UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

DISEÑO E IMPLEMENTACIÓN DE SISTEMA DE AUTENTICACIÓN DE USUARIOS CONECTANDO BASES DE DATOS PREEXISTENTES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

TOMÁS IGNACIO CORTEZ ENCINA

PROFESOR GUÍA:
Matías Toro Ipinza
Profesor Co-Guía:
Claudio Gutiérrez Gallardo

MIEMBROS DE LA COMISIÓN:
Dionisio González González
Eduardo Graells Garrido

SANTIAGO DE CHILE
2023

DISEÑO E IMPLEMENTACIÓN DE SISTEMA DE AUTENTICACIÓN DE USUARIOS CONECTANDO BASES DE DATOS PREEXISTENTES

En el presente trabajo de título se presenta una problemática que experimenta la empresa BPO Advisors. Esta empresa posee distintos sistemas web con sus bases de datos de usuarios y formas de autorización, como por ejemplo, biometría facial o uso de huella dactilar. La problemática radica en que si una persona es usuario de más de una aplicación no necesariamente tendrán la misma información de autenticación, una puede permitirle a un usuario utilizar biometría facial, pero no tener configurada la biometría por huella dactilar, aunque esa información esté disponible en otra aplicación de la empresa.

Frente a esta oportunidad de mejora se propone unificar las bases de datos de usuarios, junto con la autenticación de estos, en un sistema central para que sea utilizado por todas las aplicaciones que provee la empresa. En la presente memoria se diseñó y desarrolló un sistema de autenticación de usuarios que actúa como proveedor de identidad para utilizar el protocolo Open ID Connect, lo que permite tener un sistema central robusto y que las otras aplicaciones se puedan autenticar de forma segura.

Si bien se logró crear el sistema central de autenticación, no se logró migrar los datos para resolver la problemática inicial, debido a problemas con las librerías utilizadas en la implementación del protocolo OIDC, lo que retrasó el plan de trabajo inicial, por lo que aún se necesitan migrar las bases de datos de usuarios a la aplicación desarrollada para poder resolver la problemática.

Es mejor callar si lo que vas a decir
no es más bello que el silencio.

Hernán Rivera Letelier

Agradecimientos

A mis padres Cecilia y Marcelo, a mis hermanos Nicolás y Baltazar, a mi abuela Laura y a toda mi familia, por su apoyo incondicional y por haberme enseñado los valores y el esfuerzo que me han llevado hasta aquí. A mis profesores guía, por su orientación y por haber dedicado su tiempo para el éxito de este trabajo. A mis compañeros de trabajo y amigos, por su ánimo y por haber compartido conmigo esta etapa de mi vida. Un agradecimiento especial a mis amigos que mantengo desde mi paso por el colegio Kuky, Trini, Zedrik, David, Simón, Mati, Yangzi, Brian, Lukas, ya que me han ayudado a soportar la carga de los estudios, el trabajo y la titulación con su amistad y apoyo. A mis profesores, por su enseñanza y por haber sido una fuente de inspiración para mí. A todas las personas que han colaborado en la realización de esta tesis, por su tiempo y por su valioso aporte.

Tabla de Contenido

Índice de Ilustraciones

Capítulo 1

Introducción

La administración y mantenimiento de múltiples sistemas es una situación que experimentan muchas organizaciones, sin embargo, esto puede escalar generando problemas, siendo uno de los más comunes la necesidad de integrar los sistemas y la falta de interoperabilidad entre las bases de datos de las diferentes aplicaciones.

La empresa BPO Advisors experimenta una situación relacionada a esto, cuentan con distintas soluciones que ofrecen a sus clientes, cada una con su propio servidor y base de datos, además de la información de los usuarios que utilizan la plataforma. Las aplicaciones de BPO Advisors también cuentan con distintos sistemas de autenticación, aparte de la tradicional autenticación con usuario y contraseña, cuentan con autenticación utilizando biometría facial y biometría de huellas dactilares.

La problemática radica en que una persona que tenga cuenta en más de un sistema puede tener la capacidad de autenticarse en uno de estos utilizando biometría facial, mientras que en otro puede utilizar su huella dactilar, pero no viceversa. Esto es así, aunque la información necesaria para realizarlo en ambas aplicaciones esté disponible en las bases de datos de la empresa.

En este contexto, se planteó la necesidad de conectar las bases de datos de usuario de los distintos sistemas de la organización. Para abordar este desafío, se busca unificar los métodos de autenticación en un solo sistema y para esto se decidió implementar un proveedor de identidad y el protocolo OpenID Connect.

Este trabajo de memoria tiene como objetivo principal documentar el proceso de investigación, desarrollo e implementación de un proveedor de identidad y el protocolo OpenID Connect como solución para conectar las bases de datos de usuario de distintos sistemas. Se describirán los fundamentos teóricos y las decisiones tomadas para la elección de tecnologías a usar, así como los detalles de la implementación realizada.

Se comenzó por detallar el problema de la empresa e identificar los objetivos que se desarrollan desde la problemática, siendo el principal contar con un sistema de autenticación general y unificado para todas las aplicaciones que ofrece BPO Advisors a sus clientes.

Con la problemática ya desarrollada se investigó acerca de los distintos protocolos de autenticación federada disponibles como SAML, Kerberos y Open ID Connect, además de las soluciones de organizaciones como Google o Microsoft.

Se comenzó el desarrollo de la solución por la elección del protocolo de autenticación y las tecnologías que se utilizaron en el desarrollo de la aplicación, siendo la elección más importante el protocolo Open ID Connect. Ya con la decisión tomada se llevó a cabo el desarrollo de la solución que, aunque se presentaron problemas, se logró completar cumpliendo lo principal.

Luego de esto se evidenció la validación que se llevó a cabo para poner a prueba el éxito del desarrollo de la solución, demostrando las formas en que la solución cumple ciertas medidas de seguridad y la retroalimentación de voluntarios que hicieron uso de aplicación.

Por último, se hizo una retrospectiva del trabajo realizado, se explicitaron los aprendizajes obtenidos y el trabajo a futuro que es necesario llevar a cabo para dar un término a la problemática de la empresa. Si bien no se logró unificar las bases de datos, sí se pudo crear el proveedor de identidad con Open Id Connect, por lo que parte del trabajo a futuro a desarrollar es la migración de las bases de datos a este nuevo sistema.

Se espera que este trabajo sea de interés para aquellas organizaciones que se enfrenten a problemas similares y que puedan beneficiarse de la solución propuesta y los aprendizajes que se obtuvieron luego de la realización de este trabajo.

Capítulo 2

Problema

En este capítulo se explicará el problema principal de la memoria y el contexto en el que se desarrolla. Se presenta la empresa BPO Advisors para luego explicar el problema que experimentan, los objetivos que se buscan cumplir y la metodología que se pretende seguir para cumplirlos.

2.1. Contexto

El desafío que se presenta en este documento fue dado a conocer por la empresa BPO Advisors. BPO Advisors es una empresa chilena dedicada a optimizar procesos de gestión de negocios, identificación digital y firma electrónica de personas, empresas y organizaciones. BPO cuenta con distintos clientes, los cuales en su mayoría son empresas que necesitan servicios de firma electrónica. BPO ofrece soluciones informáticas a los clientes y estas son, por lo general, aplicaciones web con las características necesarias para satisfacer los requerimientos estipulados. Estas aplicaciones están alojadas en servidores provistos y administrados por BPO, los cuales tienen una base de datos con la información de los usuarios y otros datos pertinentes.

Un ejemplo de las aplicaciones que BPO dispone a sus clientes, es un gestor de documentos que permite crear, administrar y firmar documentos con firma electrónica simple. Cada cliente cuenta con una implementación propia de esta aplicación según lo requerido y cuenta con su servidor dedicado (a esto se le llama instancia). Los usuarios de estas aplicaciones son, en su mayoría, integrantes de la empresa cliente, por ejemplo, trabajadores que van a firmar documentos para la empresa. Otra de las aplicaciones que ha desarrollado BPO es Firma Ya, que se utiliza para firma electrónica avanzada de documentos, donde los usuarios son personas naturales que se registran en la página y hacen uso de ésta, además de empresas y los integrantes de estas.

2.2. Problemática

BPO cuenta con diversos clientes los cuales utilizan distintas instancias de las aplicaciones de la empresa, esto se traduce en más de 500.000 usuarios registrados en estas aplicaciones, en distintas bases de datos y en distintos servidores. Estos usuarios tienen diversas formas de autenticarse en

las aplicaciones ofrecidas por BPO dependiendo de los requisitos del sistema, por ejemplo, se podría utilizar las credenciales de la página de Firma Ya, biometría facial o huella digital. Estos métodos de autenticación ya están implementados en las aplicaciones de BPO y se encuentran en uso actualmente.

Debido a que una misma persona puede ser usuario de distintas aplicaciones de BPO Advisors, se puede dar la situación en la cual esta persona se pueda autenticar mediante su huella digital en uno de los servicios y utilizar su biometría facial en otro, pero no viceversa. Esto debido a que las bases de datos de ambas aplicaciones no están integradas actualmente, por ende, un usuario no puede escoger el mejor método de autenticación en toda ocasión, aunque en alguna de las otras aplicaciones estén los datos necesarios para hacerlo.

Por lo tanto, la situación actual de la empresa es tal que, la información de un usuario específico, incluyendo la necesaria para la autenticación, se encuentra entre las distintas aplicaciones de los clientes y no hay comunicación entre ellas. En el caso de que una persona cree un usuario en el gestor documental de una empresa y también en Firma Ya, la información entregada será tomada como nueva para ambos casos. Esto conlleva a que las dos bases de datos tengan información redundante en el mejor de los casos o inconsistente en el peor, aunque ambas bases de datos sean de propiedad y administración de BPO.

Específicamente, la empresa cuenta con medios de autenticación como reconocimiento por biometría facial y biometría de huella dactilar. Estos métodos son provistos por una empresa externa a BPO, lo que BPO realiza es la comunicación con sus procesos y el almacenamiento de datos necesarios, por ejemplo la referencia al usuario correspondiente, el tipo de biometría y un token para el proceso. Además de estos métodos de autenticación también está el principal por contraseña y se esperaba utilizar autenticación por clave única, pero al final se debió descartar esta opción.

La situación anterior presenta una oportunidad de mejora para la empresa, ya que el poder tener mayor disponibilidad de los servicios ya establecidos de autenticación es favorable tanto para los usuarios como para la empresa y sus clientes. Para los usuarios, tener más opciones facilita y agiliza el proceso para autenticarse, mientras que para la empresa y sus clientes entrega más seguridad de que se está dando acceso a la persona adecuada, vale decir, hay más seguridad de que la autenticación cumplirá su función correctamente.

Otra razón para llevar a cabo un proyecto que mejore la situación descrita, es tener más información del usuario para gestión y envío de información. Esto se traduce en compartir la información entre las aplicaciones de BPO Advisors para que la persona pueda agilizar el proceso de registro de usuario en la plataforma. Datos como su correo o número telefónico pueden ser entregados a la aplicación para que pueda rellenarse en el momento del registro y pedir la confirmación por parte del usuario de que son los correctos y quiere que sean usados en la plataforma.

Con esto en mente se planea integrar todos los sistemas de autenticación en uno solo con su propia base de datos de usuarios, para que así se resuelva la problemática planteada.

Cabe recalcar que este proyecto no se hace con el fin de vender los datos a una empresa externa, esto es solo para mejorar la experiencia ya entregada, como se mencionó anteriormente. La información pertinente para este proyecto ya se encuentra en los servidores de los productos de BPO y seguirá de esa forma, solamente que estará agrupada en un nuevo servicio que se utilizará para la autenticación.

Si bien ya existen métodos de autenticación estándar y software comercial para autenticación, estos no son suficientes para suplir las necesidades que iniciaron la idea de esta memoria. El software necesario debe adecuarse al contexto específico de este proyecto y para eso se debe tener en cuenta diversos factores, por ejemplo que esto es para una empresa chilena y que se deben tener en consideración datos como el RUT o pasaporte. Además, se debe tener en cuenta que la autenticación no es el principal problema de este proyecto (esto ya está hecho por parte de la empresa), sino que se debe tener un sistema que dé la posibilidad de unificar las bases de datos de usuarios y los métodos de autenticación y así permita a los usuarios tener más opciones a su disposición.

En resumidas cuentas, la empresa BPO Advisors tiene diversas aplicaciones con usuarios que deben autenticarse, pero estas aplicaciones no comparten la información de los usuarios, pudiendo llevar a inconsistencias o repetición de los datos guardados. Por esta razón, se investigarán opciones de protocolos de autenticación para tener un sistema que permita integrar lo necesario de las instancias que se tienen y así resolver el problema.

2.3. Objetivos

A continuación se listan los objetivos del proyecto, tanto generales como específicos.

2.3.1. Objetivo General

El objetivo principal que se busca lograr es tener un sistema de autenticación general y unificado para todas las aplicaciones que ofrece BPO Advisors a sus clientes. De esta forma se podrá tener la información de usuarios en un solo sistema. Por lo tanto, para unificar las bases de datos de usuario, primero se buscará unificar la autenticación y para eso se debe tener un sistema que permita administrar la autenticación y base de datos de usuario.

2.3.2. Objetivos Específicos

1. Diseñar el esquema lógico adecuado para el modelo de usuario y su información de autenticación, además de asegurar que sea compatible con las aplicaciones en las que se usará.
2. Diseñar el sistema de autenticación que recibirá las peticiones de autenticación, para luego procesarlas e indicar si fueron exitosas.

3. Explorar las formas en que se puede desarrollar la autenticación con los métodos ya presentes.

4. Como una extensión del sistema principal, se busca permitir que el sistema de autenticación entregue información extra de un usuario a las aplicaciones que lo utilicen. Por ejemplo, una vez que un usuario esté autenticado en la aplicación, se le puede indicar que ya se tiene la información de su número de teléfono y se puede utilizar.

2.4. Metodología

La metodología que se utiliza en el desarrollo de esta memoria fue llevada a cabo con un modelo de cascada. Se comenzó por analizar la situación actual, reconocer requisitos y luego crear un plan de acción que lleve al desarrollo de la solución y a su validación. Esta elección de metodología se hizo debido a que el proyecto tiene fases bien definidas, así se puede avanzar de forma directa hacia las fases. También esto facilita que se pueda visualizar el desarrollo del proyecto cronológicamente, lo que ayuda a la confección de este documento y el desarrollo del proyecto.

Capítulo 3

Conocimiento previo y estado del arte

En la siguiente sección se habla acerca de tres aristas importantes para este proyecto: conceptos necesarios para comprender de lo que se habla, protocolos de autenticación para conocer lo que existe actualmente y el estado del arte respecto a la autenticación, para así tener una visión de lo que se utiliza en el mercado.

Con más detalle, se introducen los conceptos de entidad, identidad, autorización y autenticación, además de cómo se relacionan y distinguen entre ellos. También se habla de protocolos relacionados con la autenticación y autorización, siendo estos: OAuth, OpenID Connect, SAML, Kerberos y RADIUS. Luego de esto, se abordan los sistemas de autenticación que ofrece Google y Microsoft para desarrolladores. Por último se hace un análisis en profundidad del protocolo Open ID Connect

3.1. Conceptos y definiciones

A continuación se presentan los conceptos y definiciones que se estiman necesarios para la comprensión de la problemática planteada.

3.1.1. Autenticación

La autenticación es el proceso de confirmar la identidad de una entidad (ej.: un usuario) [1] o demostrar que es quien dice ser. El proceso de autenticación usualmente confía en algún tipo de prueba. También se puede encontrar en la literatura abreviada a AuthN [2].

3.1.2. Autorización

La autorización se refiere al proceso de verificar qué entidades pueden tener acceso a algo o qué acciones pueden llevar a cabo, también se puede entender como el acto de conceder a una parte autenticada permiso para hacer algo o acceder a algo [2]. Especifica a qué datos se puede acceder y qué se puede hacer con ellos [1]. También se puede encontrar en la literatura abreviada a AuthZ [2].

3.1.3. Entidad e Identidad

Se suele hablar de entidad e identidad [1] en temas de autenticación, sin embargo, muchas veces estos términos se confunden. Entidad se refiere a algo que existe como una unidad individual, el ejemplo más común es una persona que interactúa con un *software*. Mientras que identidad se refiere al conjunto de atributos que se define para distinguir a una entidad en un cierto contexto.

3.2. Protocolos relacionados

Si se investiga acerca de protocolos para autenticación y autorización se puede encontrar que existen varios protocolos hasta la fecha, si bien no todos son para aplicaciones web es conveniente mencionarlos para poder distinguir los protocolos existentes. Una comparación de estos protocolos se hace en la sección 4.1.2.3.

3.2.1. OAuth

OAuth 2.0 [3] es un protocolo abierto para permitir autorización segura con un método simple y estándar desde web, móviles y aplicaciones de escritorio. Se utiliza principalmente para que los usuarios de alguna aplicación web puedan entregar permisos a otras aplicaciones o sitios web para acceder a su información, pero sin tener que entregar sus credenciales [4].

3.2.2. Open ID Connect

De palabras de la propia OpenID Foundation: OpenID Connect 1.0 (OIDC) [5] es una simple capa de identidad sobre el protocolo OAuth 2.0. Permite a los clientes verificar la identidad del usuario final basado en la autenticación ejecutada por un Servidor de Autorización, así como para obtener información básica del perfil del usuario final de una manera interoperable y que utilice el esquema REST.

OpenID Connect [6] permite a clientes de todos los tipos, incluyendo usuarios móviles, de web y clientes de JavaScript, pedir y recibir información acerca de sesiones autenticadas y usuarios finales. El conjunto de especificaciones es extensible, permitiendo a los participantes usar características opcionales como encriptación de los datos de identidad, poder descubrir proveedores de OpenID y administración de sesiones.

