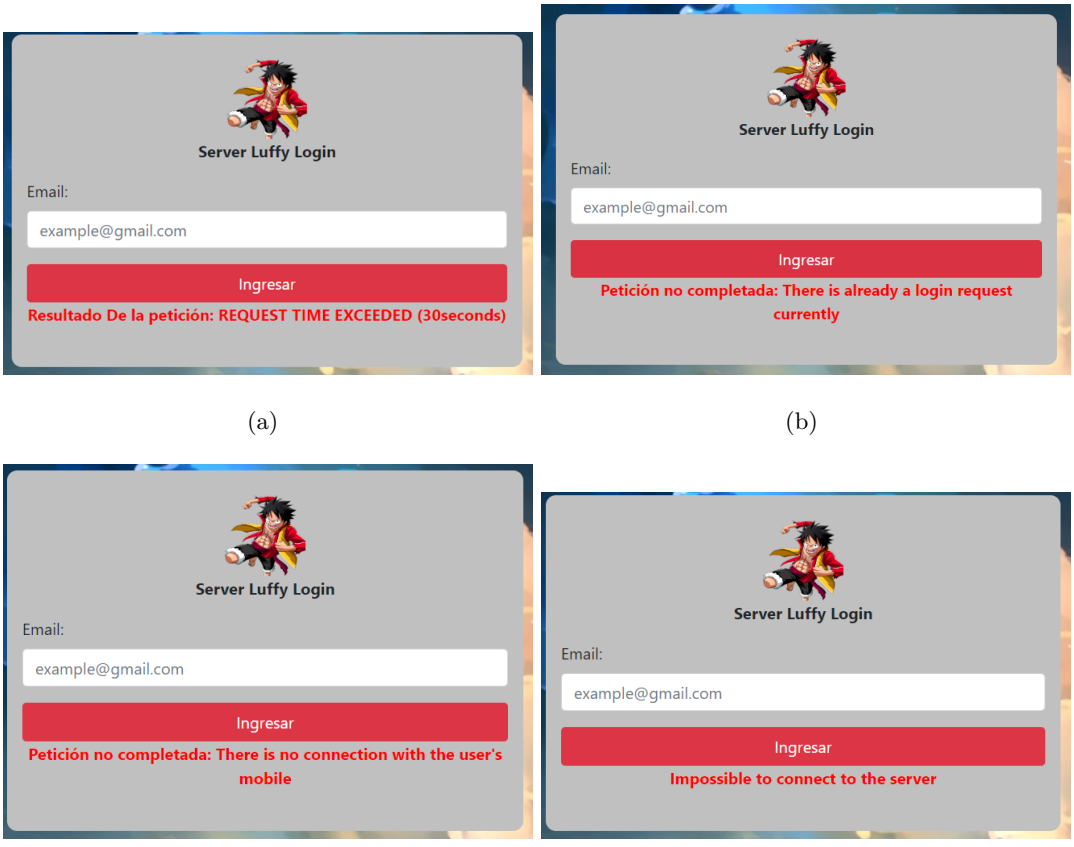


Figura 20: Web del servicio online cuando la petición es rechazada

Como se puede ver en la figura 21 también se controlan diferentes tipos de errores.

- En la figura 21(a) se ha superado el tiempo para responder a la petición por parte del usuario.
- En la figura 21(b) se informa al usuario que ya existe una petición reciente de inicio de sesión en curso.
- En la figura 21(c) se informa al usuario de que el servidor de autenticación no tiene conexión con el dispositivo móvil del usuario.
- En la figura 21(d) se informa al usuario que el servidor de autenticación del servicio online no responde.



