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

FONASA - GESTIÓN DE USUARIOS. LDAP EN LAS ORGANIZACIONES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

HERNÁN ALIRO SILVA VARGAS

PROFESOR GUÍA:
MARÍA CECILIA BASTARRICA PIÑEYRO

MIEMBROS DE LA COMISIÓN:
ROMAIN ROBBES
PABLO GONZALEZ JURE

SANTIAGO DE CHILE
MARZO 2014

Resumen

El Fondo Nacional de Salud (Fonasa) es un organismo del Estado de Chile cuya función es proporcionar a sus *beneficiarios*[1] una serie de prestaciones, de acuerdo a la legislación vigente.

Recientemente, como parte de la modernización del aparato estatal y sus reparticiones, Fonasa ha iniciado un Proyecto Global de gran envergadura, consistente en el cambio, mejora y ampliación de todos sus sistemas corporativos, incluyendo plataformas, procesos y sistemas computacionales. Su objetivo último es agilizar y mejorar sus servicios, a nivel central y regional, tanto para usuarios y beneficiarios, como en su gestión interna. Es un proyecto de largo aliento, que indudablemente tendrá un gran impacto a nivel nacional.

En la actualidad Fonasa muestra una gran variedad de plataformas y sistemas, con mecanismos propios de seguridad y control, con múltiples esquemas de manejo de usuarios, con escasa correlación entre los sistemas, y sin posibilidad de efectuar una administración centralizada de usuarios y permisos.

Por esta razón, el nuevo sistema Gestión de Usuarios, que se ocupará precisamente de estas materias, resulta ser una componente fundamental en este Proyecto Global. Este sistema dará cobertura única y transversal, en todos los aspectos relacionados con el manejo, administración y gestión de los usuarios de Fonasa, así como el control de accesos y permisos asociados.

En este contexto aparece la tecnología LDAP (Lightweight Directory Access Protocol), un conjunto de protocolos estándar para el acceso a servicios de Directorio en un entorno de red. Estos Directorios proveen mecanismos eficientes de almacenamiento y recuperación de información, y funciones básicas para modificación.

El uso de LDAP en las organizaciones está siendo cada vez más extendido. Los LDAP ofrecen una serie de beneficios y también imponen algunas restricciones. Dadas sus características, que veremos en detalle en esta Memoria, un LDAP puede resultar una significativa contribución en determinados escenarios, y particularmente en la Gestión de Usuarios.

El tema del presente Proyecto de Título aborda estas materias, y expone un caso concreto de uso exitoso de LDAP en la implementación del sistema Gestión de Usuarios de Fonasa.

[1] personas que pueden hacer uso de los servicios y beneficios de esta organización

A nuestra querida "Chachi".

Agradecimientos

Vayan mis agradecimientos a mi familia, a mis compañeros de trabajo, y particularmente a la gente de Fonasa, que siempre tuvo una gran disposición a explicar conceptos, repetir las pruebas y sobre todo, por su gran paciencia y entusiasmo. También a consultores y auditores del proyecto y a los "IBM boys".

Además mi sincero reconocimiento a la Escuela de Ingeniería de la Universidad de Chile, a sus funcionarios y profesores, y especialmente a mi profesora guía, Cecilia Bastarrica.

Tabla de contenido

Índice de tablas

Índice de figuras

Capítulo 1

Introducción

Un Lightweight Directory Access Protocol (LDAP) es un conjunto de protocolos estándar para el acceso a servicios de Directorio en un entorno de red. Los Directorios proveen mecanismos de almacenamiento y recuperación de información de diversa índole, así como funciones básicas para su modificación.

El uso de los LDAP en las organizaciones está siendo cada vez más extendido. Los LDAP ofrecen una serie de beneficios, pero también imponen algunas restricciones. Un LDAP puede resultar muy apropiado en ciertos escenarios, así como completamente inadecuado en otras circunstancias.

El tema de Proyecto de Título abordará estas materias y presentará un caso concreto de utilización de la tecnología LDAP en el desarrollo del sistema Gestión de Usuarios de Fonasa.

1.1 Introducción

1.1.1. Fonasa y su Proyecto

El Fondo Nacional de Salud (Fonasa) es un organismo del Estado de Chile, cuyo objetivo es proporcionar a sus beneficiarios una serie de prestaciones de salud contempladas en la ley.

Fonasa está llevando a cabo un proyecto de cambio global, de gran envergadura, a través de la construcción de una nueva plataforma integrada de sistemas y servicios, que de soporte a sus procesos de negocios, satisfaciendo las necesidades de Fonasa y sus beneficiarios, a lo largo de todo el país.

Para dar cumplimiento a este objetivo, Fonasa hizo un llamado a licitación, en la que consideró el análisis de procesos, desarrollo y outsorcing [1] de la nueva plataforma. Esta licitación fue adjudicada a la empresa Adexus.

Entre los sistemas que debe implementar se encuentra el sistema Gestión de Usuarios.

Figura 1.1: salud Fonasa

Este sistema es una oportunidad ideal para el uso de un LDAP que resuelva esta problemática, y que sea capaz de dar soporte y cabida a los nuevos sistemas, tanto los que se encuentran en desarrollo como aquellos que serán construidos a futuro. Por ello resulta también una inmejorable instancia para poner a prueba los conceptos de uso de LDAP en una organización.

[1] diversos servicios tipo data center, hosting, housing, procesamiento, etc.

1.1.2. Fases del Proyecto Global de Fonasa

Teniendo en mente que existe una gran cantidad y variedad de sistemas a desarrollar, Fonasa ha decidido dividir su Proyecto Global en fases o etapas (bloques de sistemas).

De esta forma, habrá una primera fase con un grupo de sistemas que se desarrollarán e implementarán en conjunto, y luego fases adicionales donde se irán incorporando el resto de los sistemas requeridos.

En el caso de Gestión de Usuarios, se trata de un sistema transversal a toda la plataforma y por tanto forma parte integral de los sistemas a implementar en la primera Fase del Proyecto Global.

1.1.3. Gestión de Usuarios - Metodología

¿Cómo se ha abordado la solución de este proyecto?

La estrategia adoptada para llevar a cabo este proyecto fue la siguiente:

- ▷ Establecer un Marco Teórico global para el desarrollo, construcción e implantación del sistema Gestión de Usuarios (es decir, cuáles son los elementos fundamentales que deben ser considerados).
- ▷ Paralelamente efectuar el levantamiento detallado de requerimientos del sistema.
- ▷ Con estos elementos, definir un modelo sustentable para la Gestión de Usuarios de Fonasa.
- ▷ Analizar y elegir dentro de las tecnologías disponibles (construir o, idealmente, utilizar componentes COTS [2], específicamente LDAP).
- ▷ Diseñar e Implementar.

1.1.4. LDAP - Una descripción inicial

Definición

Un Lightweight Directory Access Protocol (LDAP) es un conjunto de protocolos estándar para el acceso a servicios de directorio, que tienen como objetivo permitir la recuperación de diversos tipos de información en un entorno de red.[3]

La tecnología de directorios ofrece la promesa de resolver el problema de recuperar información descentralizada y rápidamente, en un ambiente informático distribuido.

[2] Commercial Off The Shelf
[3] Donley [6, capítulo 1]

Un objetivo frecuente del LDAP es administrar la funcionalidad de aplicaciones distribuidas y facilitar su uso. Esta información es de diversa índole, y puede referirse a los servicios, recursos, usuarios o cualquier otro objeto accesible o compartible entre las aplicaciones.