Cabe destacar que también existe OpenID 2.0 [6] el cual no debe confundirse con OIDC. Open ID Connect es la iteración próxima a OpenID 2.0, si bien OIDC ejecuta muchas de las mismas tareas que OpenID 2.0, Open ID Connect las hace de una manera que se integra de mejor manera con una API y que es utilizable por aplicaciones móviles. Además, OIDC tiene mecanismos opcionales para encriptación y para mayor robustez al registrarse. Por otro lado, mientras que la integración de OAuth 1.0 y OpenID 2.0 requiere una extensión en OIDC, estas capacidades vienen integradas en el protocolo. En otras palabras, Open ID Connect es la nueva versión de OpenID 2.0 y es preferible usar este protocolo más reciente.

El flujo básico de Open ID Connect [1] es el siguiente:

- Los visitantes solicitan a la aplicación empezar el proceso de autenticación.
- La aplicación redirecciona a los visitantes al proveedor de identidad.
- Los visitantes se autentican con el proveedor de identidad.
- El proveedor de identidad redirecciona de vuelta a la aplicación con tókenes.
- La aplicación utiliza los tókenes para emitir una solicitud al proveedor de identidad para terminar el proceso de autenticación.
- La aplicación indica a los visitantes que iniciaron sesión.

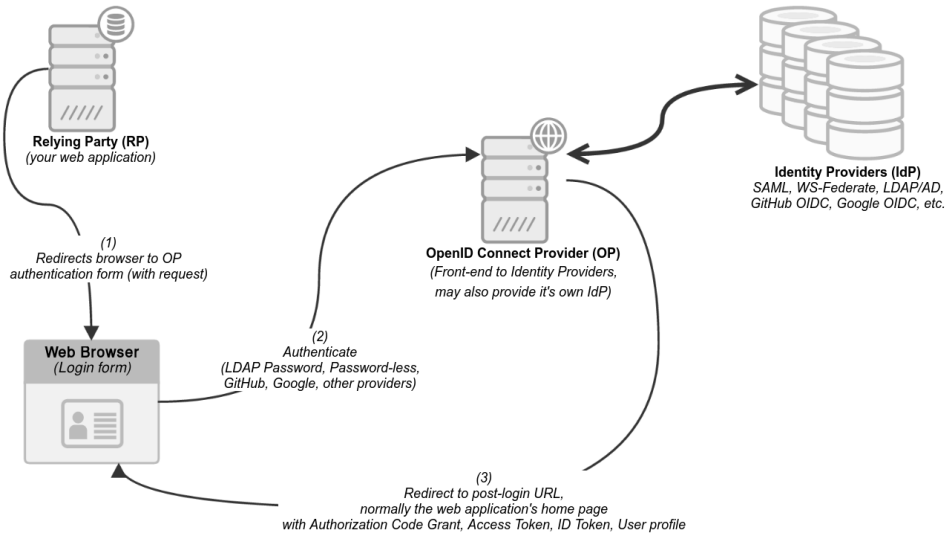

Figura 3.1: Esquema del funcionamiento de protocolo OIDC. [7]

3.2.3. SAML

El lenguaje de marcado para confirmaciones de seguridad (SAML) [8] es un estándar XML con el que los dominios web seguros pueden intercambiar datos de autenticación y autorización de los usuarios. Mediante SAML, un proveedor de servicios online puede ponerse en contacto con un proveedor de identidades online para que autentique a los usuarios que intenten acceder a contenido seguro.

El lenguaje de marcado para confirmaciones de seguridad (SAML) es un estándar de federación abierta que permite a un proveedor de identidad (IdP) autenticar usuarios y, a continuación, transferir un token de autenticación a otra aplicación conocida como proveedor de servicios (SP). SAML permite que el SP funcione sin la necesidad de realizar su propia autenticación y transferir la identidad para integrar usuarios internos y externos. Permite compartir credenciales de seguridad

con un SP a través de una red, normalmente una aplicación o un servicio. SAML permite una comunicación segura entre dominios entre la nube pública y otros sistemas habilitados para SAML, así como un conjunto concreto de sistemas de gestión de identidad distintos ubicados en entornos locales o en una nube diferente.

SAML se refiere al lenguaje de variante XML utilizado para codificar esta información y también puede abarcar varios mensajes y perfiles de protocolo que forman parte del estándar.

3.2.4. Kerberos

Kerberos [9] es un protocolo de seguridad creado por MIT que usa criptografía de claves simétricas para validar usuarios con los servicios de red, evitando así tener que enviar contraseñas a través de la red. El primer objetivo de Kerberos es el de eliminar la transmisión a través de la red de información de autenticación. Un uso correcto de Kerberos erradica la amenaza de analizadores de paquetes que intercepten contraseñas en su red.

3.2.5. RADIUS

RADIUS [10] (*Remote Authentication Dial In User Service*) es un protocolo estándar de Internet que proporciona servicios centralizados de gestión de autenticación, contabilidad e IP para los usuarios de acceso remoto en una red de acceso telefónico distribuida.

El modelo cliente-servidor de RADIUS tiene un servidor de acceso a red (NAS) que funciona como cliente para un servidor RADIUS. El sistema, al actuar como NAS, envía información de usuario y conexión a un servidor RADIUS designado, mediante el protocolo estándar de RADIUS definido en la RFC 2865.

3.3. Estado del arte

Actualmente, se tienen soluciones a problemas similares entregados por empresas de tecnología, en esta sección se ven ejemplos de esto.

3.3.1. Google

Google ofrece Identity Platform [11] es una plataforma de administración de identidades y accesos de clientes (CIAM) que ayuda a las organizaciones a incorporar estas funciones en sus apps, proteger las cuentas de los usuarios y escalar con confianza en Google Cloud.

Identity Platform proporciona servicios de *backend* y funciona con *SDK* y bibliotecas de elementos de interfaz de usuario ya hechas para autenticar a los usuarios de la app en la que se integrará. Admite la autenticación mediante contraseñas, números de teléfono, proveedores de identidad federada populares, como Google, Facebook, Twitter y cualquier proveedor que admita el protocolo SAML u OpenID Connect.

Identity Platform se integra estrechamente a los servicios de Google Cloud y aprovecha los estándares de la industria como OAuth 2.0 y OpenID Connect. Para que un usuario acceda a una app que integre la plataforma, primero se debe obtener credenciales de autenticación del usuario. Estas credenciales pueden ser la dirección de correo electrónico y la contraseña del usuario, una aserción de SAML o un token de OAuth de un proveedor de identidad federada. En el caso de los proveedores de identidad federada, los proveedores muestran esos tókenes al controlador de autenticación de Identity Platform. Google aloja este servicio, por lo que la app que integre la plataforma no tiene que recibir y validar el artefacto de autenticación. Después de recibir los tókenes, los servicios de *backend* de Google los verificarán y mostrarán una respuesta al cliente.

Después de acceder de forma correcta, se puede acceder a la información de perfil básica del usuario y controlar el acceso del usuario a los datos almacenados en Google Cloud o en otros productos. También se puede utilizar el token de autenticación proporcionado para verificar la identidad de los usuarios en los propios servicios de *backend* de la app que integre la plataforma.

3.3.2. Microsoft

Microsoft provee al público una plataforma de desarrollo llamada plataforma de identidad de Microsoft [12]. El objetivo de esta plataforma es ayudar a crear aplicaciones para que los usuarios y clientes de esta puedan iniciar sesión usando sus identidades de Microsoft o cuentas de redes sociales y proveer acceso autorizado a las API de Microsoft u otras provistas.

Para los desarrolladores, Microsoft identity platform ofrece integración de innovaciones modernas en el ámbito de la identidad y seguridad, como la autenticación sin contraseña, la autenticación de nivel superior y el acceso condicional. Para esto no es necesario implementarlo manualmente, ya

que aplicaciones integradas con la Plataforma de identidad de Microsoft nativamente toman ventaja de estas innovaciones.

Uno de los rasgos principales de la plataforma anterior es permitir que las apps que la utilicen puedan delegar la responsabilidad a un proveedor de identidad centralizado. Azure Active Directory (Azure AD) es un proveedor de identidad centralizado en la nube. Delegar autenticación y autorización a este permite escenarios como: Políticas de acceso condicional que requieren al usuario estar en un lugar específico, el uso de autenticación multifactor y/o permitir al usuario final iniciar sesión una vez y luego lo haga de forma automática en todas las apps que compartan el mismo directorio centralizado, en otras palabras inicio de sesión único (SSO).

Microsoft identity platform provee identidad como un servicio con el fin de simplificar la autorización y autenticación para desarrolladores de aplicaciones. Soporta protocolos de estándar industrial y librerías de código abierto para diferentes plataformas.

La plataforma utiliza OAuth para la autorización, mientras que para la autenticación utiliza tanto OpenID Connect como SAML. La autenticación SAML se usa normalmente con proveedores de identidades, como Servicios de federación de Active Directory (AD FS) federados con Azure AD, por lo que se usa con frecuencia en aplicaciones empresariales. OpenID Connect se usa normalmente con las aplicaciones que se encuentran exclusivamente en la nube, como aplicaciones móviles, sitios web y API web.

3.4. Open ID Connect en profundidad

Como se mencionó anteriormente Open ID Connect u OIDC, es una protocolo de autenticación, el cual está desarrollado sobre OAuth 2.0 y en esta sección se verá en mayor detalle cómo funciona este protocolo.

3.4.1. Entidades y conceptos relacionados

Para entender mejor el protocolo se explicarán los agentes principales que participan y distintas formas de referirse a estos.

Aplicación Cliente: [7] también conocida como *Relying Party*, RP o solamente cliente, es una aplicación web que busca autenticarse y eventualmente acceder a más datos.

Aplicación Proveedora: [7] también conocida como *OIDC Provider*, proveedor de OIDC, OP, IP o solamente proveedor, es un sistema que provee de autenticación y autorización a las aplicaciones clientes. Puede depender de sí misma u otro proveedor OIDC.

Proveedor de identidad: [7] también conocida como IdP, es una base de datos con las cre-

denciales de usuario y atributos con información. La aplicación proveedora también puede actuar como proveedor de identidad.

Scopes: [7] también conocido como *roles* o permisos, son información de acceso de control que utiliza el cliente para darle permisos a un usuario.

Pairwise Pseudonymous Identifiers: [13] abreviado PPID es un identificador que identifica la entidad a una aplicación cliente que no puede ser correlacionada con la PPID de la entidad en otra aplicación cliente. Esto permite una nueva capa de privacidad en el protocolo.

Sector Identifiers: [13] son una agrupación de clientes que comparten el mismo PPID de un usuario en caso de que haya necesidad de que un usuario pueda correlacionarse entre distintos clientes.

JWT: [7] abreviación para *JSON Web Tokens* son documentos JSON que representan un mensaje posiblemente firmado. El mensaje contiene el emisor del token, el sujeto(usuario del token), cómo el usuario se ha autenticado y para quién está dirigido el token.

Access token: [7] o token de acceso es un *string* que contiene un único token secreto, contiene permisos específicos y es usado para obtener datos desde una API, expira rápidamente.

Refresh token: [7] es un *string* que contiene un único token secreto, este token permite a su poseedor pedir y obtener nuevos tókenes de acceso. Estos nuevos tókenes tienen un subconjunto de permiso que el *refresh token* tiene, este token nunca expira.

ID token: [7] Es un JWT el cual es creado y firmado por el proveedor de OIDC y consumido y verificado por la aplicación cliente autenticando usuarios. Este token contiene información acerca de cómo y cuándo el usuario se autenticó junto con otros atributos varios.

3.4.2. Authorization code flow

En la figura 3.2 se tiene con mayor detalle cómo funciona el protocolo OIDC, en lo que se denomina como *Authorization code flow* [7] o flujo del código de autorización, se dividió el flujo en partes numeradas para mejor explicación.

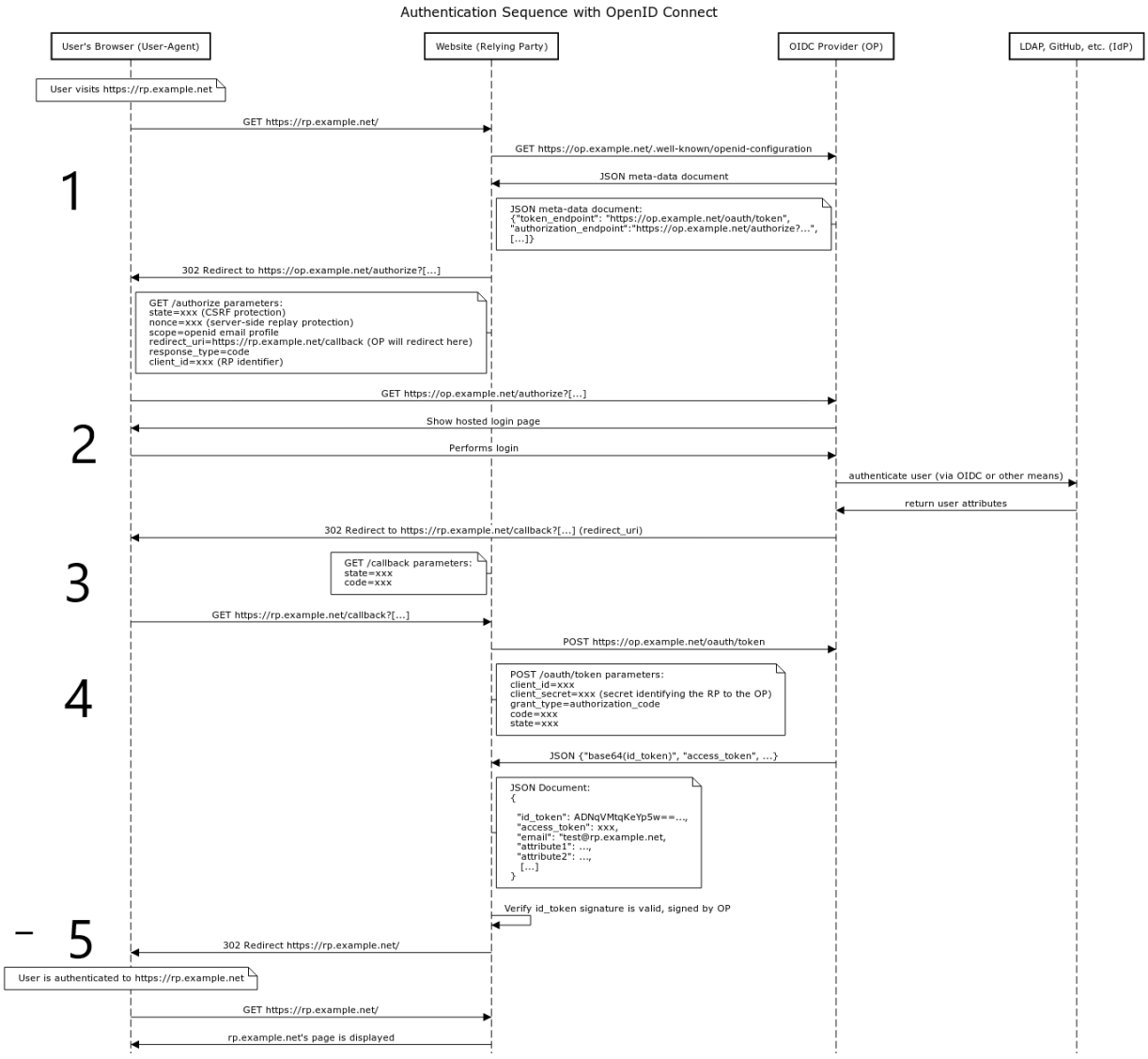

Figura 3.2: OIDC Authorization Code Flow. [7]

Como se ha dicho, OIDC es un protocolo que permite a las aplicaciones clientes autenticar a sus usuarios con un servidor externo llamado proveedor de OIDC. Este servidor obtiene los datos de un proveedor de identidad.

Cuando el usuario quiere iniciar sesión en la aplicación cliente, esta comienza el flujo del código de autorización, indicado por el número 1 en la figura 3.2

En una primera instancia el cliente obtiene los datos necesarios del flujo mediante una petición GET al proveedor de OIDC y así obtiene la información de las direcciones de los *endpoint* pertinentes.

La segunda división inicia con el navegador siendo redireccionado hacia el proveedor de OIDC (dirección /authorize), donde, si no está autenticado, debe hacerlo. Esta autenticación puede hacerse

por los medios que se estimen convenientes, además en esta sección del flujo se deben escoger los *scope* y así conferir los permisos necesarios.

Luego de que el usuario acepta lo anterior, se inicia la tercera división en la cual el navegador es redirigido hacia el cliente (dirección /callback) junto con un *authorization code*, terminando así con la interacción que el usuario tiene con el flujo.

En el punto 4 el cliente hace una petición de los tókenes al proveedor haciendo uso del código de autorización y además de su ID de cliente y secreto de cliente, previamente acordado con el proveedor. Luego de que el proveedor verifica que el ID y el secreto son los correctos, entrega un *ID Token* y *Authorization Token* a la app cliente.

Por último, en el punto 5 el cliente verifica que el JWT del *ID Token* sea válido y con firma correcta (se utiliza encriptación asimétrica) y una vez que todo está validado se puede autenticar al usuario en la aplicación y se le indica esto en el navegador terminando así el flujo. Como detalle extra, en algunas implementaciones, las llaves necesarias para validar el JWT se obtienen del cliente mediante un endpoint JWKS.

3.4.3. Otros flujos

Adicionalmente, al flujo principal existe el flujo implícito y el flujo híbrido. En el flujo implícito [14] se envía tanto el *ID Token* como el *Autorization Token* hacia el navegador. Esto puede servir en aplicaciones de una página (SPA), pero tiene ciertos peligros, debido a que la aplicación es vulnerable frente al robo de tókenes, los cuales pueden ser por CSRF o XSS [7].