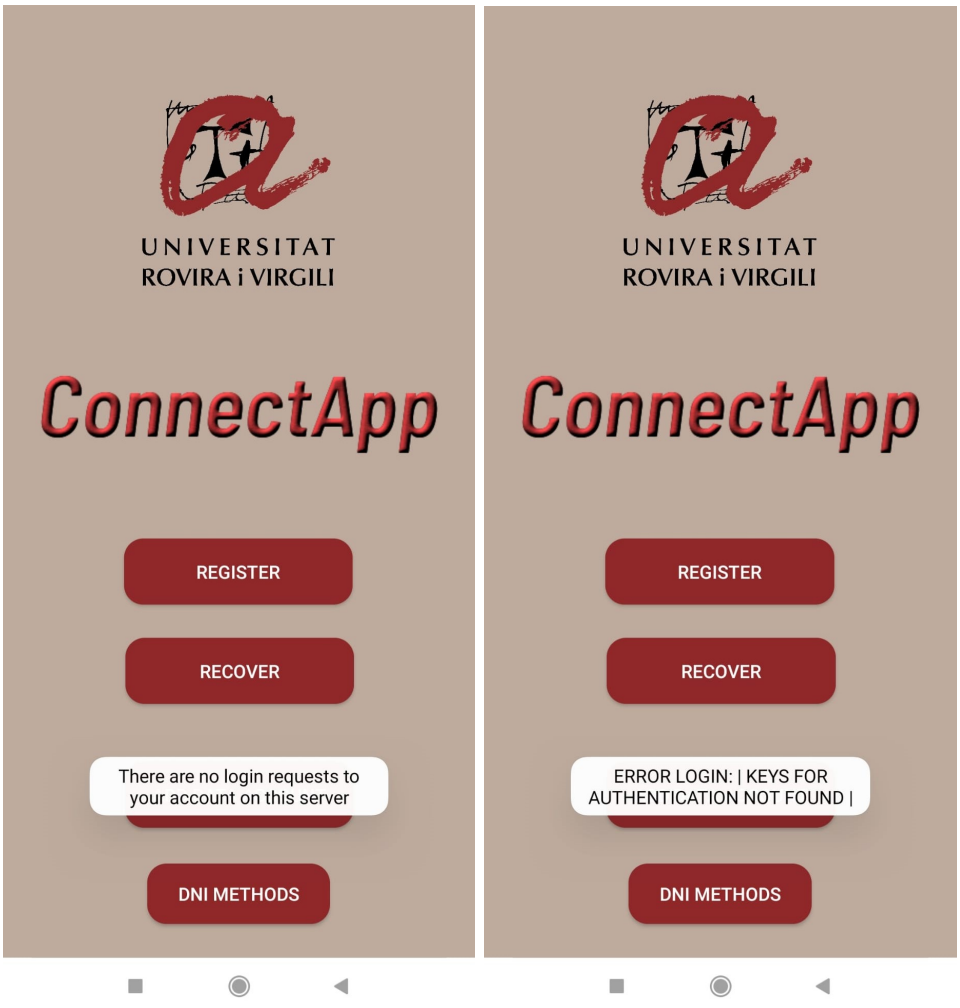
(a) (b)

(c) (d)

Figura 21: Notificaciones de errores en la página web del servicio

Por otra parte, también se notifican errores del proceso de autenticación en el dispositivo móvil del usuario, como se puede observar en la figura 22.

- En la figura 22(a) se informa al usuario que ha contestado a una petición de login que ya no existe en el servidor, debido al tiempo impuesto para la contestación de una petición.
- En la figura 22(b) se informa al usuario de que no tiene los datos de autenticación correspondientes.



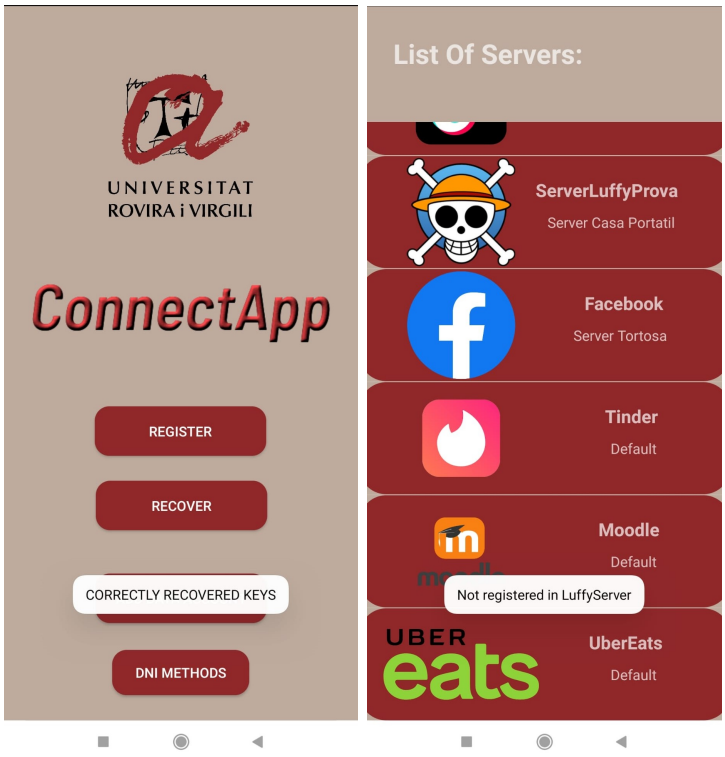
((a)) ((b))

Figura 22: Notificaciones de errores en el dispositivo móvil del usuario

5.3. Prueba de Recover en el servicio online

La prueba de la recuperación de llaves es muy parecida a la funcionalidad comprobada anteriormente de registro.