Directorio

Un Directorio es un mecanismo de almacenamiento de información especialmente diseñado y orientado a búsqueda y navegación. Además cuenta con mecanismos básicos para la modificación de dicha información. Desde ese punto de vista, se menciona que los directorios están optimizados para acceso de tipo lectura, aunque esto resulta ser una definición muy simplificada.

Como tal, el mecanismo de almacenamiento (backing store) puede ser una base de datos, un archivo o algún otro medio, el que debe ser transparente al usuario de la información.

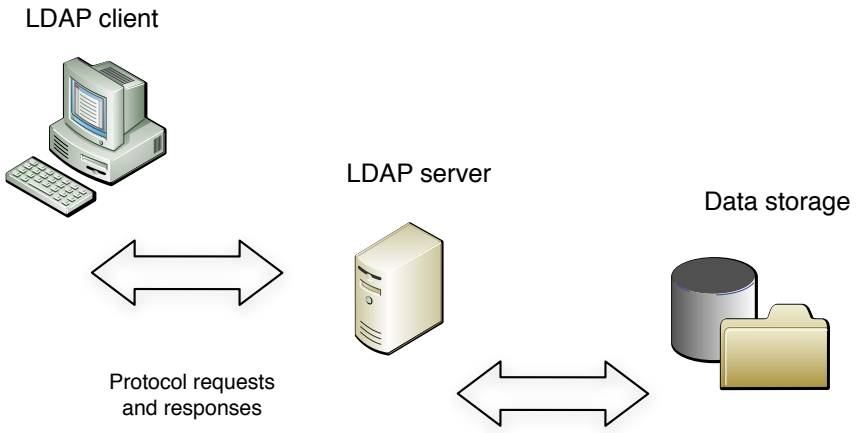


Figura 1.2: LDAP

Protocolo

LDAP es un protocolo que actualmente se encuentra en la versión LDAPv3. Como tal, las aplicaciones que utilicen el LDAP deben ajustarse a las normas y reglas definidas.

Algunas aplicaciones típicas

Algunos usos comunes de los LDAP son:

- ▷ Autenticación de máquinas
- ▷ Autenticación de usuarios
- ▷ Libros de direcciones

- ▷ Representación del organigrama de una organización
- ▷ Almacenamiento de configuración de aplicaciones

La elección de un LDAP constituye una decisión de arquitectura global de las aplicaciones y debe ser tomada tempranamente en el transcurso de la vida de un proyecto.

1.2 El Área Problema

1.2.1. Sistema Gestión de Usuarios

Entre los requerimientos de calidad más importantes para Fonasa se encuentra la seguridad de información y aplicaciones, así como el estricto control y seguimiento de los accesos a los sistemas.

El nuevo sistema Gestión de Usuarios es un aplicación transversal a toda la plataforma Fonasa. Como su nombre lo indica, permitirá gestionar y administrar a los usuarios de todas las aplicaciones, sus permisos y atribuciones, altas y bajas de usuarios, así como los cambios que experimenten durante su permanencia en la organización. Asimismo, el sistema Gestión de Usuarios debe proveer mecanismos para que todas las demás aplicaciones y sistemas puedan utilizar esta información, y cumplir así con sus propias necesidades.

1.2.2. ¿Cuál es el rol que juega LDAP en esta materia?

Un LDAP nos proporciona:

- ▷ una estructura de directorios donde almacenar la información de los usuarios.
- ▷ un paradigma estándar (objetos) para modelar esta información .
- ▷ un mecanismo estándar de comunicación (servicios) entre las demás aplicaciones y el sistema Gestión de Usuarios, para efectuar labores de validación, autenticación y autorización.
- ▷ mecanismos de recuperación de la información en un ambiente distribuido.
- ▷ mecanismos de seguridad y protección de la información.
- ▷ mecanismos de replicación del directorio para sustentar el requerimiento de alta disponibilidad.
- ▷ software (COTS) especialmente orientado al manejo y desarrollo de aplicaciones que funcionan bajo el protocolo LDAP.

En este sentido un LDAP, dados sus atributos y características, resulta muy adecuado para el desarrollo del sistema Gestión de Usuarios.

Adicionalmente, Fonasa presenta particularidades propias de su negocio, las que deben ser consideradas en el diseño y aplicación de un LDAP a la solución de esta materia.

1.2.3. Relevancia/Motivación para encontrar una solución

¿Por qué es necesario el desarrollo de un sistema Gestión de Usuarios?

La Seguridad es un requerimiento global dentro de cualquier desarrollo de envergadura. Esto es válido, por supuesto, para las nuevas aplicaciones de Fonasa, y especialmente para la gestión de los usuarios.

Actualmente Fonasa exhibe una enorme variedad de plataformas y sistemas, cada uno con mecanismos propios de seguridad y control, con múltiples esquemas para manejo de usuarios, con cuentas de acceso duplicadas, etc.

Esto dificulta la gestión y control de los usuarios y proliferan las inconsistencias, Se utilizan variados y disímiles mecanismos de conexión y permisos, y los usuarios deben recordar diferentes esquemas de user ID y password. Todo esto genera una innecesaria y extenuante cantidad de tareas administrativas y de conciliación.

¿Cuáles son los objetivos que se deben lograr en un sistema Gestión de Usuarios?

- ▷ Cumplir con principios de Seguridad (normas y procedimientos).
- ▷ Cumplir con principios de Gobernabilidad de una plataforma de software.
- ▷ Desde el punto de vista de la administración:
  Control centralizado y único de la gestión de usuarios y permisos.
- ▷ Desde el punto de vista de los usuarios:
  Mecanismo único de seguridad para los usuarios y single-sign-on.
- ▷ Desde el punto de vista de los demás sistemas:
  Proveer servicios de Autenticación y Autorización para su uso por el resto de los sistemas de la plataforma.
- ▷ Sustentabilidad para dar cabida natural a los futuros desarrollos.
  Como parte del Proyecto Global de Fonasa, y a medida que se vayan completando los desarrollos de los nuevos sistemas, será necesario incorporarlos a la nueva plataforma. Esta incorporación debe estar basada en el seguimiento de principios y normas de arquitectura y desarrollo, conocidas por todos los equipos de trabajo. El cumplimiento de estas normas debe garantizar que esta incorporación preserva los mecanismos de seguridad proporcionados por el sistema Gestión de Usuarios.
- ▷ Prevención de riesgos de seguridad.
  El nuevo sistema Gestión de Usuarios debe ser diseñado de tal forma que se integre a la plataforma de Fonasa, como un elemento activo de prevención de riesgos de seguridad.
- ▷ Detección de incidencias y apoyo a su resolución.
  El sistema Gestión de Usuarios debe manejar registros históricos de actividad, generar

reportes periódicos y proporcionar los elementos necesarios para efectuar auditorías y detección de incidentes de seguridad.

1.3 Proyecto de Título, Objetivos

1.3.1. Objetivo General

El trabajo de Proyecto de Título busca cumplir con el siguiente objetivo general:

> Implementar la gestión de usuarios de la plataforma integrada de sistemas de Fonasa, mediante la utilización de un LDAP.