Mientras que el flujo híbrido [14] tiene ambas opciones disponibles, puede enviar sus tókenes de la forma en que lo hace el flujo de *Authorization code* como el flujo implícito, llevándolo a tener las mismas vulnerabilidades que este último.

Capítulo 4

Solución

En este capítulo se abordará el diseño e implementación de la solución, además de los obstáculos que surgieron en el transcurso del proyecto. Se explicará la idea principal que se tiene para la solución junto con las tecnologías a usar y lo que esta debe cumplir. Luego se explicará el razonamiento para las elecciones que se hizo respecto al protocolo. Por último, se detallará el proceso de desarrollo y el producto obtenido de este.

4.1. Diseño de la solución

El diseño de la solución para cumplir los objetivos se puede abordar desde tres aristas, el protocolo de autenticación a utilizar, la base de datos de usuarios y el sistema que autentique a los usuarios y administre la base de datos. Teniendo esto en cuenta, se propone diseñar una aplicación web que jugará el rol de proveedor de identidad con su base de datos asociada y que permita la autenticación de usuarios en las aplicaciones haciendo uso del protocolo Open ID Connect.

4.1.1. Tecnologías para la solución

Ahora se presentan las tecnologías que se utilizarán para llevar a cabo la solución.

Respecto a la base de datos, ya que no se espera que ésta sufra cambios luego de su implementación y debido a que se busca tener un sistema que pueda ser resistente al estrés de múltiples usuarios, se propone utilizar una base de datos relacional. Entre sistemas gestores de bases de datos relacionales se postula PostgreSQL [15], debido a su diseño adecuado a ambientes de altos volúmenes, a la rapidez de sus transacciones en estos ambientes, a la robustez que presenta y a que es gratuito. Además, es una base de datos, la cual ya ha sido utilizada en proyectos previos de la empresa.

Para poder administrar la base de datos de usuarios y ejecutar toda la lógica de la autenticación, se propone crear una aplicación web que permita mostrar la interfaz de autenticación y poder hacer las verificaciones y cambios necesarios a la base de datos, cumpliendo el rol de proveedor de identidad. Para implementar el sistema se propone *Ruby* con el *framework Ruby on Rails*. Se escoge

este lenguaje y *framework* pensando en el soporte que luego deberá llevar a cabo la empresa, ya que estas tecnologías han sido utilizadas por BPO Advisors, y por ende, el costo de tener gente que pueda darle soporte a la solución es menor.

Por último, el protocolo de autenticación que se utilizará será Open ID Connect, esto permitirá que la comunicación de las aplicaciones a la solución se haga mediante una API REST, para que así la conexión sea independiente de la implementación de la solución y para que también sea sencilla de configurar y así los cambios que se deban llevar a cabo en las aplicaciones de los clientes sean acotados.

4.1.2. Open ID como solución al problema

En esta sección se explica a mayor profundidad por qué se selecciona OIDC como el protocolo de autenticación para el proyecto.

4.1.2.1. Por qué OIDC es beneficioso para el proyecto

Dos problemáticas a resolver en el contexto del proyecto son las siguientes: Se puede tener usuarios que crean una nueva cuenta para uno de los servicios de la empresa, siendo que ésta ya tiene en otro de sus servicios los datos de inicio de sesión de ese usuario. Por otro lado, se tienen las bases de datos de usuarios separadas, lo que crea problemas de mantención y seguridad. Ambos problemas tienen solución al delegar la autenticación de usuario a un servicio dedicado y hecho para esto, llamado proveedor de identidad o proveedor de OpenID en el caso de este protocolo. Con esto en cuenta, OIDC se presenta como el estándar de proveedor de identidad a elegir para este proyecto.

De manera general, las ventajas de utilizar OIDC son su seguridad y simplicidad de mantención, además de la capacidad de utilizarse desde aplicaciones web y móviles. También las respuestas y peticiones del protocolo tienen un formato de JSON, el cual es legible por humanos y funciona bien en operaciones de intercambio de datos [16].

Teniendo en cuenta aspectos específicos para este proyecto, una de las ventajas que tiene OIDC es que no define cómo se lleva a cabo la autenticación, pero provee una regla estandarizada para pedirla y cómo el resultado de esta autenticación se debe presentar al cliente, ya que para este proyecto ya se cuentan con los distintos métodos de autenticación es una ventaja utilizar OpenID Connect. Además, el hecho de que facilita el uso para API y que utilice el formato JSON para las peticiones es ventajoso para el proyecto, pues son dos aspectos con los que se tiene familiaridad en la empresa.

Como dato final se debe tener en consideración que empresas y organizaciones como Google, Microsoft, Yahoo! Japón, y el gobierno de Chile, con la clave única, utilizan el protocolo en aplicaciones que han desarrollado, lo que demuestra que es un protocolo confiable y que en caso de que

se presente una brecha de seguridad, las probabilidades de que se detecten y corrijan a tiempo son altas.

4.1.2.2. Librerías y documentación

Otro factor importante a tener en consideración es si se encuentran disponibles librerías relacionadas con el protocolo, ya que para estos proyectos en los que la seguridad es tan importante un error en la implementación puede llevar a una vulnerabilidad en la aplicación.

La fundación OpenID cuenta con implementaciones certificadas [17] en su página web para distintos lenguajes de programación y *frameworks* de estos. La certificación de OpenID [18] es una declaración formal por una entidad de que la puesta en marcha de un producto o servicio se ajusta a un perfil de conformidad específico del protocolo OpenID Connect, además de superar un conjunto de pruebas provistas por la fundación. Una organización puede autocertificar el producto o que este sea certificado por un tercero, cualquier sea el caso quien certifica declara tanto a la fundación OpenID como al público general que lo tratado en el certificado es preciso.

4.1.2.3. Comparación con otros protocolos

Como se mencionó antes, existen otros protocolos de autenticación, pero no todos son adecuados para el proyecto. Tanto Kerberos como RADIUS no están pensados para utilizarse en aplicaciones web, Kerberos está orientado a la conexión entre dispositivos bajo una misma red, mientras que RADIUS fue pensado para el acceso a conexiones de redes wifi, por esta razón estos protocolos son descartados.

Por otro lado, SAML sí se utiliza en aplicaciones web, sin embargo, solamente puede soportar aplicaciones web en su protocolo [6], mientras que OIDC puede utilizarse también para aplicaciones nativas y móviles, lo que se traduce en una ventaja si se quieren incorporar estas en un futuro. Además, OIDC es un protocolo más moderno y fácil de implementar que SAML [19], lo que facilita la mantención a futuro de la solución, aspecto que es importante a tener en cuenta. Por último, la arquitectura de OIDC, está centrada para API, ayudando a que sea más fácil conectar las instancias de las aplicaciones.

4.1.2.4. Desarrollar versus adquirir

Por último, es importante analizar la opción de utilizar las plataformas de identidad disponibles en el mercado como las de Google o Microsoft, si bien estos sistemas cuentan con seguridad confiable e infraestructura adecuada, puede que no sea la solución adecuada para esto. Para empezar una preocupación importante es la personalización, puede que los requisitos iniciales pueda ser satisfechos por alguna de estas soluciones, sin embargo, estos requisitos pueden ir cambiando, BPO trabaja con diversos clientes y sus requerimientos pueden cambiar y diversificarse, por lo que es necesario que el sistema pueda adaptarse y actualizarse, por lo que crear una solución propia es beneficioso, sumado a esto es el menor coste que se tiene al tener un sistema propio.

4.2. Desarrollo de la solución

Ya con la decisión tomada sobre lo que se busca en la solución se comenzó por el diseño de las partes que la componen y en las semanas posteriores se desarrolló la solución en sí, como se verá en esta sección.

4.2.1. Diseño flujos y casos de uso

En las primeras etapas del desarrollo se diseñó el flujo a seguir en la aplicación, además de casos de uso.

En primera instancia se tiene el flujo que seguirá el protocolo OIDC para la autenticación. Servidor de autenticación y *Open ID Provider* se refieren a la solución que se desarrolla, mientras que Cliente, *Relying Party* e instancia de aplicación se refieren a la aplicación que hará uso de la solución como se mencionó anteriormente.

A continuación se puede ver en la figura 4.1 y 4.2 la forma en que las plataformas interactúan para lograr la autenticación del usuario. Esto es el flujo de autenticación que sigue el protocolo de OIDC.

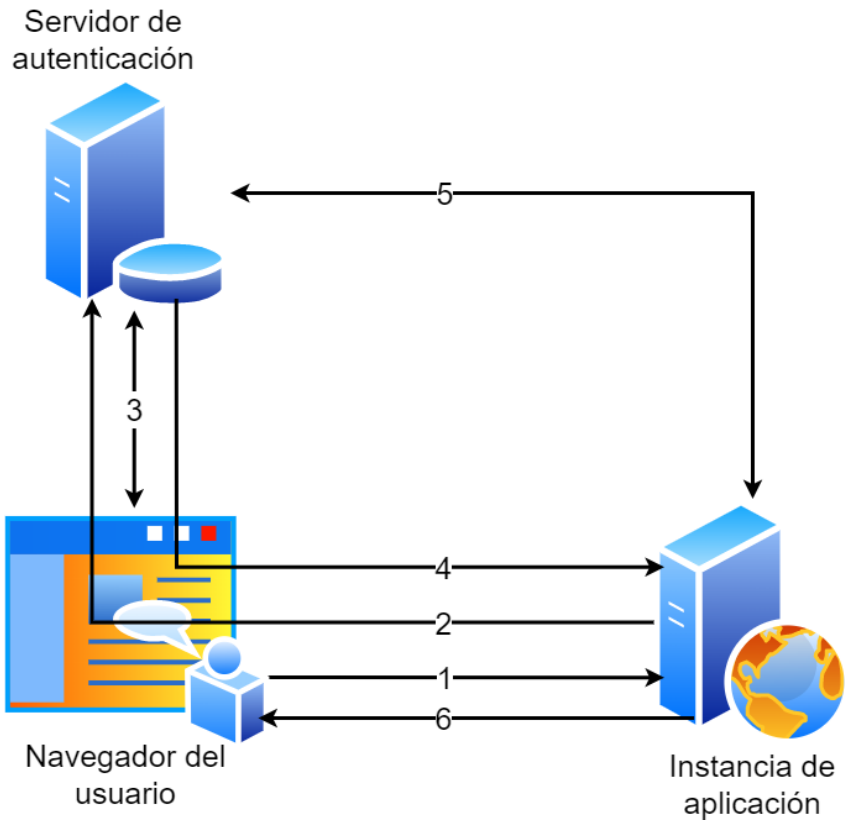

Figura 4.1: Esquema del funcionamiento esperado de la solución.

1. El usuario accede a la instancia de la aplicación cliente, por ejemplo Firma Ya.

2. En caso de no estar autenticado, la aplicación redireccionará al usuario hacia la solución de este proyecto, el servidor de autenticación.

3. Una vez allí se le pide que se autentique con alguno de los métodos disponibles. Dependiendo de la instancia y el usuario, puede haber algunos métodos que no se puedan utilizar o puede que se necesite más de un método.

4. Luego de una autorización exitosa, el servidor de autenticación redirigirá al usuario de vuelta a la aplicación inicial.

5. La aplicación utiliza los tókenes para emitir una solicitud al proveedor de identidad para terminar el proceso de autenticación.

6. La aplicación indica a los visitantes que iniciaron sesión.

A continuación se presenta el flujo propuesto en mayor detalle como se ve en la figura 4.2.

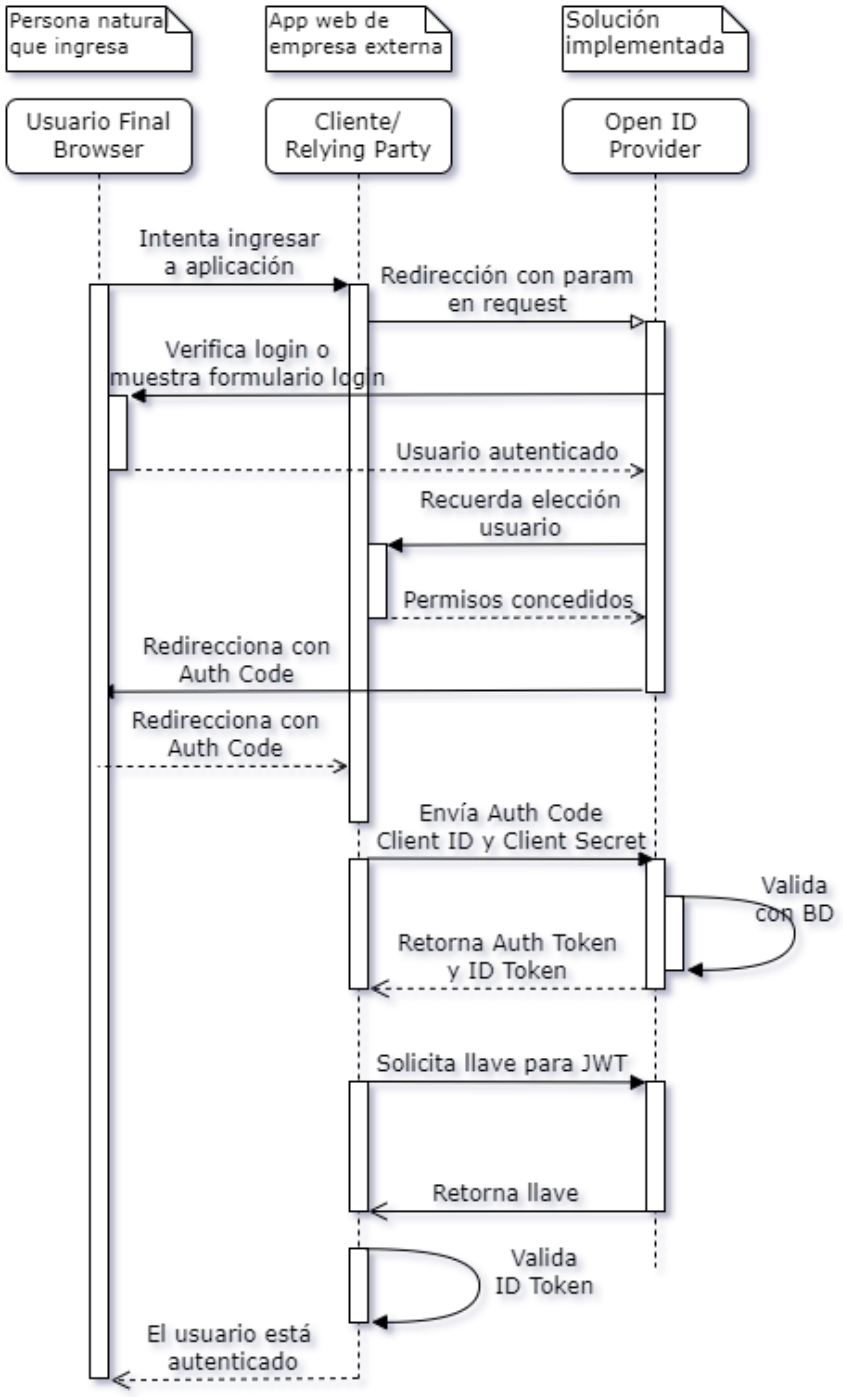

Figura 4.2: Diagrama del funcionamiento esperado de la solución.

La secuencia anterior utiliza el flujo de código de autorización. Para empezar, el usuario intenta entrar a la aplicación y el navegador se redirecciona a la aplicación proveedora. En este caso se asumirá que el usuario ya se ha autenticado antes y la aplicación recuerda sus datos, debido a esto verifica el login y el usuario está autenticado. Luego de esto, el proveedor recuerda la elección de *scopes* y entrega los permisos concedidos. Como paso siguiente se redirecciona al navegador con el código de autorización, el cual es enviado al proveedor de OIDC y este entrega de vuelta los tókenes

de autorización e identidad. Ya con los tókenes en poder de la aplicación cliente, se solicitan las llaves para los JWT y con esta se valida el *Token ID* autenticando así al usuario y terminando el flujo.

Respecto a la configuración necesaria para que se pueda lograr el flujo del protocolo anterior, en la figura 4.3 se muestra lo necesario para los clientes, mientras que en la figura 4.4 se tiene la necesaria para los usuarios.

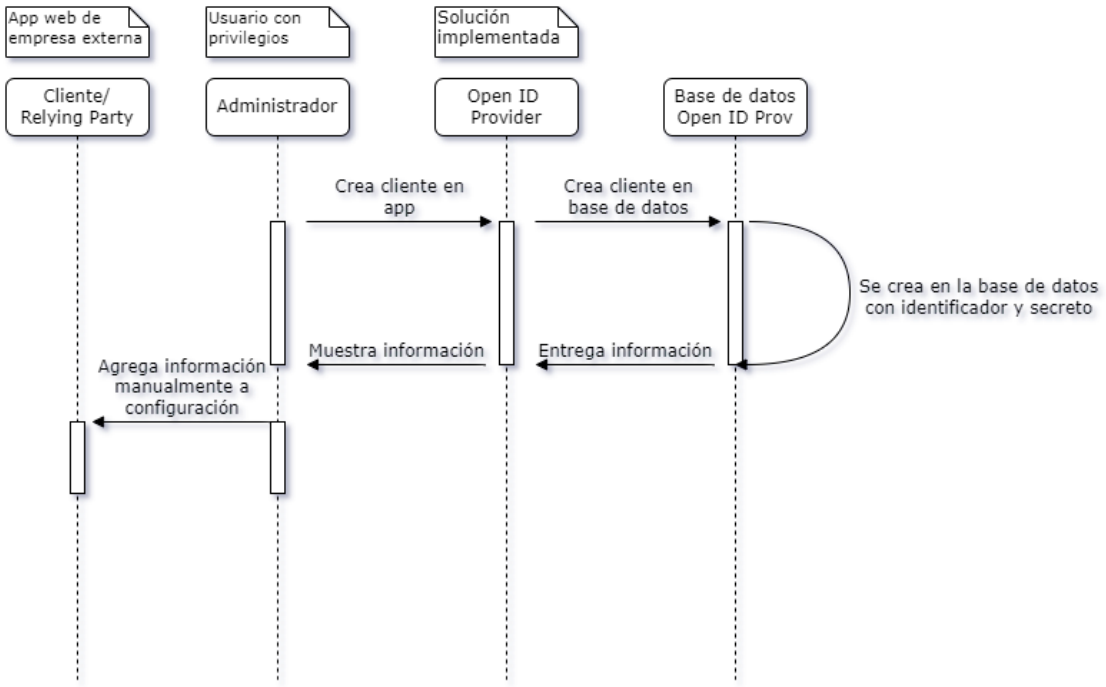

Figura 4.3: Diagrama de la configuración de los clientes.