Se muestra al usuario la lista de servicios online en los que puede recuperar sus credenciales de autenticación (figura 14(a)), seguidamente al seleccionar uno de ellos se le pide al mismo usuario que supere la autenticación biométrica (figura 14(b)), y a continuación se le requerirá la utilización de su DNI descrito anteriormente y observable en la figura 13, terminando con el resultado mostrado en la figura 23(a) si la operación se ha completado correctamente o la figura 23(b) si no ha sido así.



((a)) Recuperación completada ((b)) Recuperación fallida

Figura 23: Respuestas de la aplicación al realizar la recuperación de credenciales en un servicio online

6. Conclusiones

En conclusión, se han obtenido los resultados esperados, el sistema cumple todas las funcionalidades establecidas; la utilización de este sistema facilita el proceso de autenticación y evita tener que recordar las contraseñas, manteniendo la privacidad y seguridad en el proceso.

La elaboración del trabajo ha tenido varios puntos más complicados que otros, inicialmente realizar la lectura tanto de los datos como de los certificados del DNI utilizando NFC, y posteriormente conseguir firmar con ellos. No ignorar tampoco la dificultad de haber diseñado el sistema empleando mecanismos de seguridad tanto como para proteger las comunicaciones como para verificar los mensajes y su respectivo emisor. La mayor dificultad que se ha presentado ha sido conseguir que las estructuras del sistema se comuniquen correctamente para efectuar las diferentes funcionalidades, sobre todo a la hora de establecer la comunicación del servidor de autenticación hacia el dispositivo móvil del usuario.

6.1. Trabajo futuro

La principal tarea que se debería realizar y no se ha efectuado por falta de tiempo, es cambiar la manera en la que se llevan a cabo las notificaciones al cliente, ya que con esta implementación el dispositivo móvil del usuario tiene que tener tantas conexiones abiertas como servicios online tenga para autenticarse; esto provoca que no sea escalable. Para solucionar dicho problema se puede utilizar Firebase para que se encargue de producir las notificaciones; este actuaria como intermediario entre el usuario y los servicios online volviéndose así escalable, también se podría crear un servidor propio que realizara la misma función que el Firebase.

Por otra parte, las notificaciones de autenticación solo se pueden recibir si el usuario está en la pantalla inicial de la aplicación, lo cual limita mucho su uso, se añadiría que la aplicación pudiera recibir notificaciones indistintamente de su estado, incluso en segundo plano.

También se debería verificar el certificado del DNI en la parte del servidor contra la policía, para contrastar su veracidad; lo cual permitiría el hecho de que si un usuario pierde su DNI, no tenga problemas en seguir usando la aplicación y sus cuentas en los diferentes servicios al cambiarlo.

Referencias

[1] Cuerpo Nacional de Policía, v. Sdk dniedroid. https://www.dnielectronico.es/portaldnie/PRF1_Cons02.action?pag=REF_1120. Online.

[2] Farik, M., Lal, N., and Prasad, S. A review of authentication methods. *International Journal of Scientific Technology Research 5* (11 2016), 246–249.

[3] Florencio, D., and Herley, C. A large-scale study of web password habits. In *Proceedings of the 16th international conference on World Wide Web* (2007), pp. 657–666.

[4] mundial de datos, B. Personas que usan internet (% de la población). https://datos.bancomundial.org/indicator/IT.NET.USER.ZS. Online.

[5] Nordpass.com. 2021 most common passwords. https://nordpass.com/es/most-common-passwords-list/. Online.

[6] welivesecurity. La mayoría de los ataques de fuerza bruta buscan descifrar contraseñas cortas. https://www.welivesecurity.com/la-es/2021/11/25/mayoria-ataques-fuerza-bruta-buscan-descifrar-contrasenas-cortas/. Online.

[7] xataca. Artículo sobre tecnología nfc. https://www.xataka.com/moviles/nfc-que-es-y-paraque-sirve. Online.

[8] Zscaler. Uncovering new techniques and phishing attack trends from the cloud. https://www.zscaler.com/blogs/security-research/uncovering-new-techniques-and-phishing-attack-trends-cloud. Online.

[9] Zviran, M., and Haga, W. J. Password security: an empirical study. *Journal of Management Information Systems 15*, 4 (1999), 161–185.