1.3.2. Objetivos específicos

Analizar diferentes LDAP, su arquitectura, ventajas y desventajas, y su aplicación concreta en el proyecto Gestión de Usuarios de la plataforma integrada de sistemas del proyecto Fonasa. Es decir:

1. Estudio de los LDAP y sus aplicaciones
2. Implantación del sistema Gestión de Usuarios de Fonasa

1.3.3. Descripción Objetivos específicos

Estudio de los LDAP y sus aplicaciones

1. Estudio de algunos LDAP disponibles. Específicamente, se analizará una alternativa comercial, LDAP Tivoli de la suite IBM y una opción open source, OpenLDAP.
2. Análisis de características y comparación de ambas herramientas.
3. Ventajas y desventajas.

Implantación del sistema Gestión de Usuarios de Fonasa

1. Levantamiento de requerimientos.
2. Análisis y selección de alternativas de diseño.
3. Implementación (desarrollos y customización de Producto).
4. Diseño y ejecución de pruebas funcionales (unitarias, de sistema, integradas).

5. Diseño y ejecución de pruebas no funcionales.
6. Migración de datos, piloto y puesta en marcha.

1.3.4. Cumplimiento de objetivos

¿Cómo se verificará el cumplimiento de objetivos?

Al final de este proceso se debe lograr:

1. Informe comparativo de los LDAP analizados.
2. Sistema Gestión de Usuarios en marcha blanca/producción operando correctamente.
3. Migración de datos realizada, con un porcentaje mínimo de error (en acuerdo con Fonasa se estableció un error de migración menor a 1 %).
4. Pruebas funcionales completas, clasificando eventuales errores/observaciones según su impacto en la operación (inhabilitante, menores, mejoras deseables). El sistema no debe presentar ningún error inhabilitante.
5. Pruebas no funcionales completas, a plena satisfacción del usuario.

1.4 Alternativas de solución

1.4.1. Introducción

Se analizaron diferentes escenarios para dar solución al problema planteado. En lo fundamental, las opciones son:

- ▷ Desarrollo.
- ▷ Tecnología LDAP : COTS (OpenLDAP e IBM Tivoli).

La opción desarrollo a la medida siempre es una alternativa de máxima flexibilidad. El principal problema son los plazos y los períodos de prueba asociados.

Por otro lado, el disponer de sofware empaquetado y customizable es claramente una opción muy atractiva. Se trata de aplicaciones probadas y orientadas al ámbito de negocio que se desea resolver. El mayor esfuerzo, en este caso, está concentrado en el aprendizaje de las facilidades que estas herramientas proporcionan, y el análisis del ajuste de los requerimientos usuarios hacia los paradigmas y restricciones que conllevan.

Sopesando estos antecedentes, la opción adoptada fue utilizar un COTS de tecnología LDAP.

1.4.2. LDAP

¿Por qué LDAP?

Durante los últimos años, la tecnología LDAP ha sido utilizada exitosamente en sistemas relacionados con la gestión de usuarios, y el manejo de información de los usuarios en las organizaciones. Hoy por hoy, en este ámbito es prácticamente un estándar de facto.

Si bien conceptualmente el foco de LDAP es el manejo de información en un entorno de red, este simple principio es absolutamente ajustado a lo que se requiere realizar en los sistemas de gestión de usuarios. En todo punto de la plataforma, en cada aplicación, permanentemente se requiere verificar que los usuarios sean válidos, y que estén autorizados a efectuar la acciones que solicitan realizar. Esto es, disponer de información en forma ágil y confiable, y en un entorno de red.

LDAP es una tecnología madura, estándar y en permanente mejora y revisión. Existe una base importante de empresas del área de software que proporcionan implementaciones de LDAP, así como varias iniciativas open source. Los lenguajes de programación más modernos (y los clásicos OOP [4]) ya incorporan librerías y sintaxis para utilizar esta tecnología. Además, muchas soluciones de software y servidores de aplicaciones actuales se declaran *LDAP-enabled* como un importante atributo de calidad.

1.4.3. COTS - Alternativas Open y Suite IBM

Entre las alternativas COTS disponibles y dentro del mundo open source, OpenLDAP es una excelente opción. Se trata de un proyecto estable, en plena actividad, con mucha documentación, implementado exitosamente en muchas organizaciones y full compliance con los estándares LDAP.

Por otro lado, la suite IBM Tivoli es una tecnología respaldada por la corporación IBM, utilizada en muchas empresas y organizaciones alrededor del mundo, en clientes de todos los tamaños y diferentes configuraciones, algunos con alta complejidad. Además de la tecnología LDAP estándar, cuenta con una serie de módulos adicionales que facilitan las labores del desarrollador. Por otro lado la suite agrega funcionalidad y APIs para muchos aspectos específicos a un sistema de gestión de usuarios. En este sentido, la suite IBM proporciona una cobertura funcional mucho más amplia y orientada a la materia que nos ocupa.

Tomando en consideración aspectos estratégicos, financieros y técnicos, la ventaja que representa contar con un proveedor que participe activamente en asegurar los altos niveles de uptime y tolerancia a fallas exigidos en la licitación, y el mayor grado de cobertura de los requerimientos de Fonasa, en definitiva se optó por la suite IBM Tivoli.

[4] Object Oriented Programming

1.5 La solución

1.5.1. Descripción

Para llevar a cabo el proyecto se hizo uso de la tecnología LDAP, especificamente la suite IBM Tivoli.

Las tareas fueron:

- ▷ Levantamiento formal de requerimientos y establecimiento de acuerdos y alcances de la solución.
- ▷ Diseño de la arquitectura de la solución.
- ▷ Diseño del modelo de usuarios en términos de LDAP y su customización en las componentes IBM Tivoli.
- ▷ Integración de las componentes de la suite IBM Tivoli de acuerdo a la funcionalidad cubierta.
- ▷ Proyecto paralelo de migración de datos.
- ▷ Validación de la solución (funcional y calidad), por medio de un extenso plan de pruebas.
- ▷ Plan de puesta en producción (Piloto, apoyo a Gestión del Cambio para inicio producción).

1.5.2. Resultados obtenidos con la solución aplicada

Como balance del proyecto se logró aplicar exitosamente la tecnología LDAP en el sistema Gestión de Usuarios.

Los resultados finales fueron:

- ▷ Inicio exitoso de la operación (Piloto y a nivel nacional).
- ▷ Puesta en producción en forma conjunta y simultanea con otros sistemas de la plataforma Fonasa, tanto antiguos como nuevos.

1.6 Estructura del documento

Este documento está organizado de la siguiente forma:

- ▶ Capítulo 2 - Antecedentes

  En este capítulo se establece el Marco Teórico que sustenta las decisiones de diseño

adoptadas para el desarrollo del sistema Gestión de Usuarios. Se fijan los conceptos de Seguridad y su relación con este sistema, y se describe LDAP como una tecnología que proporciona funcionalidad y atributos de calidad relevantes para el sistema Gestión de Usuarios.

- Capítulo 3 - Descripción del Problema a resolver

  Se expone en líneas generales el Proyecto Global de Fonasa y en forma detallada el nuevo sistema Gestión de Usuarios. El capítulo finaliza con el Plan de Trabajo asociado al desarrollo e implantación de este sistema.