Para la creación de clientes, un usuario con rol de administrador debe crearlos en la aplicación proveedora (la solución), estos serán generados en la base de datos con un identificador y un secreto el cual podrá verse en consola. El administrador deberá utilizar estos datos para agregar la información manualmente a la configuración de la aplicación cliente y así se pueda reconocer en el proveedor de OIDC.

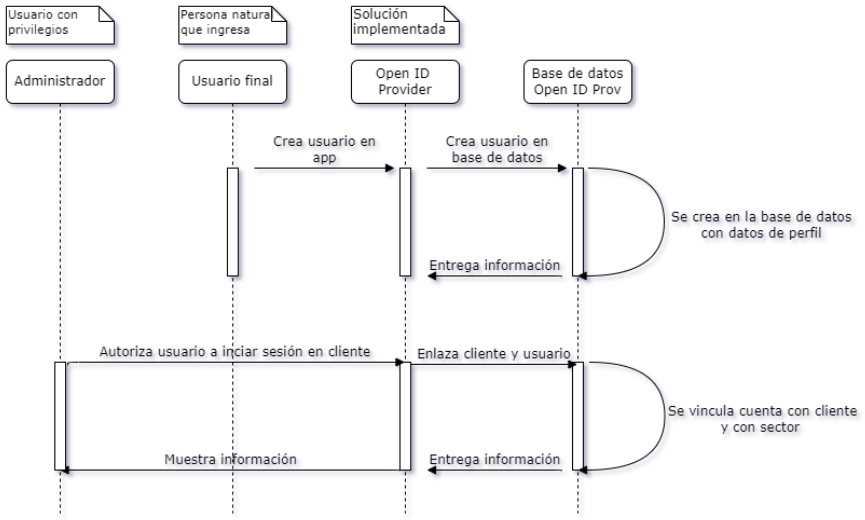

Figura 4.4: Diagrama de la configuración de los usuarios.

Para la creación de cuentas de usuarios en la app, los usuarios deben crearlas por sí mismos utilizando su correo y ciertos datos obligatorios, siendo estos rut, nombre y apellidos. Una vez que el usuario esté creado, un administrador debe enlazarlo a un cliente previamente creado para que este esté autorizador a poder autenticarse con ese cliente.

Desde el punto de vista del flujo que seguirán los usuarios se hizo el siguiente diagrama como objetivo a alcanzar como se ve en la figura 4.5. Se contempla desde que el usuario busca autenticarse en la plataforma cliente hasta que lo logra o se termina el flujo sin el usuario autenticado.

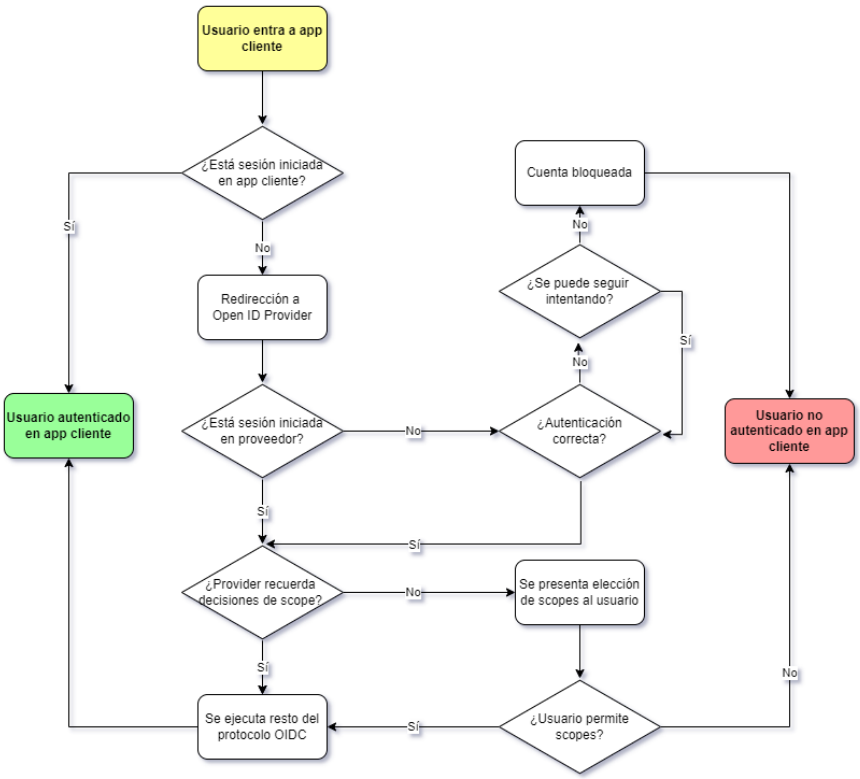

Figura 4.5: Diagrama de flujo del funcionamiento esperado de la solución para los usuarios.

El flujo empieza con el usuario queriendo autenticarse en la aplicación cliente, si ya está autenticado el flujo termina exitosamente, en caso de no estarlo se redirecciona al proveedor de OIDC como se ha visto en los otros gráficos de esta sección. Luego de esto se revisa si está autenticado en el proveedor, en caso de no estarlo debe autenticarse y en caso de fallar la autenticación más de tres veces la cuenta se bloquea y el flujo termina con el usuario sin estar autenticado en la aplicación. Si el ingreso de credenciales es correcto o si el usuario está autenticado en el proveedor de identidad desde antes se chequea, si la aplicación tiene recuerdo de las decisiones de *scope* que tomó anteriormente, en caso de no tener las decisiones en el sistema se muestran los *scopes* para que el usuario seleccione sus preferencias. Si el usuario rechaza los *scopes* el flujo termina con el usuario sin autenticarse. Si el usuario acepta los scopes estos se guarda en la aplicación para que los recuerde, se continúa el flujo de OIDC y termina con el usuario autenticado en la aplicación.

Relacionado a este último flujo se encuentra la secuencia para autenticarse, la cual se ve en la figura 4.6:

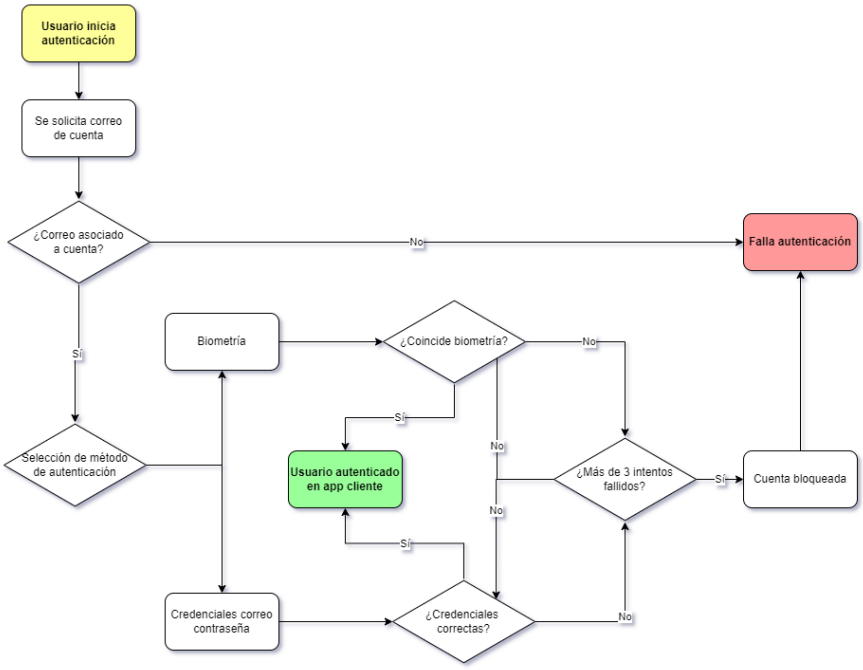

Figura 4.6: Diagrama de flujo del funcionamiento de la autenticación para los usuarios.

El proceso comienza desde la autenticación que puede ser al ingresar a la aplicación proveedora o al intentar hacer el flujo de OIDC. En esta, el usuario debe ingresar su correo electrónico y en caso de estar asociado a una cuenta se puede elegir el tipo de autenticación. En caso de ser con contraseña se debe ingresar esta y en caso de ser por biometría se debe utilizar con cámara o lector de huella digital, en ambos casos se dan tres intentos en caso de fallo y se bloquea al cuarto. Si la autenticación funciona se da acceso a la cuenta.

4.2.2. Diseño modelo base de datos

El diseño del modelo de la base de datos se hizo tomando en cuenta el funcionamiento de la gema de *openid_connect*, más detalles de esto se pueden encontrar en la sección 4.2.3.1.

Se comenzará a explicar el modelo empezando por las entidades Account y Data. Account representa la cuenta de un usuario que utiliza la plataforma, tiene un tipo el cual indica si es un usuario principal o administrador, también cuenta con información como el correo electrónico de la cuenta y el último *login* registrado. Data representa los datos extra que tiene un usuario, estos son los datos que son comunes a todos los sistemas de la empresa y son: nombre, rut, teléfono, apellido paterno, apellido materno, dirección, comuna, ciudad y país. Las cuentas solo pueden tener asociada a lo más una data y la data solamente puede pertenecer a una cuenta, por último las cuentas de administrador no tienen data asociada.

Otro pilar fundamental del protocolo son los clientes que representan las aplicaciones cliente a utilizar el proveedor de OIDC, estos tienen un identificador y secreto que les permite validar su

identidad como se detalló en la sección 3.4.2, además de URIS para ciertos *endpoints*. Un cliente puede estar relacionado con muchas cuentas y una cuenta relacionada con muchos clientes, esto ya que se debe seleccionar cuáles usuarios se pueden autenticar en ciertos clientes.

La siguiente entidad son los sectores, los cuales tienen un nombre para identificarlos y corresponden a una agrupación de clientes. Estos están relacionados con las cuentas para conferirles un PPID como se detalló en la sección 3.4, el cual es guardado en la relación. Un cliente siempre pertenece a un sector y solo a un sector (si no comparte PPID tiene un sector consigo mismo), mientras que un sector tiene uno o más clientes. Un sector puede tener varias cuentas a las que les crea un PPID y una cuenta puede tener varios sectores que le dan su PPID.

Luego se tiene *Authorization Code* que funciona como código de autorización y token de autorización, por lo que tiene un parámetro que indica si es un token, esto debido a que comparten muchos parámetros y relaciones con otras entidades. Además de esto, guardan la URI de redirección al cliente, la expiración que tienen y *nonce* para mayor seguridad. Un Athorization Code solamente puede pertenecer a un par cuenta cliente, mientras las cuentas y clientes pueden tener varios Athorization Code.

De la mano de lo anterior están los Scope como se menciona en la sección 3.4, estos cuentan con su nombre que corresponde al permiso que representan. Están relacionados con Authorization code denotando los permisos que están confiriendo. Un scope puede delimitar el alcance de varios Authorization codes y un Authorization code puede contar con el alcance de varios scopes.

Por último se tiene a los ID Token, los cuales indican que una cuenta se autenticó y que será enviado a un cliente, lo cual queda demostrado por la relación, tiene información de su expiración y cuenta con un *nonce* para mayor seguridad. Un ID Token solamente puede pertenecer a un par cuenta cliente, mientras las cuentas y clientes pueden tener varios ID token.

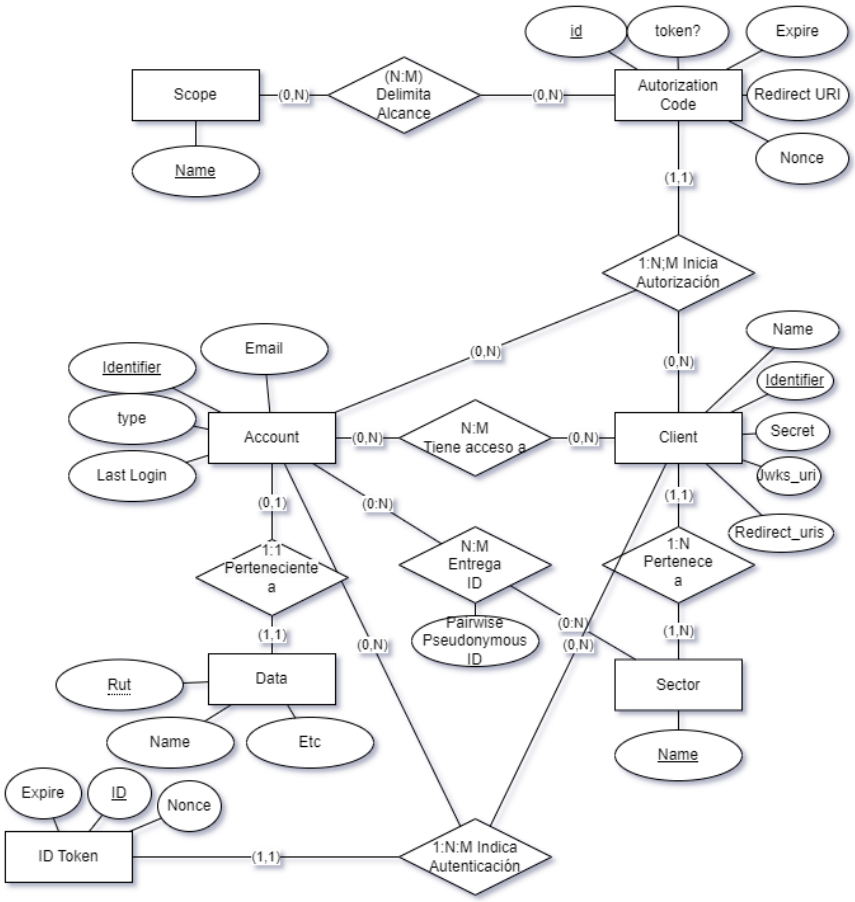

Figura 4.7: Modelo de base de datos.

4.2.3. Desarrollo proyecto base

El proyecto inició su desarrollo como una aplicación web utilizando el lenguaje de programación *Ruby* y el *framework Ruby on Rails*, se utilizaron las versiones más recientes al momento de comenzar el proyecto, *Ruby* 3.1.2 y *Ruby on Rails* 7.0.4. Agregado a esto se utilizó la librería *Bootstrap* de HTML, CSS y JavaScript.

Para la aplicación base, el objetivo en mente era el de tener un sistema de ingreso y administración de usuarios robusto sobre el que se pueda llevar a cabo el protocolo OIDC. Para esto se utilizó el *framework* de autenticación *Rodauth* [20], ya que tiene características beneficiosas para el proyecto. Para empezar provee soporte de autenticación multifactor y métodos de autenticación sin necesidad de contraseña (*passwordless authentication*), lo que es bastante útil para la solución a desarrollar, ya que permite utilizar los sistemas de autenticación como segundo factor. Por otro lado, se utiliza un modelo de guardado de hash de contraseñas que evita que se puedan filtrar los hash sin un ataque escalado de privilegios previo en la base de datos, esto es debido a que hace uso de funciones de la base de datos para acceder a los hash de las contraseñas, por lo que la aplicación no es directamente quien las obtiene, sino la base de datos. En caso de que un atacante pueda acceder a la base de datos y obtener las contraseñas, se encontrará con los hash de contraseñas creados

con Bcrypt [21], la ventaja de esta función de hash es que a diferencia de otras funciones como SHA-256, esta fue diseñada con el objetivo en mente de que las operaciones sean lentas, por lo que se dificulta el trabajo de hacer ataques de fuerza bruta. Además, se utiliza "sal"(agregar un valor al azar antes del hash), por lo que ataques de tablas de hash o tablas arcoíris son menos fructuosos.

Se utilizaron los generadores del *framework* para crear la aplicación base, y la gema de autenticación generó modelos, vistas y controladores para la autenticación básica de usuarios (*Ruby on Rails* utiliza la metodología MVC), ya que para los usuarios se necesita información extra se creó un modelo de perfil para las cuentas de usuario, que albergue toda la información adicional que se necesite y se modificaron las vistas y controladores de forma acorde.

Además, haciendo uso de la gema *Rodauth* se pudo agregar autenticación multifactor con TOTP y códigos de recuperación, junto con agregar otros métodos para autenticarse, siendo estos webauthn que puede hacer uso de la cámara de un dispositivo móvil para la validación de identidad y autenticación mediante correo (sin contraseña), lo último que se hizo para la aplicación principal fue diferenciar a los usuarios administradores.

4.2.3.1. Desarrollo protocolo OIDC

Para la implementación del protocolo OIDC se utilizó en primera instancia la gema *openid_connect* en su versión 1.3.0, la más reciente en ese momento. Si bien esta librería tiene la certificación de la fundación Open ID, no cuenta con suficiente documentación para utilizarse de forma directa, por lo que se utilizó uno de los ejemplos que provee. Los ejemplos de *provider* y *relying party* sirvieron como guía para el flujo del proyecto y la base de datos.

Con esto en mente se diseñó la base de datos, teniendo en cuenta lo necesario para el flujo y rediseñando la base de ejemplo para así evitar redundancias, mejorar la lógica y actualizar el diseño, obteniéndose el resultado que se observó anteriormente en la figura 4.7.

Al momento tener la base de datos hecha en la aplicación y comenzar a programar el flujo para el protocolo se presentó un error silencioso en la aplicación. Luego de recibir el llamado en el *endpoint* de autorización del proveedor desde la aplicación cliente, se utilizan los datos de entorno de la petición para la autorización.

Para esto se utiliza el método *call* de *Ruby* en la estructura del *endpoint* de autorización el cual cuenta con información del cliente, el método *call* ejecuta un método que se le dé como parámetro sobre el objeto al cual se le llama y el parámetro en el código utilizado era el entorno de la petición desde la aplicación cliente. Esta variable cuenta con mucha información distinta, lo que dificulta el proceso de descubrir el método que invoca la llamada, ya que tiene la información de todo el entorno de la petición. Se inició un proceso de debugging e investigación para poder solucionar

el problema, pero sin grandes avances. Se cree que la diferencia entre versiones (el proyecto de ejemplo utiliza *Ruby on Rails* 3.2 llevó a un comportamiento distinto al esperado, mientras que la falta de documentación dificultó la tarea de diagnosticar el error.

4.2.3.2. Cambio de librería para el protocolo

Si bien la librería funcionaba de buena manera, debido a las dificultades de utilizar la gema anterior y debido a que se estuvo trabajando durante un período sustancial de tiempo sin obtener resultados satisfactorios, se optó por probar con otra librería. La gema *Rodauth* utilizada para la autenticación cuenta en su página con características externas desarrolladas por otros usuarios distintos a su creador, siendo una de estas *rodauth-oauth* que implementa el protocolo *Oauth 2* incluido el protocolo *OIDC*. Se optó por esta gema debido a que es una extensión de una que ya se está utilizando, por lo que se esperó que se complementen sin mayores complicaciones, además de que la documentación de *Rodauth* se extrapola a esta gema para su uso.

La nueva gema utilizada hace gran parte del trabajo, incluyendo el modelo de datos para llevar a cabo el protocolo, por esta razón lo que se necesitó hacer fue configurarla de la manera adecuada para satisfacer lo que necesita el proyecto. En esta ocasión la librería presenta más documentación y guías de su uso, se siguió una guía de configuración para *Ruby on Rails* haciendo los cambios necesarios para utilizarse en la nueva versión. Se añadieron las vistas y se adaptaron a la aplicación web, por último se utilizó el cliente de prueba que presenta la librería para comprobar el funcionamiento del flujo por completo.

El modelo de datos utilizado por la gema Rodauth Oauth se puede apreciar en la figura 4.8 en este el modelo de Account y Data funcionan de la misma manera a excepción de que aquí es llamado Profile.

La librería tiene el modelo Oauth APP que hace de cliente y relación con las cuentas a la vez. Cuenta con toda la información necesaria, nombre de cliente, identificador, secreto, además de la URI de redirección y página de inicio. Una diferencia es que aquí se tiene un atributo que indica los *scopes* recordados para este cliente. Un Oauth APP tiene siempre una cuenta y una cuenta puede tener muchos OAuth APP.

Luego se tiene OAuth Grants que corresponden a los códigos de autorización, cuentan con las *scopes* seleccionadas, la URI a la cual deben redireccionar, la fecha de expiración y atributos para la seguridad como un código de seguridad y *nonce*. Están relacionados con un OAuth App y esta puede tener varios OAuth Grants.

Por último se presentan los OAuth Token, los cuales se utilizan para crear los tókenes, por lo que cuentan con los *scopes* elegidos, expiración del token y *nonce*. Está relacionado con OAuth Grants, ya que son los tókenes creados al obtener el código de autorización. Puede estar relacionado con

un OAuth Grant y un OAuth Grant puede tener a lo más un OAuth Token.

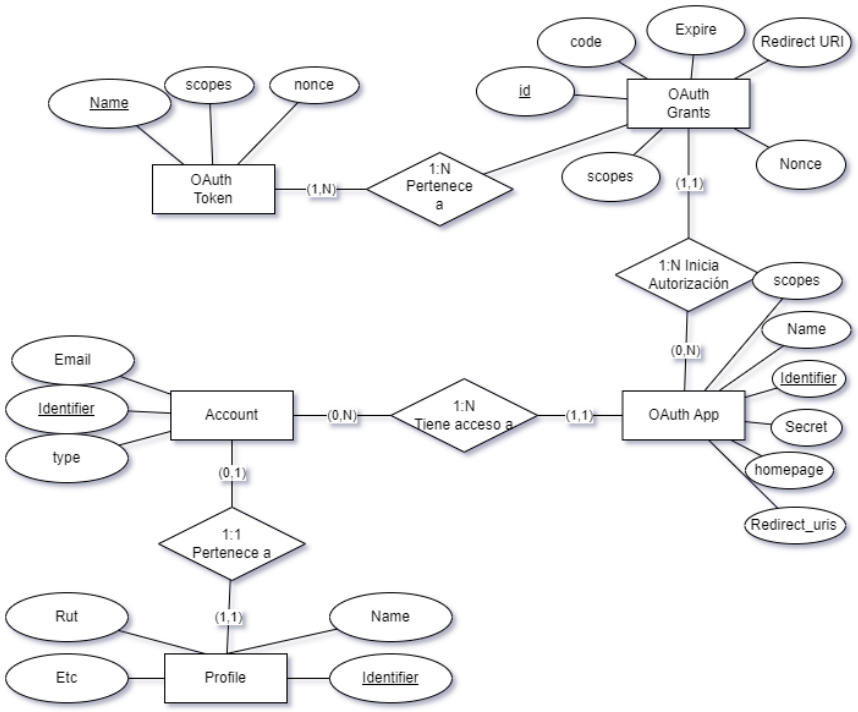

Figura 4.8: Modelo de datos simplificado proporcionado por gema Rodauth Oauth.

Al intentar con el cliente de ejemplo se obtuvieron unos cuantos errores y se tuvo que hacer correcciones a los archivos generados para poder solucionarlos. En primera instancia el flujo no se podía completar, ya que el cliente tomaba la *request* del proveedor como un ataque CSRF luego de que se seleccionasen los *scopes* para los permisos. Esto se debía a que las vistas generadas por la gema en conjunto con la aplicación hacían uso de Turbo [22], una librería de *Javascript* que permite agilizar la carga de páginas al solamente reemplazar el cuerpo de las distintas vistas de la aplicación, debido a esto los *headers* necesarios para evitar que se provoque una alerta de ataque CSRF no se estaban enviando, por lo que se tuvo que desactivar Turbo en la petición para poder comunicarse efectivamente con el cliente.

Luego de eso, el protocolo entregaba un error sin cuerpo de mensaje, por lo que para diagnosticarlo hubo que descargar el código fuente de la librería y comenzar el proceso de *debugging* analizando el flujo que se seguía. Se llegó a la conclusión que uno de los problemas era que el cliente de pruebas provisto por la gema carecía de un *endpoint JWKS*, lo que evitaba que se pudieran obtener las llaves necesarias.

A pesar de las dificultades obtenidas por la migración a la nueva librería, se logró completar el flujo, lo que permitió la autenticación de los usuarios en la plataforma cliente.

4.3. Desarrollo post validación

Debido a los problemas mencionados anteriormente que consumieron más del tiempo que se tenía presupuestado inicialmente (de 4 semanas para implementar el protocolo OIDC a 8 semanas) no se pudo llevar a cabo la validación con usuarios reales que se esperaba, pero sí se logró tener una instancia de retroalimentación como se detalla en el capítulo 5. Luego de este *feedback* se lograron hacer ciertos cambios menores, como validación de rut en *frontend* para que agregue automáticamente el guión e indique con colores si es válido o no, más claridad en la configuración de autenticación con *Web Authn* y por último, consistencia en el idioma del sistema, además de otros cambios menores como posición de botones y agregado de etiquetas.

4.4. Resultado

Con el desarrollo de la solución terminado, la aplicación resultante logra cumplir con el protocolo de Open ID Connect, a continuación se verán los casos de uso mediante las vistas de la aplicación.

4.4.1. Flujo Open ID Connect

Para el protocolo principal por un usuario que se conecta por primera vez con su cuenta se tiene lo siguiente.

Para empezar el usuario ingresa a la aplicación cliente, donde esta requiere autenticarse, como no está autenticado no se muestra nada en ésta.

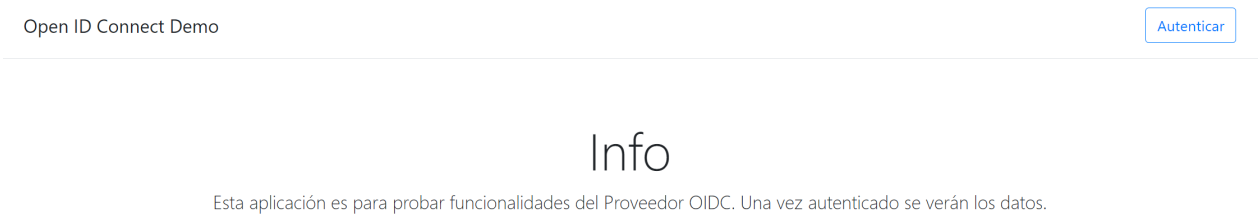

Figura 4.9: Usuario entra a app cliente.

Cuando el usuario selecciona el botón de autenticar, es redireccionado a la solución, la cual requiere que esté con su sesión iniciada, por lo que muestra un mensaje de error y solicita que se inicie la sesión ingresando el correo electrónico.

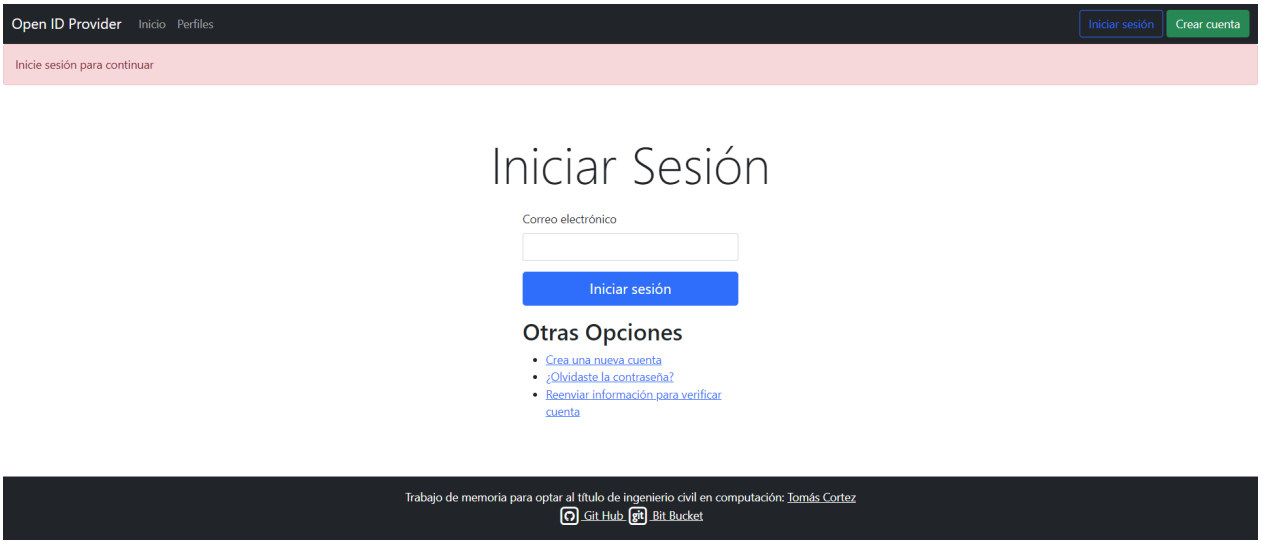

Figura 4.10: Sesión no iniciada en proveedor.

Una vez que el usuario indica su correo electrónico, la plataforma solicita que se autentique con alguno de los métodos disponibles. Como se ve en la figura 4.11 se puede ingresar mediante contraseña, mediante Web Authn si se ingresa el nombre que se asignó a la llave como se mostrará más adelante en la sección 4.4.3 o mediante un enlace enviado al correo electrónico del usuario.

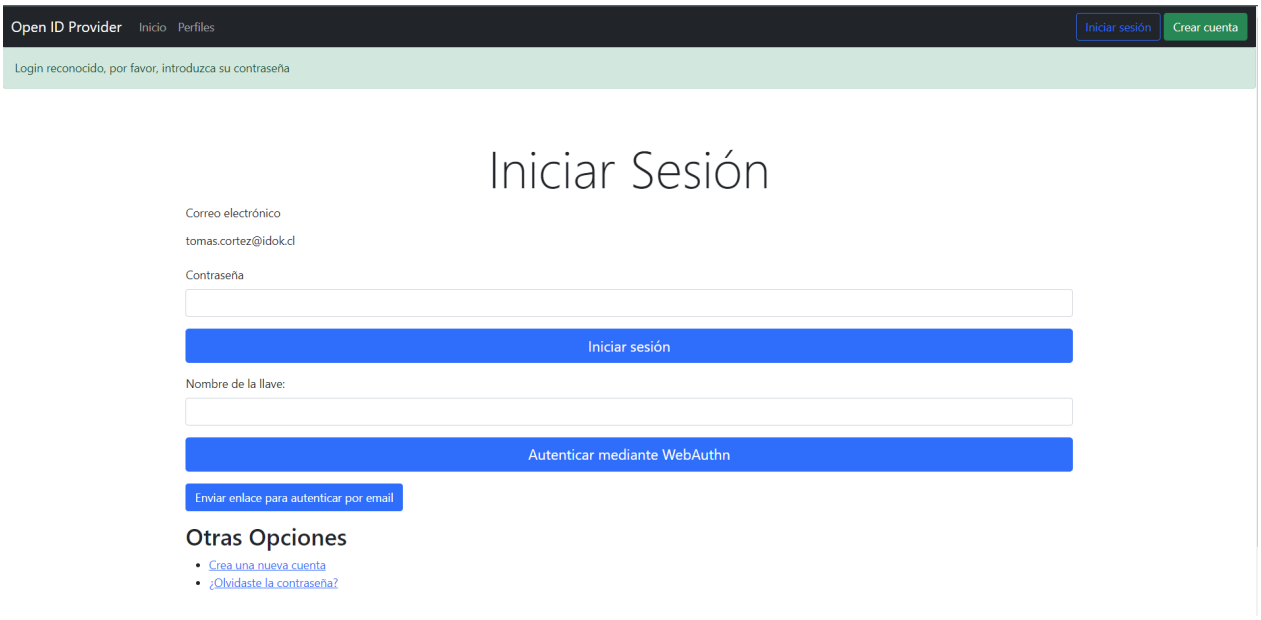

Figura 4.11: Ingreso de credenciales en proveedor.

Una vez que el usuario se autenticó de forma exitosa a este se le presentan los *scope* de la aplicación, como se puede ver los primeros tres son obligatorios, mientras que los últimos tres son seleccionables para poder elegir. En este ejemplo se seleccionan todos menos el número de teléfono.

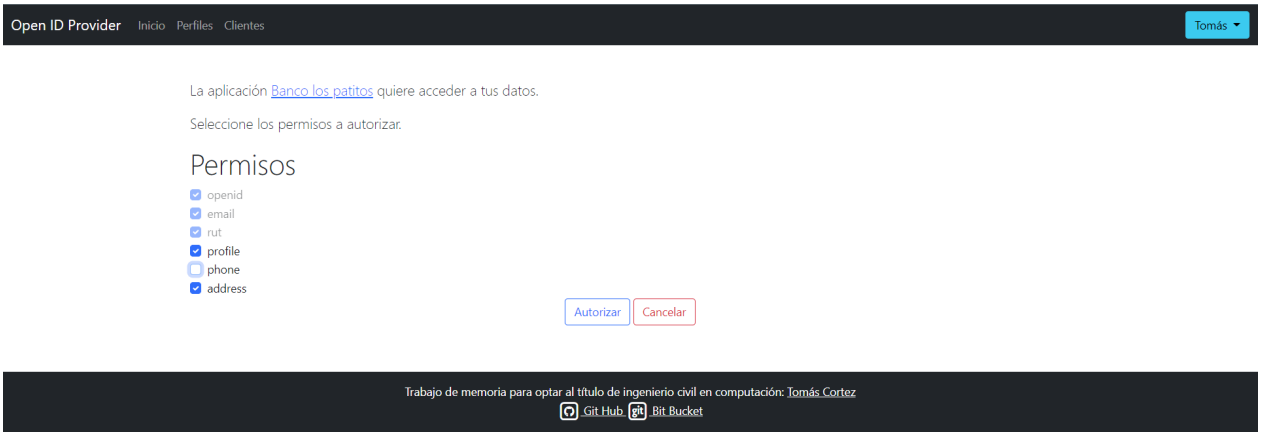

Figura 4.12: Se presenta elección de scopes al usuario.

Luego de que el usuario selecciona aceptar, es redireccionado a la aplicación cliente, en la cual puede observar los datos que permitió compartir con la aplicación, terminando así el flujo.