- Capítulo 4 - Solución

  En este capítulo se describe la solución aplicada en el desarrollo e implantación del sistema Gestión de Usuarios. Comienza por explicar la estructura, metodología y visión global del proyecto. En seguida, entrando en materia, se muestran los análisis efectuados para decidir entre las alternativas de desarrollar la aplicación, versus la utilización de COTS. Además se muestra la comparativa entre dos COTS (IBM Tivoli y OpenLDAP) y se expone los motivos que llevaron a optar por la opción IBM Tivoli.
  Luego se describe la Arquitectura y detalles de la solución. Adicionalmente, se incluyen los trabajos asociados a la migración y carga inicial de datos, requeridos para la puesta en marcha del sistema.
  Finalmente se exponen las actividades de implementación, Piloto y puesta en marcha y las tareas de apoyo al proceso de Gestión del Cambio de Fonasa.

- Capítulo 5 - Validación

  Describe la estrategia y Plan de Pruebas del nuevo sistema, la ejecución de dichas pruebas y los resultados obtenidos.

- Capítulo 6 - Conclusiones

  En este capítulo se efectúa una revisión y balance final del desarrollo del sistema Gestión de Usuarios. Se establecen como conclusiones las claves del éxito del proyecto, y como subproducto relevante una metodología de desarrollo e implantación de un sistema de gestión de usuarios.
  Adicionalmente se analiza el crecimiento del sistema Gestión de Usuarios, en futuras etapas del proyecto Fonasa.
  Finalmente se analiza el uso posible de LDAP en otras áreas y aplicaciones.

Capítulo 2

Antecedentes

En este capítulo se entrega una serie de conceptos y antecedentes que resultan necesarios para el desarrollo del presente Trabajo de Título.

De esta forma se establecerá el Marco Teórico que sustenta las decisiones de diseño que fueron adoptadas durante el transcurso del proyecto.

En este Marco Teórico se consideran:

- ▷ Seguridad en una organización y su estrecha relación con la Gestión de Usuarios
- ▷ Detalle de los conceptos asociados a la tecnología LDAP

2.1 Seguridad

Si bien hablar de Seguridad en las organizaciones y sistemas excede con mucho el alcance de esta Memoria, no podemos dejar de mencionar varios aspectos que resultan relevantes a la hora de tomar decisiones para el sistema Gestión de Usuarios.

La Gestión de Usuarios y la Seguridad están íntimamente vinculados. En efecto, como veremos, la Gestión de Usuarios resulta ser una componente fundamental de todo Modelo de Seguridad de una organización.

En los párrafos siguientes se enuncian y describen una serie de elementos asociados al concepto de Seguridad y luego mencionaremos las directrices definidas por la Norma Chilena de Seguridad.

2.1.1. Elementos de Seguridad

Definición de Seguridad

Una definición de Seguridad es proporcionada por el National Institute of Standards and Technology (NIST), organismo del Departamento de Comercio de Estados Unidos, en su Handbook o manual de Seguridad, publicado en 1995:

> "Computer Security: The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunications)." [1]

A partir de esta definición, según indican los autores Stallings y Brown, en su libro de Seguridad de Computadores [2] surgen tres objetivos claves de seguridad, definidos como la "tríada CIA", que corresponden a:

- ▷ Confidencialidad
- ▷ Integridad
- ▷ Disponibilidad (Availability)

Adicionalmente se agregan los objetivos:

- ▷ Autenticidad
- ▷ Accountability : Registro de Actividad, que se traduce en monitoreo a través de herramientas de Audit, Logging, Tracing

[1] Guttman & Roback [7, capítulo 1.4]
[2] Stallings & Brown [17, capítulo 1]

Taxonomía de Seguridad

Según señalan los autores Schumacher y otros en su libro de Patrones de Seguridad [3] , una "vista" de los elementos de seguridad en una organización debe considerar (tabla 2.1):

Aspecto		Descripción
Estrategias y Políticas de Seguridad	Propiedades • Confidencialidad • Integridad • Disponibilidad • Accountability	Incidentes • Acceso no autorizado • Usurpación de Identidad • etc Manejo de Riesgos • Evaluación y clasificación • Determinación Vulnerabilidades • Planes Enfoques • Prevención • Detección • Mitigación
Servicios	Servicios de Seguridad	• Identificación y Autenticación • Control de Acceso • Accounting • Non-Repudiation • System Recovery
	Servicios de Soporte de Seguridad	• Autorización • Mantención • Operación
Mecanismos e Implementación	Automáticos	• Encriptación • Scanners • Firewalls, Proxies, Sniffers • Logon/off (User Id y Password) • etc
	Físicos	• Guardia humano, llaves, sensores
	Procedimientos	• Operación, Respaldos, etc.
	Mecanismos de Soporte a la Administración	• Entrenamiento • Documentación • Disaster Recovery

Tabla 2.1: Taxonomía de Seguridad

[3] Schumacher *et al.* [16, capítulo 2]

Modelos de Seguridad y la Gestión de Usuarios

Como hemos visto, la seguridad tiene que ver con una serie de elementos y funciones. De este modo, entre algunos elementos principales identificamos a los Usuarios, los Recursos y los servicios de seguridad asociados. Estos servicios son la Identificación/Autenticación de Usuarios, el Control de Acceso o Autorización, y el Registro o Accounting de los accesos a los diferentes Recursos de la Organización.

Y estos son precisamente los elementos que constituyen los conceptos con los que debe lidiar el sistema Gestión de Usuarios.

Adicionalmente identificamos atributos de calidad de Confidencialidad, Integridad y Disponibilidad.

Estos conceptos nos permiten desarrollar un *Modelo* que sea coherente con los objetivos de seguridad y que resulte apropiado a los intereses de la organización. En consecuencia, en conjunto con los atributos de calidad, los elementos que debe considerar el Sistema Gestión de Usuarios son:

- Administración
  Registro, creación, eliminación y modificación de los diferentes elementos de seguridad que componen la plataforma.
- Autenticación
  Servicios de identificación y autenticación de los usuarios de la plataforma.
- Autorización
  Servicios de autorización de acceso a los recursos de la plataforma.
- Accounting (Auditoría, Logging, Tracing)
  Registro de las actividades de seguridad efectuadas sobre la plataforma (conexión, identificación, autorización, incidentes de seguridad, etc).

Patrones de Seguridad

A la hora de modelar y definir un diseño de un sistema, resulta de gran utilidad verificar la existencia de Patrones que permitan la reutilización de conocimiento y la implementación de soluciones probadas y conocidas.

El área de seguridad efectivamente presenta Patrones recurrentes en los elementos que hemos enunciado. En los párrafos que siguen, y basándonos en el trabajo de Schumacher y otros [4], delinearemos brevemente aquellos Patrones que resultarán de utilidad para el desarrollo del sistema Gestión de Usuarios.

[4] Schumacher *et al.* [16, capítulos 7,8,9,11]

2.1.2. Patrones de Identificación y Autenticación (I&A)

Existen varios patrones asociados a la Identificación y Autenticación de los usuarios. En general, cuando hablamos de usuarios estamos refiriéndonos a usuarios individuales, procesos o cualquier entidad que sea sujeto potencial de accesar y utilizar un determinado Recurso de la Organización.

Aunque están íntimamente relacionados, la Identificación y la Autenticación son conceptualmente dos procesos diferentes: La Identificación se refiere al proceso de requerir y obtener (mediante algún mecanismo) la Identidad de un usuario o actor que desea interactuar con la plataforma sistémica. Por su parte, la Autenticación corresponde al proceso de verificar y validar que la Identidad obtenida corresponde efectivamente a dicho usuario o actor, en otras palabras, que el usuario/actor es efectivamente quien dice ser.