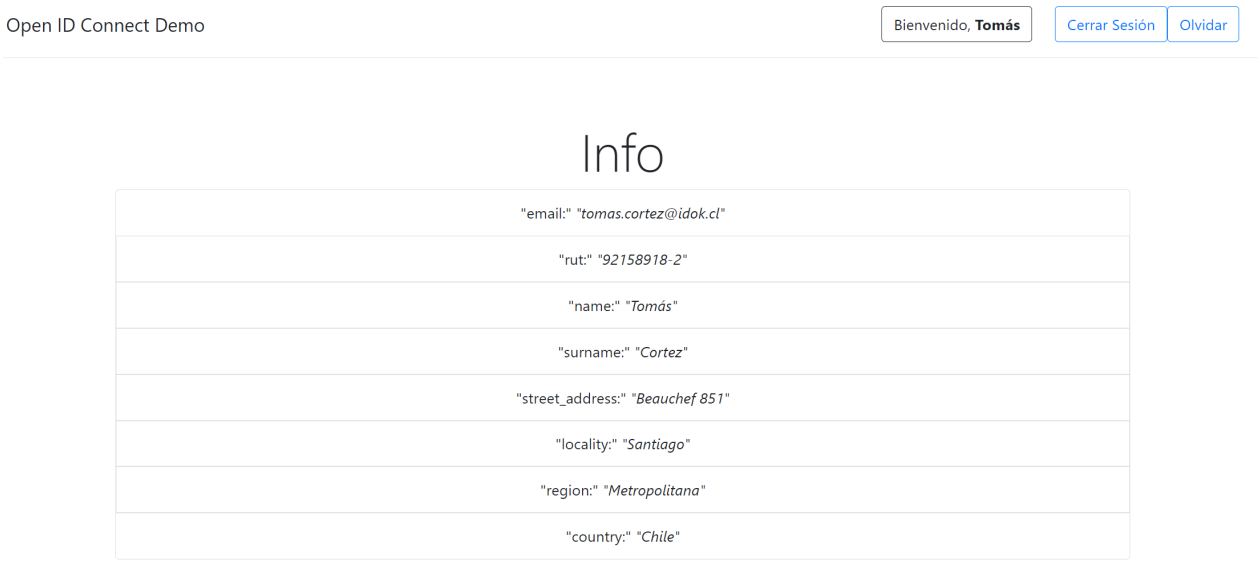

Figura 4.13: Se termina el proceso con usuario autenticado en app cliente.

4.4.2. Flujo Open ID Connect con login en memoria

En caso de que un usuario quiera ingresar a la app luego de haber cerrado su sesión (pero no habiendo olvidado esta) y se encontrará con la página principal de la aplicación cliente y el botón de login.

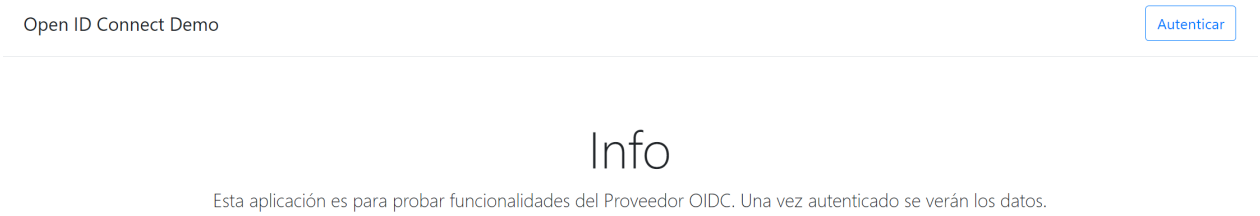

Figura 4.14: Usuario entra a app cliente nuevamente.

Al hacer *clic* sobre el botón de autenticar, este será conectado automáticamente, ya que el flujo de OIDC recuerda sus decisiones y se lleva a cabo.

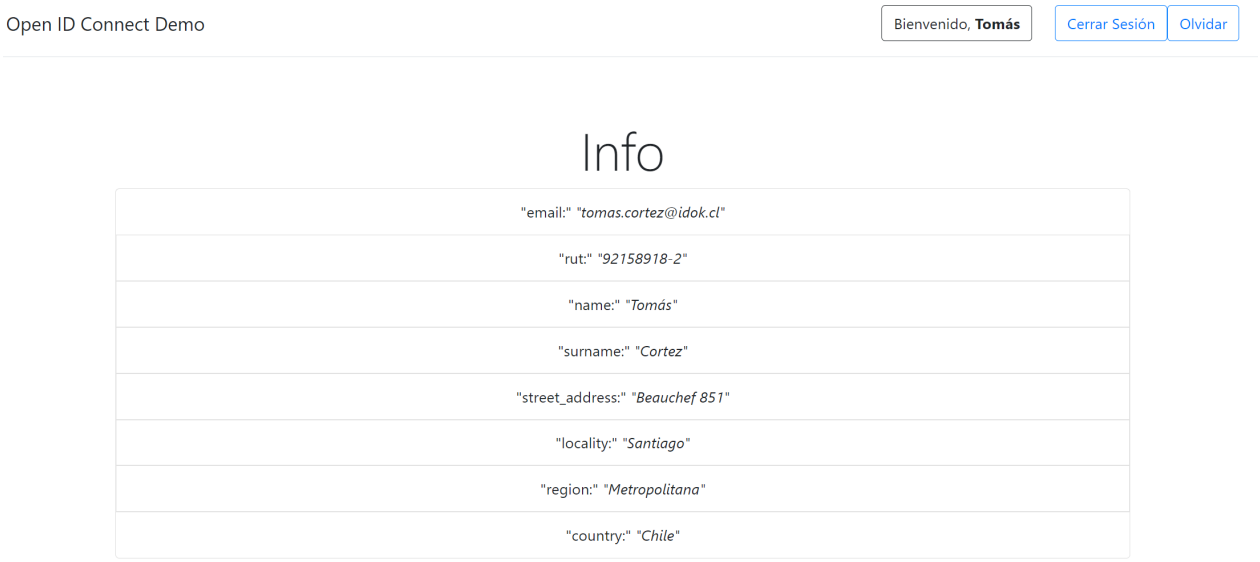

Figura 4.15: Se termina el proceso con usuario autenticado en app cliente.

También da la posibilidad de que en el flujo anterior la sesión en la aplicación proveedora no haya estado iniciada, en ese caso se debe iniciar la sesión y el flujo seguirá autenticándolo.

4.4.3. Configuración de WebAuthn

A continuación se mostrará como se configura y utiliza WebAuthn para la autenticación. Para empezar se debe ingresar a la sección de Administrar Auth Multifactor para poder comenzar la configuración.

Figura 4.16: Página principal.

Luego se debe reingresar la contraseña para poder acceder a la configuración.

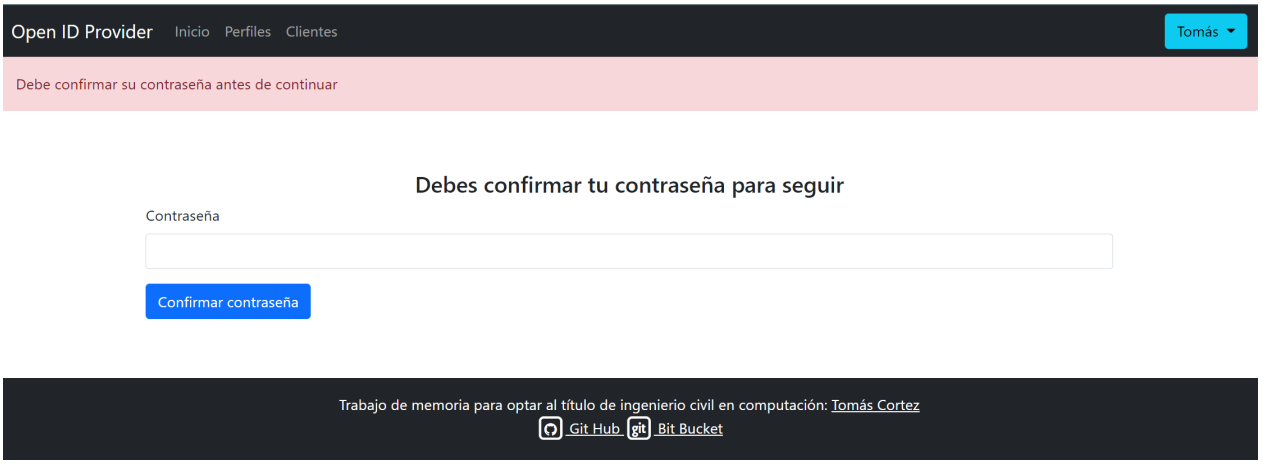

Figura 4.17: Petición de contraseña.

Una vez que las credenciales se aceptan, se muestra la página de configuración de WebAuthn con una breve descripción del estándar e indicaciones para utilizarlo.

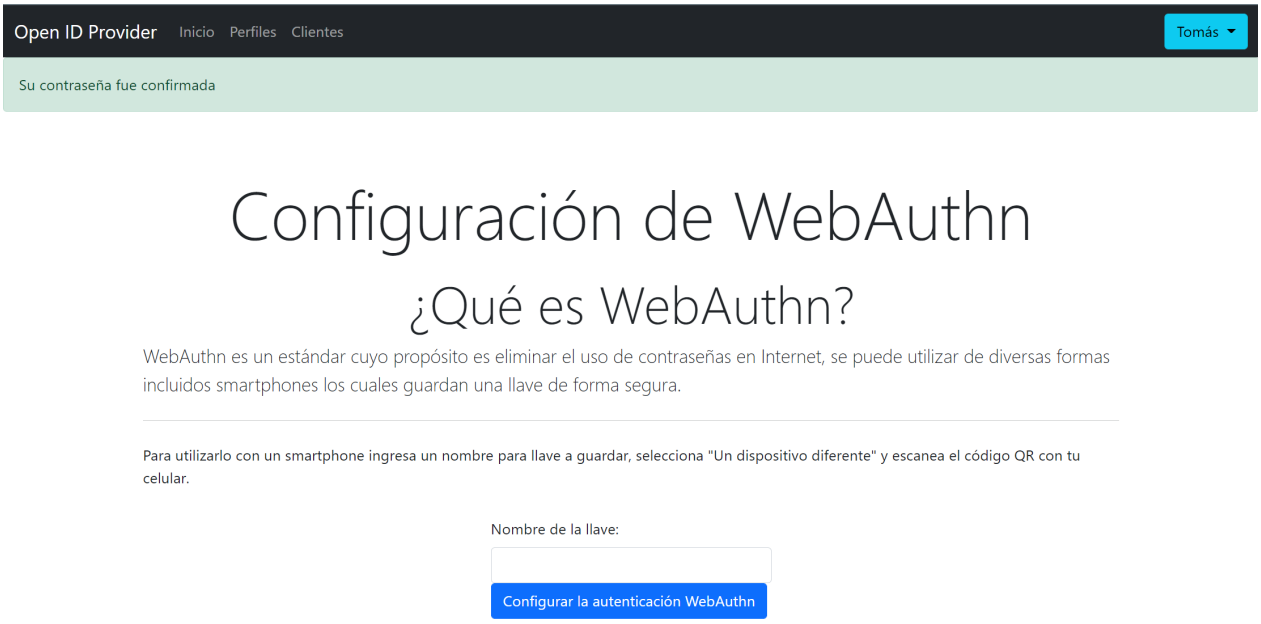

Figura 4.18: Configuración de WebAuth.

Siguiendo las indicaciones se entrega un nombre para la llave y se selecciona la opción de configurar autenticación WebAuth. Esto muestra una petición que debe aceptarse.

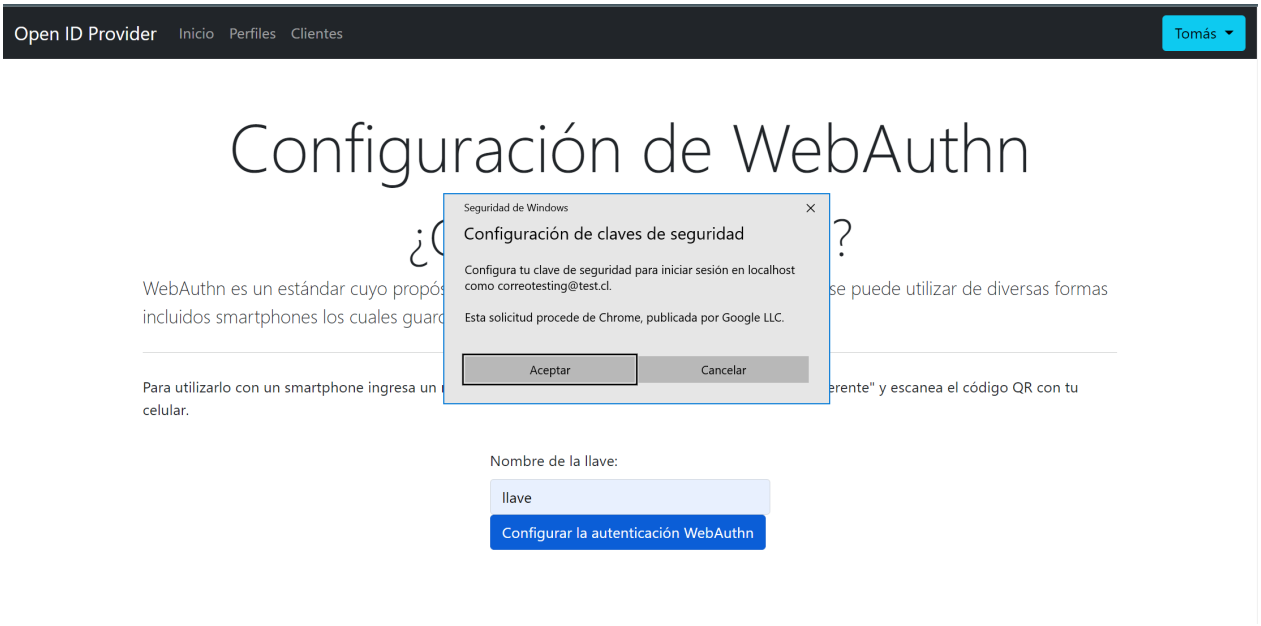

Figura 4.19: Petición de configuración.

Una vez se acepta se debe seleccionar la forma en que se quiere crear una llave, al indicar un dispositivo diferente se puede escanear un código QR que permite hacerlo desde un dispositivo inteligente.

Figura 4.20: Selección de creación de llave.

Una vez que se escanea el código QR debe aceptarse la petición en el dispositivo, este se puede aceptar utilizando biometría facial para la autenticación.

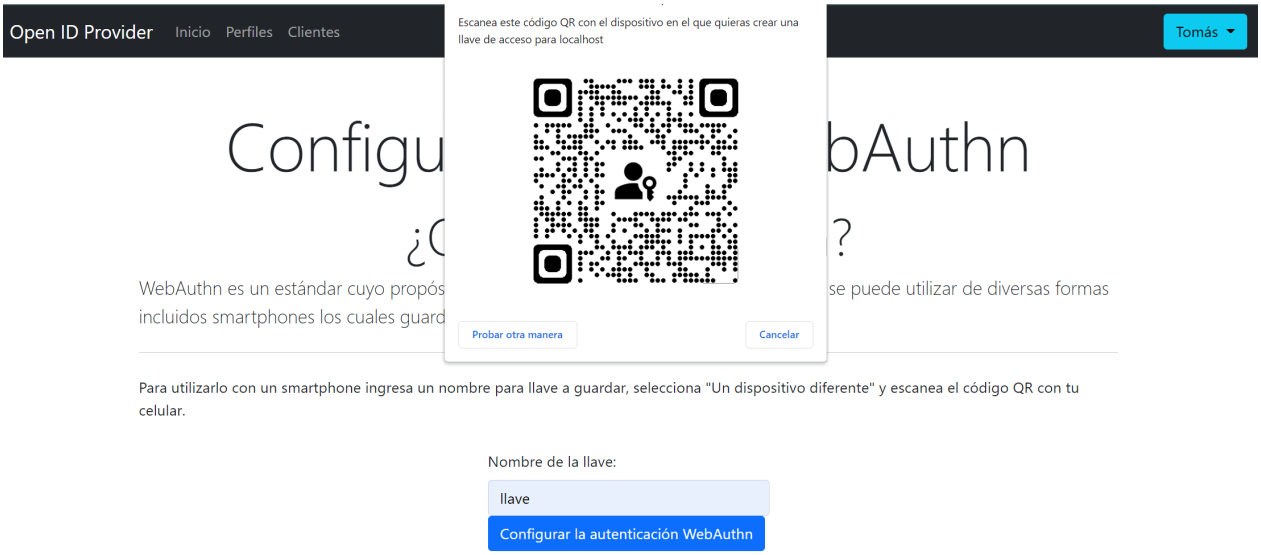

Figura 4.21: Código QR.

Con esto está lista la configuración y se puede utilizar WebAuthn para autenticarse en el futuro.

4.4.4. Uso de WebAuthn

La forma de utilizar WebAuthn para autenticarse es similar a la forma en la que se hizo la configuración. Partiendo desde el inicio de sesión con el correo electrónico ya ingresado se debe ingresar adicionalmente el nombre de la llave creada.

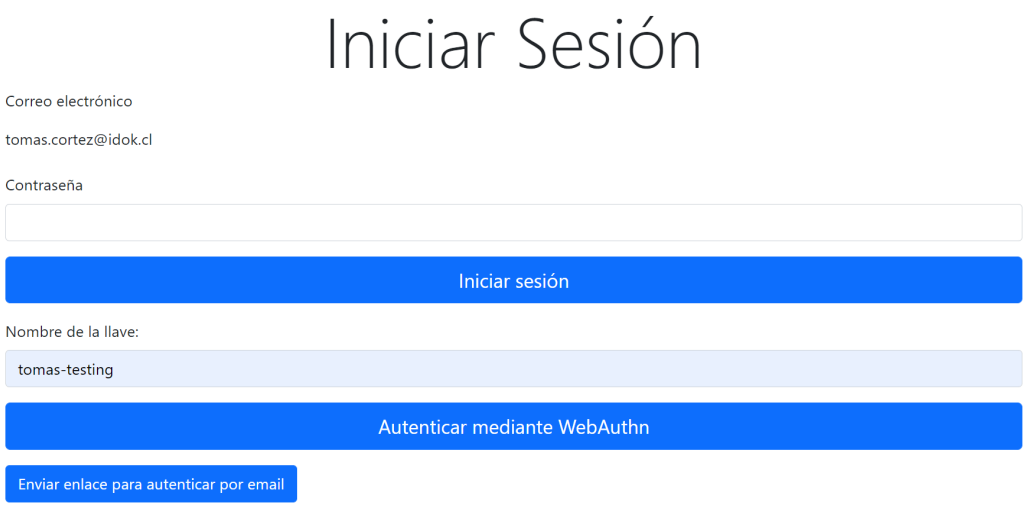

Figura 4.22: Petición de contraseña.

Luego de esto, al igual que en la configuración, se debe seleccionar un dispositivo diferente como el dispositivo que tiene la llave de acceso para la plataforma.

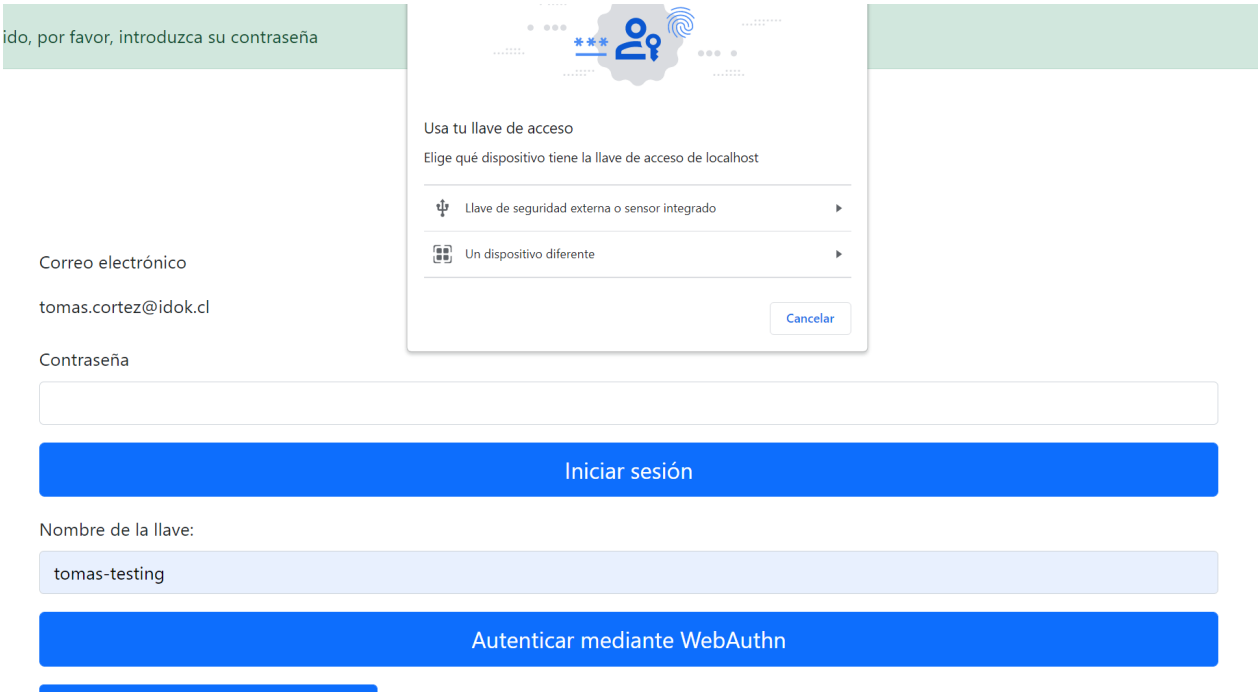

Figura 4.23: Petición de contraseña.

Por último se escanea el código QR y dependiendo del dispositivo se utiliza biometría facial para utilizar la llave y terminar la autenticación.

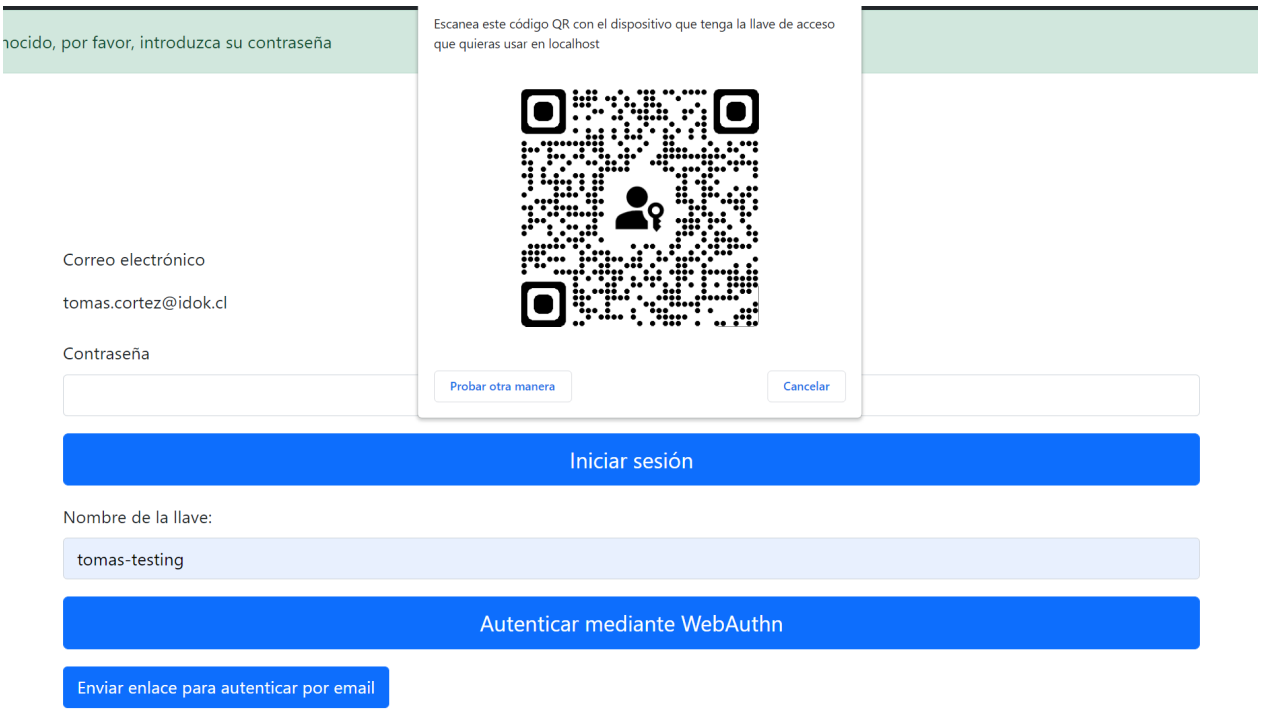

Figura 4.24: Petición de contraseña.

De esta forma el usuario se autentica y termina el flujo.

Capítulo 5

Validación

A continuación se explicará como la solución resolverá el problema planteado y cuán bien lo hace, esto se hará viendo que cumpla con los casos de uso planteados, la usabilidad que presenta y si cuenta con estándares de seguridad necesarios.

Empezando por los casos de uso que se propusieron en la sección 4.2, estos fueron completados en su totalidad, repasando los puntos importantes relacionados con el protocolo OIDC se tiene que:

- Un usuario puede autenticarse en la aplicación cliente utilizando su cuenta de la aplicación proveedora.
- La aplicación cliente puede recordar la sesión de la cuenta del usuario.
- La aplicación proveedora puede recordar la sesión de la cuenta del usuario.
- El usuario puede autenticarse en la aplicación proveedora para continuar el protocolo.
- La aplicación proveedora puede recordar los *scopes* elegidos por un usuario.
- La aplicación solicita que el usuario seleccione los *scopes* si no tiene registro anterior de estos.
- El usuario puede negar los *scopes* terminando así el flujo.
- El usuario puede cerrar sesión en la aplicación cliente.
- El usuario puede olvidar su sesión en la aplicación proveedora para que el flujo empiece desde cero.

Del punto de vista de la administración de usuarios de la aplicación de *Open ID provider* la aplicación cumple con lo siguiente:

- Usuario puede crear su cuenta con su correo electrónico.
- La aplicación solicita datos obligatorios como rut y nombres.
- Un usuario puede editar su perfil para agregar datos extra como dirección y teléfono.

- Un usuario puede iniciar y cerrar sesión en la aplicación proveedora.
- Un usuario puede cambiar la contraseña de su cuenta.
- Un usuario puede cambiar el correo electrónico de su cuenta.
- Un usuario puede autenticarse mediante su contraseña.
- Un usuario puede autenticarse mediante su correo electrónico (*passwordless*).
- Un usuario puede autenticarse con biometría facial mediante *webauthn* a través de su celular.
- Un usuario puede administrar la autorización multifactor para habilitar verificación con TOTP y códigos de recuperación.
- Una cuenta de administrador se puede crear mediante consola.
- Las cuentas de administrador pueden registrar aplicaciones cliente en la aplicación y eliminarlas.

Lamentablemente, no se pudo agregar todos los sistemas de autenticación que están en las aplicaciones de la empresa actualmente. La solución no cuenta con autenticación mediante huella digital y la autenticación provista mediante biometría facial es distinta a la utilizada en las aplicaciones productivas de la empresa, es por esta razón que aquellas metas no se cumplieron, sin embargo, se pueden agregar a la aplicación con mayor tiempo de desarrollo.

Debido a las complicaciones en el desarrollo de la aplicación no se pudo realizar las pruebas con usuarios reales que se esperaba tener, aun así se realizó una instancia de *feedback* con voluntarios tomando como métrica el cuestionario SUS (*System Usability Scale*) [23] el cual busca poder medir la usabilidad de una pieza de *software* dándole una puntuación con 68 como el promedio. En una primera instancia se llevó a cabo una primera sesión de *feedback*, la cual constaba en hacer uso de la plataforma para luego obtener el *feedback* de la experiencia. Luego de eso y resolver algunas de las preocupaciones planteadas, se llevó a cabo una segunda instancia haciendo uso del cuestionario SUS. Los voluntarios fueron miembros de la empresa, además de trabajadores de diversos rubros que cumplían el rol de posibles usuarios en la aplicación.

En la primera instancia aún no se utilizó por completo el cuestionario SUS, las primeras sesiones se llevaron a cabo recibiendo retroalimentación de la experiencia. Luego de esto, se utilizó el cuestionario. Los comentarios recibidos mencionan algunas inconsistencias en el idioma, la falta de componente visual en la validación del RUT al crear usuario y falta de explicación en ciertas acciones como la activación de *WebAuthn* para la autenticación. De las cinco personas que contestaron el cuestionario SUS junto con el feedback se obtuvo las siguientes puntuaciones: 47.5, 55, 57.5, 55 y 50, lo que da un promedio de 53, por debajo de la puntuación promedio de 68.

Luego de obtener esos resultados se hizo cambios en la plataforma para resolver los problemas y se hizo una nueva ronda de *feedback*, esta vez intentando recibir la colaboración de voluntarios diferentes y utilizando el cuestionario SUS para todos. Se obtuvo los siguientes resultados; 65, 67.5, 62.5, 60, 65, 57.5 y 67.5, obteniendo un promedio de 63.5 puntos, mejorando bastante la puntuación anterior. Otros comentarios obtenidos de la experiencia explicaban que el diseño de la página no da seguridad al navegar en esta, será necesario cambiar los estilos a futuro. Se pueden ver todos los resultados en el anexo.

La seguridad del sistema es de suma importancia para el éxito del mismo, es por esta razón que se debe hacer un análisis acerca de lo seguro del sistema. Empezando por la aplicación de *Open ID provider*, esta al utilizar *Rodauth* provee cierta seguridad con respecto a la administración de contraseñas. Como se indica en la sección 4.2.3 la base de datos es la única que tiene acceso a los hash de contraseñas, a diferencia de la aplicación, por lo que un atacante que tenga una cuenta de administrador no tiene acceso a los hash de contraseñas. Además de esto, se menciona también el uso de Bcrypt con "salçomo función de hash para evitar los ataques de tablas de hash y fuerza bruta, teniendo así contraseñas seguras para la aplicación.

Por otro lado, se implementaron cuentas de administrador, las cuales solamente se pueden crear por consola, además de tener mayores requerimientos de seguridad como necesitar una complejidad mayor de contraseña y tener que utilizar de forma obligada autenticación multifactor agregando así una nueva capa de seguridad.

Respecto a la seguridad, al implementar el protocolo OIDC hay ciertos puntos que son importantes, para empezar se utiliza el flujo del código de autorización visto en la sección 3.4.2, que a diferencia de los flujos implícito e híbrido vistos en la sección 3.4.3, es más seguro, ya que el traspaso entre el ID y secreto del cliente con los tókenes de autorización y ID se hace directamente entre el cliente y el proveedor sin pasar por el navegador, haciéndolo de esta forma más seguro.

Además de esto, lo anteriormente mencionado no se recibe por una petición GET, sino que solamente se puede hacer por POST, además que no se aceptan sitios que no utilicen el protocolo HTTPS (A excepción de en el entorno de desarrollo), esto junto con el uso de *JWT* como se menciona en la sección 3.4 minimiza el robo de tókenes de autenticación.

Capítulo 6

Conclusiones y trabajo a futuro

En este trabajo se analizó la problemática de BPO Advisors empresa la cual cuenta con distintas plataformas para satisfacer las necesidades de sus clientes y cada una tiene su propio sistema de usuario y autenticación. Esto provoca que las personas que son usuarios en más de una aplicación tengan que tener dos cuentas distintas que pueden tener distintas opciones para autenticación, aunque se tenga la información en manos de la empresa.

Con esta problemática en mente se diseñó y desarrolló una aplicación proveedor de Open ID Connect la cual será la que administre los usuarios y permita que las aplicaciones de la empresa puedan autenticarse.

Con el desarrollo de la aplicación completa se puede analizar los objetivos cumplidos por esta. Tomando en cuenta los objetivos específicos desarrollados en la sección 2.3.2 se logró diseñar el modelo de usuario con la información pertinente y común al modelo de usuario de las otras aplicaciones, por lo que el primer objetivo está completo, también se diseñó y desarrolló la aplicación que hace uso del protocolo OIDC, por lo que recibe las peticiones de autenticación y permite devolver la información para completar el proceso, cumpliendo así el segundo objetivo. Si bien la solución cuenta con más de un método para poder autenticarse, estos no son los que usa específicamente la empresa, por lo que, aunque se pueden agregar en un futuro, no se cumple el punto tres de los objetivos específicos. El último punto de los objetivos no se desarrolló debido a los cambios en la planificación.

A pesar de lo anterior, se logró dar con un producto el cual permite tener un sistema de autenticación general y unificado para las aplicaciones de la empresa, el cual también permite la organización de las cuentas de usuario, como indica el objetivo descrito en la sección 2.3.1. Sin embargo, no se pudo unificar las bases de datos de usuarios en este sistema, por lo que la problemática no fue resuelta.

Como se mencionó anteriormente, hubo objetivos que no pudieron ser completados en su totalidad, por lo que aún hay trabajo que se puede realizar con el proyecto. De las quince semanas que se

tenían presupuestadas para el proyecto, se comenzó de buena forma en las primeras, obteniendo de todos los distintos sistemas los datos que tenían en común en sus usuarios para comenzar a planificar los modelos a hacer, más adentrado en el proyecto hubo complicaciones para poder coordinar reuniones con la gente pertinente en la empresa, lo cual llevó a que el proyecto fuese a un ritmo más lento del que se buscaba. Cuando se llegó a la fase de implementar el protocolo OIDC, el cual se buscaba completar en una semana, se presentaron los problemas mencionados anteriormente, lo que llevó a que la tarea se alargase el doble, consumiendo 8 semanas, prácticamente la mitad del tiempo del proyecto, esto provocó que se tuviese que dejar de lado ciertas tareas para lograr lo prioritario de la solución.

En retrospectiva se pueden reconocer ciertos aspectos a mejorar y corregir respecto al desarrollo de la memoria. Empezando por las decisiones de lenguajes de programación y librerías, pudo haber sido necesario llevar una investigación más profunda de las opciones disponibles, sus ventajas y desventajas, si bien las decisiones tomadas tenían fundamento, un análisis más profundo pudo haber cambiado algunas decisiones. Por ejemplo, al analizar la librería para implementar el protocolo OIDC.

Respecto a la planificación, esta fue muy optimista, ya que se pronosticó que el período de implementación del protocolo OIDC sería mucho menor al que se obtuvo en realidad, si bien se tuvo en consideración que sería la fase que iba a requerir más trabajo y que probablemente podría haber alguna eventualidad, no se consideró todo el tiempo que se tuvo que utilizar para resolver los problemas que hubo. Bajo esta misma línea para el trabajo práctico se debió haber restado tiempo de cosas que no eran del todo esenciales, por ejemplo, a la hora de tener la aplicación base con la administración de usuarios, ciertas cosas como el agregar autenticación multifactor no eran cruciales para la resolución del problema, si bien son un buen agregado la solución podía valerse por sí misma sin esto y dejarse como trabajo a futuro.

Debido a que los productos que ofrece la empresa están expandiéndose y obteniendo nuevos usuarios conforme pasa el tiempo, el trabajo realizado en este semestre será de gran impacto una vez se adapte en todas las plataformas, sin embargo, por esta misma razón es importante que se encuentre pulido en su totalidad y se lleve a cabo todo el trabajo adicional que se requiera realizar, más información adelante en la sección 6.2. Pese a que es un trabajo importante y de gran valor para la empresa, no es urgente en estos momentos ni en el futuro cercano y se tiene actualmente como un proyecto deseado para proveer un mejor servicio, por esta razón se puede llevar a cabo otra iteración del proyecto para alcanzar a cumplir todos los objetivos y asegurarse de tener una aplicación completa, íntegra y por sobre todo segura.

La experiencia vivida al llevar a cabo esta memoria y el posterior análisis que se hizo del desarrollo de ésta, permite rescatar lecciones que se deben tener en cuenta para proyectos futuros. Como ya se mencionó, es importante hacer una buena planificación de los tiempos a utilizar en

cada una de las tareas de un proyecto y trabajos como este permiten reunir la experiencia necesaria para llevar a cabo planificaciones más precisas y adecuadas.