La I&A tiene una serie de usos:

- ▷ Mecanismo de log-on para sistemas
- ▷ Activar otros servicios de seguridad: Autorización, Accounting
- ▷ Funcionales: Por ejemplo, asignación de transacciones a un usuario correctamente identificado (en un sistema de compras, un sistema de cuentas corrientes bancarias, etc.)

Veamos ahora los patrones de I&A:

I&A Requirements

- ▷ Función
  Definir los elementos y mecanismos a utilizar en los procesos de I&A. En un escenario de múltiples requerimientos, este patrón busca definir un procedimiento que permita realizar un diseño que satisfaga de la mejor forma los requerimientos de I&A.
- ▷ Los requerimientos pueden ser múltiples e incluso entrar en conflicto entre sí. El patrón define los siguientes pasos:
  - Definir los dominios de utilización de la I&A. Esto está relacionado con las aplicaciones que requieren I&A. Por ejemplo, los sistemas internos pueden requerir un mecanismo de user-password, las conexiones con plataformas externas el uso de Certificados de Seguridad, una aplicación de Compra de Bonos puede necesitar autenticación con mecanismos biométricos, etc.
  - Definición de requerimientos genéricos
  - Definición de requerimientos específicos
  - Priorización de requerimientos y explicitar los posibles trade-off
  - Determinación de costos y beneficios (frecuencia de uso, costos de implementación, impactos de obtener falsos positivos y falsos negativos)

Automated I&A Design Alternatives

- ▷ aka [5] Decision Tradeoffs for automated I&A
- ▷ Función
  Este patrón se complementa con el anterior, en cuanto a que busca un procedimiento para realizar la elección más adecuada del o los mecanismos de I&A a utilizar.
- ▷ Pasos a seguir:
  - Matriz de costos/beneficios y priorización (dominios de utilización, análisis de costos, frecuencia de uso, impacto de errores, etc)
  - Definición de estrategia
  - Análisis y selección de alternativas (o combinación de alternativas)
  - Especificaciones detalladas
- ▷ Técnicas frecuentes de I&A
  - User ID/Password
  - Biométrica
  - PKI (Public Key Infraestructure)
  - Hardware Token
  - Usuarios No registrados (usuarios sólo se identifica pero no se efectúa autenticación. Por ejemplo, cuando sólo entrega un mail)

Password Design and Use

- ▷ Función
  Este patrón aplica cuando se utilizan técnicas de user ID/password y describe las mejores prácticas y consideraciones para el diseño y uso de passwords en una organización. El objetivo es evitar y/o minimizar el robo de passwords, la suplantación, etc., así como definir mecanismos para distribución, recuperación de passwords olvidadas y otros.
- ▷ Algunos elementos a considerar cuando se utiliza el mecanismo de passwords
  - Diseño de passwords
    - Composición de caracteres válidos
    - Largo mínimo y máximo
    - Caracteres obligatorios (por ej, debe contener números y letras, mayúsculas, etc)
  - Uso
    - *Lifetime* o duración máxima de la password
    - Usos (por ejemplo, si se aceptará más de una conexión con una password determinada)
    - Mecanismo de ingreso de la password (digitación, cantidad de intentos, etc)
    - Períodos entre autenticaciones: El máximo tiempo aceptable entre una autenticación y una subsecuente autenticación, dentro de una sesión

[5] alias

- Protección de passwords
  - Mecanismo de distribución de passwords (por ejemplo, mail, links, sobres físicos, etc)
  - Almacenamiento: Métodos válidos para almacenar passwords (ofuscamiento, encriptación, segmentación y hashing, etc)
  - Transmisión: Métodos válidos para comunicar una password desde su punto de ingreso hacia el punto de validación

2.1.3. Patrones de Autorización (Control de Acceso)

En lo que sigue describiremos varios patrones asociados a la Autorización o Control de Acceso a los Recursos. Como ya se indicó, cuando hablamos de usuarios estamos refiriéndonos a usuarios individuales, procesos o cualquier entidad que sea sujeto potencial de accesar y utilizar un determinado Recurso de la Organización.

Autorization Pattern

- aka DAC (Discretionary Access Control)
- Elementos: usuarios y recursos
- Los permisos son asociados entre usuarios individuales y recursos, indicando directamente para cada usuario cuáles son los recursos que puede accesar y el tipo de acción que puede realizar

RBAC (Role-Based Access Control)

- Elementos: usuarios y recursos
- Los permisos son asociados entre las funciones o tareas (Roles) que los individuos deben realizar y los recursos. Los usuarios tienen roles asociados (uno o más) y heredan los permisos propios de cada rol.

Multilevel Security

- aka MAC (Mandatory Access Control)
- Elementos: usuarios, recursos y niveles de seguridad
- Muy utilizado en organizaciones altamente jerarquizadas
- Los recursos se clasifican de acuerdo a su nivel de criticidad y sensibilidad para la organización (por ejemplo, documento o recurso público, interno, confidencial, top secret, etc).
- Los usuarios se agrupan de acuerdo a su rango/jerarquía y funciones en la organización

- ▷ Los usuarios tienen acceso a los recursos de acuerdo a su jerarquía y según reglas definidas por los niveles de seguridad

Role Rights Definition

- ▷ Objetivo: Asociado al RBAC. Mecanismo para definir los permisos asociados a cada Rol.
- ▷ Se basa en determinar roles y permisos en base al análisis de los Casos de Uso
- ▷ Un mecanismo de implementación es a través de una *Matriz de Permisos (Roles vs Recursos)*

2.1.4. Patrones Arquitecturales de Autorización

A continuación describiremos algunos patrones asociados a la Arquitectura para Autorización o Control de Acceso a los Recursos..

Single Access Point

- ▷ aka One Way In, Login Window, Guard Door
- ▷ Objetivo: Proveer a clientes externos de un punto de acceso hacia el sistema, evitando el mal uso o intromisión de clientes no deseados
- ▷ Se basa en establecer un punto único para la conexión o Login hacia las aplicaciones y recursos

Check Point

- ▷ aka Policy Definition Point (PDP), Access Verification, Pluggable Authentication
- ▷ Objetivo: Proveer a clientes externos de un punto único de control de autenticación, para habilitar el acceso hacia el sistema, y evitar accesos no deseados.
- ▷ Se basa en definir un punto único de verificación de I&A y autorización. Puede estar asociado con el patrón Single Access Point.

Reference Monitor

- ▷ aka Policy Enforcement Point (PEP)
- ▷ Objetivo: Asegurar se de cumplimiento a las restricciones de autorización.
- ▷ Proceso abstracto que *intercepta* todos los requerimientos de accesos a los recursos y chequea si procede o no la autorización

Security Session

- ▷ aka Session, Namespace
- ▷ Objetivo: El sistema está conformado por múltiples componentes que necesitan compartir información y datos de seguridad asociados a los usuarios.
- ▷ Establece un mecanismo para registrar atributos de seguridad asociados a los usuarios. Con ello se evita repetir la autenticación del usuario. Además, puede usarse o no en conjunto con los permisos (atributos de autorización).