Respecto al gran obstáculo que se hizo presente en este proyecto, las librerías, se puede llegar a la conclusión que se deben investigar y analizar con cuidado, especialmente cuando están relacionadas al ámbito de la seguridad computacional. En muchas actividades relacionadas a la programación se trabaja con librerías las cuales se asume funcionarán de manera correcta y que habrá información necesaria para utilizarse, como por ejemplo en los cursos de la universidad. Sin embargo, en este proyecto se presentó esa problemática, en primera instancia con una librería que funcionaba correctamente, pero que su falta de documentación impedía el uso correcto de ésta sin conocimientos específicos del protocolo *Oauth 2*, lo cual se puede solventar con el tiempo adecuado, pero por el otro lado la segunda librería presentaba problemas que debían ser resueltos, aún estaba en desarrollo, por lo que presentaba algunas falencias importantes. Debido a esto hay que prestar debida atención, ya que Rodauth, una librería que está completa y es robusta, tenía en su documentación a Rodauth Oauth como proyectos relacionados hechos por otros autores, siendo que esto no estaba del todo pulido en su totalidad.

Por último, el llevar a cabo un proyecto principalmente solo y con restricción de tiempo (a diferencia de un proyecto personal), permite entender de mejor manera las etapas necesarias y cómo se llevan a cabo estas para que un trabajo pueda completarse, a diferencia de las labores que se realizan con un equipo de trabajo en el cual las tareas se delegan y hacen en conjunto.

6.1. Retrospectiva

Luego del análisis anterior se puede pensar en la forma en la cual se pudo haber hecho el proyecto si se contase con la experiencia que se obtuvo al llevar a cabo la memoria. Para empezar con el planteamiento del proyecto, el utilizar OIDC como protocolo que permite autenticar usuarios gracias a una aplicación proveedora que los administra es un acierto, ya que esto permite dirigir esfuerzos y recursos para mantener esa aplicación segura, en vez de que cada aplicación tenga su sistema de autenticación, lo que puede llevar a que alguno sea más vulnerable.

Sin embargo, se debería hacer un análisis más exhaustivo acerca de las tecnologías a utilizar en este proyecto, específicamente lenguajes de programación y librerías para utilizar el protocolo OIDC. Como se mencionó en el diseño de la solución, el utilizar *Ruby* y *Ruby on Rails*, presenta un menor costo para la empresa, ya que no se requiere capacitación para una futura mantención, sin embargo, en esa etapa del proyecto no se vislumbraron los problemas que acarrearía esa decisión. *Ruby* no es un lenguaje de programación tan popular, por lo que no cuenta con las mismas opciones como *Python* por dar un ejemplo, el cual cuenta con mayor cantidad de librerías y se puede investigar más de estas, por lo que se debería hacer un análisis que permita decidir bien con la nueva información.

Si de todas formas se decide utilizar *Ruby on Rails*, la decisión de la librería también es importante, contando con más tiempo de proyecto, una idea puede ser contactar al creador de la gema *OpenID Connect* o abrir un *ticket* en el repositorio de ésta para resolver el problema que se presentó. Considerando la otra opción, puede que en un futuro la gema *Rodauth Oauth* resuelva los percances que presenta, pero por el momento es una inconveniencia en el mejor de los casos y un riesgo de seguridad en el peor.

Por último, en el contexto de este proyecto fue una buena idea el permitirse dejar de lado un problema e intentar con una solución alternativa, ya que agilizó el proceso y a pesar de todo se pudo obtener un resultado satisfactorio, mientras que si se mantenía la solución inicial se pudo haber llegado a nada concreto en una última instancia.

6.2. Trabajo a futuro

Debido a las limitaciones de tiempo y alcance de este proyecto, hay ideas, mejoras y validaciones que no se lograron desarrollar, sin embargo, pueden proveer gran utilidad al proyecto y a resolución de la problemática planteada.

Comenzando por el proyecto desarrollado y tomando en cuenta que se espera contar con miles de usuarios utilizando la plataforma, es pertinente que se lleven a cabo pruebas de estrés para asegurar un correcto y expedito funcionamiento en la mayoría de los casos, ya que la mayoría de los servicios de la empresa dependerán del proyecto para su funcionamiento.

Por otro lado, tomando en cuenta que la librería que cuenta con la certificación de la fundación *Open ID* no fue utilizada, se propone que se lleven a cabo las pruebas para obtener la certificación en este sistema, estas pruebas y validaciones se pueden llevar a cabo y así asegurar un estándar de calidad y seguridad en el protocolo *Open ID Connect*

Tomando en consideración la importancia de la seguridad informática para esta pieza de software se pueden agregar registros para auditoría, lo que permite tener constancia de los cambios en la plataforma. La gema *Rodauth* provee la capacidad de generar *audit logging* de las acciones de su *framework* de autenticación, por lo que se podría complementar y expandir esto.

Otros agregados que se le puede hacer a la aplicación es agregar soporte con otros métodos de autenticación como cuentas de Google o Microsoft y por último adecuar el diseño y estilo al utilizado por la empresa.

6.3. Trabajo relacionado

Complementando lo ya mencionado hay trabajo futuro que puede ser considerado como proyectos por sí mismos, debido a los alcances que tienen y la cantidad de recursos necesarios para

llevarlos a cabo.

El primero de estos proyectos es la incorporación de la aplicación cliente utilizada en este proyecto a las otras aplicaciones de la empresa, junto con las validaciones y reestructuración de la administración de usuarios asociada a este cambio. Relacionado a lo anterior sería beneficioso llevar a cabo una migración de datos a la nueva plataforma, se debe procurar que no se pierdan datos y que estos se traspasen de forma segura, íntegra y privada. Si bien ambas cosas son necesarias para resolver la problemática, el alcance de estos y los recursos necesarios para llevarlos a cabo, son mayores al que se presupuestaba en un comienzo, por lo que deberán hacerse como trabajo futuro.

Por último, se plantea un proyecto que permita aumentar la seguridad en la aplicación utilizando un sistema que pueda detectar actividad sospechosa (por ejemplo cambio reciente de contraseña o fallos múltiples en métodos de autenticación) y actualizar la cantidad de autenticación multifactor necesaria para ingresar, logrando que la verificación de un usuario sea más estricta en caso de ser necesaria.

6.4. Palabras finales

A modo de conclusión se puede decir que la problemática que presenta BPO Advisors será resuelta con la solución propuesta, diseñada y desarrollada, sin embargo, debido a complicaciones y restricciones por la naturaleza de la memoria, necesita más trabajo para poder llegar a una etapa que pueda utilizarse en producción.

Bibliografía

[1] Krebs, B., The Open ID Connect Handbook. 2019.

[2] Werner, C., Wike, R., Richards, O., Macy, M., Waweru, E., Coulter, D., y Adman, N., "Authentication vs. authorization.", https://docs.microsoft.com/en-us/azure/active-directory/develop/authentication-vs-authorization.

[3] OAuth, "What is oauth.", https://oauth.net/.

[4] Microsoft, "Modern authentication: how we got here – microsoft identity platform," 2020, https://www.youtube.com/watch?v=7_vxnHiUA1M&list=PLLasX02E8BPBxGouWlJV-u-XZWOc2RkiX.

[5] OpenIDFoundation, "Welcome to openid connect.", https://openid.net/connect/.

[6] OpenIDFoundation, "Openid connect faq and q&as.", https://openid.net/connect/faq/.

[7] Mozilla, "Oidc in a nutshell.", https://infosec.mozilla.org/guidelines/iam/openid_connect.html.

[8] Oracle, "¿qué es lenguaje de marcado para confirmaciones de seguridad (saml)?.", https://www.oracle.com/cl/security/cloud-security/what-is-saml/.

[9] MIT, "What is kerberos?.", https://web.mit.edu/kerberos/\#what_is.

[10] IBM, "Visión general de radius (remote authentication dial in user service)," 2021, https://www.ibm.com/docs/es/i/7.1?topic=authentication-remote-dial-in-user-service-overview.

[11] Google, "Autenticación.", https://cloud.google.com/identity-platform/docs/concepts-authentication.

[12] Microsoft, "Microsoft identity platform documentation.", https://docs.microsoft.com/en-us/azure/active-directory/develop/.

[13] Curity, "Pairwise pseudonymous identifiers.", https://curity.io/resources/learn/ppid-intro/.

[14] IBM, "Oauth 2.0 and oidc workflows," 2021, https://www.ibm.com/docs/en/sva/9.0.5?topic=SSPREK_9.0.5/com.ibm.isam.doc/config/concept/con_oauth20_workflow.html.

[15] PostgreSQL, "About.", https://www.postgresql.org/about/.

[16] TudipDigital, "What is openid connect authentication and benefits of using openid connect?.", https://tudip.com/blog-post/openid-connect-authentication/.

[17] OpenIDFoundation, "Certified openid connect implementations.", https://openid.net/develop

ers/certified/.

[18] OpenIDFoundation, “Openid certification frequently asked questions (faq).”, https://openid.net/certification/faq/.

[19] OneLogin, “Oidc vs saml: All you need to know.”, https://www.onelogin.com/learn/oidc-vs-saml.

[20] Evans, J., “Documentation for rodauth (v2.26.1).”, http://rodauth.jeremyevans.net/documentation.html.

[21] Arias, D., 2021, https://auth0.com/blog/hashing-in-action-understanding-bcrypt/.

[22] 37signals LLC, “The speed of a single-page web application without having to write any javascript.,” 2021, https://turbo.hotwired.dev/.

[23] Brooke, J., “Sus: A quick and dirty usability scale,” 1995, https://www.researchgate.net/publication/228593520_SUS_A_quick_and_dirty_usability_scale.

Capítulo 7

Anexo

Validación con cuestionario SUS

A continuación se presentan los resultados del cuestionario SUS para la primera ronda de *feedback*:

	Totalmente en desacuerdo	*En desacuerdo*	*Neutro*	*De acuerdo*	*Totalmente de acuerdo*
Creo que me gustaría utilizar este sistema con frecuencia			X		
Encontré el sistema innecesariamente complejo	X				
Pensé que el sistema era fácil de usar			X		
Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema				X	
Encontré que las diversas funciones de este sistema estaban bien integradas			X		
Pensé que había demasiada inconsistencia en este sistema		X			
Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente				X	
Encontré el sistema muy complicado de usar			X		
Me sentí muy seguro usando el sistema	X				
Necesitaba aprender muchas cosas antes de empezar con este sistema			X		

Figura 7.1: Respuestas de la primera ronda de feedback.

	Totalmente en desacuerdo	*En desacuerdo*	*Neutro*	*De acuerdo*	*Totalmente de acuerdo*
Creo que me gustaría utilizar este sistema con frecuencia			X		
Encontré el sistema innecesariamente complejo			X		
Pensé que el sistema era fácil de usar				X	
Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema			X		
Encontré que las diversas funciones de este sistema estaban bien integradas			X		
Pensé que había demasiada inconsistencia en este sistema				X	
Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente				X	
Encontré el sistema muy complicado de usar		X			
Me sentí muy seguro usando el sistema	X				
Necesitaba aprender muchas cosas antes de empezar con este sistema			X		

Figura 7.2: Respuestas de la primera ronda de feedback.

	Totalmente en desacuerdo	*En desacuerdo*	*Neutro*	*De acuerdo*	*Totalmente de acuerdo*
Creo que me gustaría utilizar este sistema con frecuencia				X	
Encontré el sistema innecesariamente complejo		X			
Pensé que el sistema era fácil de usar			X		
Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema			X		
Encontré que las diversas funciones de este sistema estaban bien integradas				X	
Pensé que había demasiada inconsistencia en este sistema			X		
Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente			X		
Encontré el sistema muy complicado de usar			X		
Me sentí muy seguro usando el sistema		X			
Necesitaba aprender muchas cosas antes de empezar con este sistema		X			

Figura 7.3: Respuestas de la primera ronda de feedback.

	Totalmente en desacuerdo	En desacuerdo	Neutro	De acuerdo	Totalmente de acuerdo
Creo que me gustaría utilizar este sistema con frecuencia	X				
Encontré el sistema innecesariamente complejo			X		
Pensé que el sistema era fácil de usar		X			
Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema				X	
Encontré que las diversas funciones de este sistema estaban bien integradas					X
Pensé que había demasiada inconsistencia en este sistema		X			
Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente				X	
Encontré el sistema muy complicado de usar			X		
Me sentí muy seguro usando el sistema	X				
Necesitaba aprender muchas cosas antes de empezar con este sistema		X			

Figura 7.4: Respuestas de la primera ronda de feedback.

	Totalmente en desacuerdo	En desacuerdo	Neutro	De acuerdo	Totalmente de acuerdo
Creo que me gustaría utilizar este sistema con frecuencia			X		
Encontré el sistema innecesariamente complejo			X		
Pensé que el sistema era fácil de usar				X	
Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema			X		
Encontré que las diversas funciones de este sistema estaban bien integradas			X		
Pensé que había demasiada inconsistencia en este sistema	X				
Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente			X		
Encontré el sistema muy complicado de usar		X			
Me sentí muy seguro usando el sistema	X				
Necesitaba aprender muchas cosas antes de empezar con este sistema			X		

Figura 7.5: Respuestas de la primera ronda de feedback.

A continuación se presentan los resultados del cuestionario SUS para la segunda ronda de *feedback*:

	Totalmente en desacuerdo	En desacuerdo	Neutro	De acuerdo	Totalmente de acuerdo
Creo que me gustaría utilizar este sistema con frecuencia			X		
Encontré el sistema innecesariamente complejo		X			
Pensé que el sistema era fácil de usar				X	
Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema		X			
Encontré que las diversas funciones de este sistema estaban bien integradas				X	
Pensé que había demasiada inconsistencia en este sistema	X				
Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente		X			
Encontré el sistema muy complicado de usar		X			
Me sentí muy seguro usando el sistema		X			
Necesitaba aprender muchas cosas antes de empezar con este sistema	X				

Figura 7.6: Respuestas de la segunda ronda de feedback.

	Totalmente en desacuerdo	En desacuerdo	Neutro	De acuerdo	Totalmente de acuerdo
Creo que me gustaría utilizar este sistema con frecuencia			X		
Encontré el sistema innecesariamente complejo		X			
Pensé que el sistema era fácil de usar				X	
Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema		X			
Encontré que las diversas funciones de este sistema estaban bien integradas				X	
Pensé que había demasiada inconsistencia en este sistema		X			
Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente		X			
Encontré el sistema muy complicado de usar		X			
Me sentí muy seguro usando el sistema				X	
Necesitaba aprender muchas cosas antes de empezar con este sistema			X		

Figura 7.7: Respuestas de la segunda ronda de feedback.

	Totalmente en desacuerdo	En desacuerdo	Neutro	De acuerdo	Totalmente de acuerdo
Creo que me gustaría utilizar este sistema con frecuencia			X		
Encontré el sistema innecesariamente complejo			X		
Pensé que el sistema era fácil de usar				X	
Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema			X		
Encontré que las diversas funciones de este sistema estaban bien integradas				X	
Pensé que había demasiada inconsistencia en este sistema		X			
Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente				X	
Encontré el sistema muy complicado de usar		X			
Me sentí muy seguro usando el sistema	X				
Necesitaba aprender muchas cosas antes de empezar con este sistema			X		

Figura 7.8: Respuestas de la segunda ronda de feedback.

	Totalmente en desacuerdo	En desacuerdo	Neutro	De acuerdo	Totalmente de acuerdo
Creo que me gustaría utilizar este sistema con frecuencia				X	
Encontré el sistema innecesariamente complejo			X		
Pensé que el sistema era fácil de usar			X		
Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema			X		
Encontré que las diversas funciones de este sistema estaban bien integradas					X
Pensé que había demasiada inconsistencia en este sistema			X		
Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente			X		
Encontré el sistema muy complicado de usar		X			
Me sentí muy seguro usando el sistema			X		
Necesitaba aprender muchas cosas antes de empezar con este sistema	X				

Figura 7.9: Respuestas de la segunda ronda de feedback.

	Totalmente en desacuerdo	En desacuerdo	Neutro	De acuerdo	Totalmente de acuerdo
Creo que me gustaría utilizar este sistema con frecuencia			X		
Encontré el sistema innecesariamente complejo		X			
Pensé que el sistema era fácil de usar				X	
Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema		X			
Encontré que las diversas funciones de este sistema estaban bien integradas				X	
Pensé que había demasiada inconsistencia en este sistema	X				
Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente		X			
Encontré el sistema muy complicado de usar	X				
Me sentí muy seguro usando el sistema		X			
Necesitaba aprender muchas cosas antes de empezar con este sistema		X			

Figura 7.10: Respuestas de la segunda ronda de feedback.

	Totalmente en desacuerdo	En desacuerdo	Neutro	De acuerdo	Totalmente de acuerdo
Creo que me gustaría utilizar este sistema con frecuencia				X	
Encontré el sistema innecesariamente complejo		X			
Pensé que el sistema era fácil de usar			X		
Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema		X			
Encontré que las diversas funciones de este sistema estaban bien integradas			X		
Pensé que había demasiada inconsistencia en este sistema		X			
Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente		X			
Encontré el sistema muy complicado de usar		X			
Me sentí muy seguro usando el sistema			X		
Necesitaba aprender muchas cosas antes de empezar con este sistema		X			

Figura 7.11: Respuestas de la segunda ronda de feedback.

	Totalmente en desacuerdo	En desacuerdo	Neutro	De acuerdo	Totalmente de acuerdo
Creo que me gustaría utilizar este sistema con frecuencia			X		
Encontré el sistema innecesariamente complejo			X		
Pensé que el sistema era fácil de usar					X
Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema			X		
Encontré que las diversas funciones de este sistema estaban bien integradas				X	
Pensé que había demasiada inconsistencia en este sistema		X			
Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente				X	
Encontré el sistema muy complicado de usar		X			
Me sentí muy seguro usando el sistema	X				
Necesitaba aprender muchas cosas antes de empezar con este sistema			X		

Figura 7.12: Respuestas de la segunda ronda de feedback.