2.1.5. Patrones de Accounting

En lo que sigue se describen varios patrones vinculados con el Registro de Actividades de Seguridad, asociadas con los accesos a los Recursos de la Organización. Estos accesos se originan en las solicitudes generadas por las entidades (usuarios, sistemas, aplicaciones, etc) y, dependiendo del contexto y objetivos, se deben considerar tanto aquellas que resultaron exitosas como las que no prosperaron. Cabe señalar que para implementar este Registro de Actividad se utilizan normalmente mecanismos de Audit, Logging y Tracing, localizados en diferentes puntos de la plataforma sistémica.

La utilidad del Accounting es diversa, por ejemplo:

- ▷ Determinación y ajustes de niveles de carga del sistema de Seguridad
- ▷ Niveles de uso de la plataforma sistémica (c/r al sistema de Seguridad)
- ▷ Ajustes de horarios peak y valle
- ▷ Incidentes de seguridad
- ▷ Generación de reportes
- ▷ Auditorías
- ▷ Registro histórico

Además, un registro de Accounting debe ser capaz de responder:

- ▷ Qué se hizo
- ▷ Quién lo hizo
- ▷ Cuándo
- ▷ Cómo (mecanismo utilizado)
- ▷ Dónde (interno, externo, web, direcciones IP)
- ▷ Por/para qué (obedece a procedimientos habituales o a excepciones)

Similarmente, el diseño debe definir:

- ▷ Qué registrar y cuál o cuáles elementos de la plataforma serán los encargados de este registro
- ▷ Cómo y cuándo registrar (qué datos, formatos, qué información)

- ▷ Dónde (medios de almacenamiento)
- ▷ Por/para qué, es decir, apuntando a los objetivos para los cuales se requiere dicho registro

En definitiva, los patrones de interés son los siguientes:

Security Accounting Requirements

- ▷ Función: Registrar acciones, eventos e incidentes relacionados con la seguridad (por ejemplo, accesos no autorizados a un recurso o una base de datos, transmisión de virus, etc.) Aunque cada evento e incidente de seguridad es único, este patrón busca definir un procedimiento que permita definir qué registrar, cuándo hacerlo, etc, de modo de satisfacer de la mejor forma posible los requerimientos de registro de eventos de seguridad de la organización.
- ▷ Los requerimientos de registro pueden ser múltiples e incluso entrar en conflicto entre sí. El patrón define los siguientes pasos:
  - Definición de requerimientos genéricos
  - Definición de requerimientos específicos
  - Priorización de los requerimientos y explicitar los posibles trade-off
  - Especificaciones detalladas

Análogamente, existen patrones para levantar y definir requerimientos de accounting más específicos. A saber:

- ▷ Audit Requirements
- ▷ Audit Trails and Logging Requirements
- ▷ Intrusion Detection Requirements
- ▷ Non-Repudiation Requirements

2.2 Normativa Chilena de Seguridad

La Norma Chilena de Seguridad es utilizada frecuentemente como exigencia de certificación para las empresas desarrolladoras de software, así como las proveedoras de servicios. Estas certificaciones se refieren tanto a la calidad como al cumplimiento de estándares con que opera la organización.

El Estado de Chile ha incorporado esta norma como exigencia para varias de sus reparticiones y ministerios, y es parte de un programa global de mejoramiento que busca que todos los servicios públicos operen bajo dicha especificación [6].

También es requerida en los diferentes procesos de certificación que las organizaciones desean realizar (por ejemplo, ISO).

Existen muchas certificaciones, que se diferencian por su alcance, los aspectos que cubre, las industrias que abarcan, etc. En términos generales, las certificaciones son exigidas por diferentes empresas u organismos como prerrequisito para operar con la organización certificada. Además una certificación habitualmente es evidencia de calidad, estandarización y nivel de madurez operacional exhibido por dicha organización. Por último, señalar que estas certificaciones son otorgadas por diferentes entidades (empresas, organizaciones, consultoras, etc), tanto a nivel nacional como internacional.

En particular, la normativa chilena permite establecer un piso mínimo de exigencias, recomendaciones y buenas prácticas que toda organización debe seguir, si desea operar en condiciones adecuadas de seguridad.

2.2.1. La Normativa

La normativa chilena de seguridad está descrita en la Norma ISO/IEC 27002:2005, la que a su vez se encuentra incorporada en la Norma Chilena de Seguridad NCH-ISO 27001:2009, referente a buenas prácticas en seguridad de la información. Esta norma define un conjunto de exigencias, recomendaciones y buenas prácticas, que se describen en lo que sigue:

En lo esencial, la Normativa indica que se debe establecer un conjunto documentado y verificable de procedimientos y sistemas, que permita cumplir con:

- ▷ Acceso garantizado: Garantizar la disponibilidad del acceso a los sistema y servicios
- ▷ Acceso confiable: Veracidad de la información del sistema
- ▷ Acceso exclusivo: Para los usuarios autorizados, de acuerdo a los perfiles y funciones que sean asignados.

[6] PMG SSI, Programa de Mejoramiento de la Gestión, Ley N°19.553 de 1998. A partir de 2010 incluye el SSI - Sistema de Seguridad de Información.

La Norma establece *Objetivos de Control*. Un Objetivo de Control es un área o aspecto de seguridad, que:

- ▷ Tiene un objetivo de seguridad definido
- ▷ Tiene controles asociados
- ▷ Se encuentra documentado

Los Objetivos de Control (en términos de su descripción y controles) que deben ser considerados son:

1. Política de seguridad

Proporcionar una dirección y apoyo de la dirección de la empresa/organización (niveles gerenciales) a la seguridad de la información de acuerdo a los objetivos de negocios, leyes y regulaciones relevantes.

- ▷ Documento de política de seguridad de información.
- ▷ Revisiones periódicas de la política de seguridad de información

2. Aspectos organizativos para la seguridad

Para administrar la seguridad de información dentro de la organización.

- ▷ Participación de la Gerencia en Seguridad de Información.
- ▷ Coordinación de seguridad de información.
- ▷ Asignación de responsabilidades de seguridad de información
- ▷ Proceso de autorización para la instalaciones de procesamiento de información
- ▷ Acuerdos de confidencialidad
- ▷ Contacto con autoridades
- ▷ Revisiones independientes de seguridad de información
- ▷ Identificación de riesgos relacionados con partes externas
- ▷ Directriz de seguridad en el trato con clientes
- ▷ Directriz de seguridad en acuerdos con terceras partes

3. Gestión de activos

Para asegurar y mantener la apropiada protección de los activos de la organización y del servicio

- ▷ Inventario de activos
- ▷ Propiedad de activos
- ▷ Uso aceptable de activos
- ▷ Guías para clasificación de la información
  - – Debe existir una definición de los tipos de información, en particular que identifique la que es confidencial y debe ser protegida
  - – Debe existir un manual/procedimiento que detalle cómo se maneja la información de acuerdo a su clasificación, en particular la clasificada como confidencial.
- ▷ Manejo y etiquetado de información
  - – Debe existir un manual/procedimiento que detalle cómo se etiqueta la información de acuerdo a su clasificación, en particular la identificada como confidencial.

4. Seguridad ligada a los recursos humanos

Para asegurar que los empleados, contratistas y usuarios de terceras partes comprenden sus responsabilidades, que éstas son acordes con sus roles y reducir el riesgo de hurto, fraude o uso malicioso de instalaciones.

- ▷ Previo al empleo
  - – Identificación de roles y responsabilidades
  - – Investigación de antecedentes
  - – Términos y condiciones del empleo
- ▷ Durante el empleo
  - – Responsabilidades administrativas
  - – Capacitación, educación y entrenamiento en seguridad de información.
  - – Proceso disciplinario por no cumplimiento de las políticas
- ▷ Al término o cambio de empleo
  - – Responsabilidades de término de contrato
  - – Reintegro de activos
  - – Remoción de derechos de acceso

5. Seguridad física y del entorno

Áreas seguras; seguridad de los equipos. Para prevenir acceso físico no autorizado, daño e interferencia a las definiciones de la organización y de la información.

- ▷ Perímetro de seguridad física
- ▷ Controles de ingreso físico
- ▷ Asegurando oficinas, salas e instalaciones
- ▷ Protección contra amenazas externas y ambientales
- ▷ Áreas de acceso público, carga y despacho.
- ▷ Ubicación y protección de equipos
- ▷ Seguridad del cableado
- ▷ Mantenimiento de equipos
- ▷ Remoción de propiedad (ante cese de uso, *wipe* o borrado seguro de la información)

6. Gestión de comunicaciones y operaciones

Para asegurar la correcta y segura operación de las instalaciones de procesamiento de información.

- ▷ Procedimientos y responsabilidades operativas
- ▷ Administración de servicios entregados por terceros
- ▷ Aceptación y planificación de sistemas
- ▷ Protección contra código malicioso
- ▷ Recuperación y respaldos
- ▷ Administración de seguridad de redes
- ▷ Manejo de medios
- ▷ Intercambio de información (cifrado de información)
- ▷ Transacciones en línea (cifrado de información)
- ▷ Monitoreo

7. Control de accesos

Para controlar el acceso a la información

- ▷ Requerimientos de negocios para control de acceso
  - El acceso a las aplicaciones debe requerir de autenticación y autorización mediante la administración de perfiles y usuarios con claves
  - Las claves de acceso a los sistemas deben tener como largo mínimo 6, contener letras, números y símbolos
  - El intercambio de transacciones/documentos debe considerar firma electrónica, con sistema criptográfico asimétrico.
  - El acceso a las aplicaciones debe requerir de autenticación y autorización mediante la administración de perfiles y usuarios con claves y un dispositivo de autenticación único tipo token o similar variable en el tiempo.

- ▷ Administración de accesos de usuarios
  - Debe existir una administración de usuarios y claves
  - Debe existir un log asociado al proceso de autenticación y la administración de usuarios que contenga el registro de al menos los últimos 6 meses
- ▷ Responsabilidades de usuarios
- ▷ Control de acceso a redes
- ▷ Control de acceso al sistema operativo
- ▷ Control de acceso a aplicaciones e información

8. Adquisición, desarrollo y mantenimiento de sistemas de información

Garantizar que la seguridad es una parte integral de los sistemas de información.

- ▷ Requerimientos de seguridad de sistemas de información
  - Las aplicaciones y/o sistemas deben funcionar publicando servicios del tipo web con cifrado (https), mediante intercambio de transacciones en línea via WebServices SOAP WS-I BP 1.0 y con FTP seguro para el intercambio de archivos con datos (para efectos de conciliación de datos).
  - Sólo se deben publicar los servicios requeridos para el funcionamiento del sistema.
  - Debe existir un proceso de autenticación y autorización de los usuarios/sistemas que utilizan las aplicaciones
  - Debe existir un sistema de administración de usuarios con diferentes roles
  - Debe existir registro de las transacciones realizadas por al menos 6 meses
- ▷ Procesamiento correcto en aplicaciones
- ▷ Encriptación
  - El intercambio de información debe ser cifrado. En el servicio habilitado debe ser cifrado (SSL, SFTP)
- ▷ Seguridad en los sistemas de archivos
  - Deben estar definidos y aplicados distintos perfiles de acceso a los sistemas de archivos, diferenciando al menos permiso de lectura, escritura y ejecución
- ▷ Seguridad en los procesos de desarrollo y soporte
  - El ambiente de desarrollo debe ser separado del ambiente de producción
- ▷ Administración de vulnerabilidades técnicas
  - Se deben mantener las actualizaciones de seguridad declarada por los proveedores al día
  - Se deben realizar evaluaciones de vulnerabilidades de los servicios/servidores publicados al menos una vez al año.

9. Gestión de incidentes de seguridad

Para asegurar que los eventos de seguridad de información y las vulnerabilidades asociadas son comunicadas en forma que permita tomar acciones correctivas oportunas.

- ▷ Reportar vulnerabilidades y eventos de seguridad
- ▷ Administración y mejora de incidentes de seguridad de información

10. Gestión de continuidad del negocio

Para evitar la interrupción de actividades de negocios y para proteger los procesos críticos de negocios de los efectos producidos por una falla mayor de sistemas de información o un desastre, y asegurar su reconstitución oportuna.

- ▷ Assessment (Evaluación) del riesgo y continuidad de Negocios
- ▷ Diseño e implantación de planes de continuidad que incluyan seguridad de información
- ▷ Marco de trabajo para planificación de continuidad de negocios
- ▷ Pruebas, mantenimiento y reevaluación de planes de continuidad de negocios
- ▷ Arquitectura de alta disponibilidad (Activo-Activo)

11. Conformidad o Cumplimiento

Con respecto de cualquier ley, estatuto, regulación u obligación contractual y cualquier requerimiento de seguridad.

- ▷ Conformidad con los requerimientos legales
- ▷ Conformidad con política y estándares de seguridad, y conformidad técnica
- ▷ Consideraciones de auditoria de sistemas

2.2.2. La Normativa Chilena y el Sistema Gestión de Usuarios

¿Cuál es la relación entre la Normativa Chilena de Seguridad y la Gestión de Usuarios?

Como se observa, la Normativa Chilena enumera una serie de aspectos que deben ser tomados en cuenta para garantizar la seguridad al interior de la organización. Además, resulta claro que el sistema Gestión de Usuarios impacta directamente en varios (todos, en mayor o menor medida) de los aspectos señalados.

Esto puede verse en la tabla 2.2 :

	Objetivo de Control	Relación con Gestión de Usuarios
1	**Política de seguridad**	Implantar la política de seguridad, de acuerdo a los requerimientos del negocio
2	**Aspectos organizativos para la seguridad**	Protección de la información
3	**Gestión de activos**	Etiquetar las funciones y accesos a la información, de acuerdo a su nivel de sensibilidad y confidencialidad.
4	**Seguridad ligada a los recursos humanos**	Implementar flujos para acciones de alta, permanencia y baja de funcionarios
5	**Seguridad física y del entorno**	• Remoción o *wipe* de información ante cese de uso • Eliminación de cuentas
6	**Gestión de comunicaciones y operaciones**	• Definición de roles internos y de terceros • Cifrado en el intercambio de información
7	**Control de accesos**	Triple A. Autenticación, Autorización, Accounting.
8	**Adquisición, desarrollo y mantenimiento de sistemas de información**	Encriptación, roles y permisos, accounting
9	**Gestión de incidentes de seguridad**	Accounting
10	**Gestión de continuidad del negocio**	• Accounting. • Alta disponibilidad
11	**Conformidad o Cumplimiento**	Cumpimiento de requisitos del negocio y accounting.

Tabla 2.2: Relación Objetivos de Control y la Gestión de Usuarios

2.3 La Seguridad y el sistema Gestión de Usuarios

En los párrafos precedentes hemos revisado varios elementos asociados a la seguridad de una organización.

▷ Patrones de seguridad

▷ La Normativa Chilena de Seguridad

2.3.1. Elementos de Seguridad en un modelo de Gestión de Usuarios

¿Cuáles son los elementos a considerar en el modelamiento de un sistema de Gestión de Usuarios?

Según se desprende de los conceptos analizados (Seguridad, Norma Chilena de Seguridad), estos elementos son:

- Administración
- Autenticación
- Autorización
- Accounting (Auditoría. Logging, Tracing)

2.4 LDAP

Como hemos visto, un sistema de gestión de usuarios es una componente activa en la seguridad de la organización. A la vez, debe cumplir con una serie de atributos de calidad, asociados a altos niveles de rendimiento, disponibilidad y especialmente una alta capacidad de interacción con diversos sistemas y plataformas. Esto último - idealmente - a través de mecanismos estándares y de rápida implementación.

En lo que sigue se describirá en detalle la tecnología LDAP, y veremos que, en atención a sus características, resulta extraordinariamente apropiada para los fines perseguidos en el desarrollo del sistema Gestión de Usuarios.

2.4.1. Definiciones

Directory
: Un directorio es simplemente un conglomerado o conjunto de información. Por ejemplo, un conjunto de URLs, una lista de usuarios, una guía telefónica, etc.
Adicionalmente, un directorio nos provee de algún mecanismo para almacenar dicha información. Este mecanismo puede ser una base de datos, un archivo o algún otro medio. En cualquier circunstancia, el medio utilizado debe ser transparente al usuario de la información

Directory Service
: Un servicio de directorio es un mecanismo que provee acceso a la información contenida en el directorio. Estos servicios, con múltiples opciones y variantes, corresponden a búsqueda de información, así como mecanismos de actualización.
Es habitual que los directorios sean usados intensamente en búsquedas y en menor

medida en actualizaciones. Por ello, históricamente se consigna que estos servicios están especialmente optimizados en búsqueda y navegación, en desmedro de sus rendimientos en modificación. No obstante, cabe señalar que, hoy por hoy, las implementaciones de directorios son cada vez más eficientes en ambos tipos de servicios.

Directory Server
Un servidor de directorio es una aplicación cuya función es proveer servicios de directorio a otras aplicaciones y usuarios finales.

Client
Una usuario final, aplicación u otro ente que requiera utilizar los servicios de directorio

Protocol
Un protocolo es un conjunto de normas y especificaciones, sin ambigüedades y debidamente documentado.

2.4.2. La tecnología LDAP

Las iniciales LDAP se refieren al término, en inglés, *Lightweight Directory Access Protocol*, y significa Protocolo Liviano de Acceso a Directorio.

LDAP se basa en el estándar X.500, el que corresponde a un conjunto de especificaciones de redes de computadoras y servicios de directorio, definidas por la ITU-T. Este último es el organismo internacional dedicado a definir normas y especificaciones a nivel mundial, en el área de las Telecomunicaciones.

El protocolo LDAP nació como una versión "liviana" del X.500, y su uso en las organizaciones está cada vez más difundido.

En términos breves, un Lightweight Directory Access Protocol (LDAP) es un conjunto de protocolos estándar para el acceso a servicios de directorio, que tienen como objetivo permitir la recuperación de diversos tipos de información en un entorno de red.

Dependiendo del contexto, cuando se habla de LDAP, nos referimos a [7] :

- ▷ Un protocolo liviano, estándar y extensible para utilizar servicios de directorio
- ▷ Un conjunto de cuatro modelos que guían el uso del directorio, a saber: Un modelo de información que indica qué poner en el directorio, un modelo de nomenclatura que indica cómo disponer y referir los datos en el directorio, un modelo funcional que describe qué se puede hacer con dichos datos y un modelo de seguridad que describe cómo proteger los datos de accesos no autorizados
- ▷ El LDIF (LDAP Data Interchange Format) que corresponde a un formato estándar de intercambio de información de los datos
- ▷ Un conjunto de comandos y utilitarios asociados al servidor y aplicaciones basadas en LDAP

[7] Howes *et al.* [8, capítulo 2]

- ▷ Las LDAP APIs (Application Programming Interfaces), usadas para desarrollar aplicaciones clientes LDAP

2.4.3. Ejemplos de LDAP

Algunas implementaciones de LDAP:

- ▷ Microsoft Active Directory [8]
- ▷ IBM Tivoli Directory Server [9]
- ▷ OpenLDAP [10]

2.4.4. Componentes de un LDAP

- ▶ LDAP Directory

  El directorio propiamente tal, es decir, el repositorio de datos o *backing store* donde se almacena la información.

- ▶ LDAP Services

  Los servicios de administración, configuración, actualización y consulta de información. Estos servicios pueden ser proporcionados y accedidos en diferentes modalidades: A través de linea comandos, invocando el LDAP server, mediante el uso de APIs, etc.

- ▶ LDAP Server

  El servidor que provee servicios de directorio, utilizando el protocolo LDAP

2.4.5. Organización de la información en LDAP

En líneas generales, la información en un directorio LDAP se organiza [11] como un modelo jerárquico de objetos, cada uno de los cuales se denomina *Entry* (entrada en el directorio). La estructura jerárquica o árbol se denomina *Directory Information Tree* (DIT). Cada entry está compuesta o es una instancia de uno o más *ObjectClass*. Los ObjectClass se componen, a su vez, de una colección de *atributos*.

Además el LDAP provee mecanismos para actualizar y recuperar la información asociada al modelo definido.

[8] Microsoft [12, web]
[9] IBM-TDS [11, web]
[10] OpenLDAP [14, web]
[11] Carter [4, capítulo 1.3]

El concepto fundamental es que la información de la empresa/organización debe ser modelada de acuerdo a este esquema, lo que permite entonces sacar el máximo provecho de las facilidades ofrecidas por el LDAP.

2.4.6. Características de un LDAP

Algunos atributos o características relevantes de un LDAP:

- ▷ Diseñado para manejar un alto volumen de transacciones de consulta
- ▷ Bajo volumen de actualizaciones/modificaciones
- ▷ Los datos del directorio pueden estar distribuidos en diferentes servidores (los datos pueden estar segmentados por alguna característica y distribuidos de acuerdo a dicha característica)
- ▷ Los datos del directorio frecuentemente están replicados en más de un servidor, lo cual contribuye al atributo de alta disponibilidad
- ▷ Proporciona diferentes elementos de privacidad y encriptación
- ▷ Altos niveles de rendimiento
- ▷ Debe cumplir estrictamente con los estándares LDAP, lo que promueve la compatibilidad

2.4.7. ¿Cuándo aplicar un LDAP?

Un LDAP permite consolidar información que cumple con ciertas características:

- ▷ transversal a toda una organización
- ▷ disponible en todo instante
- ▷ con niveles de seguridad bien definidos
- ▷ alto volumen de tráfico
- ▷ bajo nivel de modificaciones
- ▷ requerimientos directos o a través de otros sistemas
- ▷ capacidad de interacción con múltiples y diversas plataformas

¿Cuándo NO es apropiado utilizar un LDAP?

No es sencillo responder esta pregunta. Pero una clara señal es:

- ▷ No usarlo si existe un alto volumen de modificación de los datos
- ▷ Un directorio LDAP almacena información útil para múltiples plataformas y sistemas. Aún cuando puede contener variada información de usuarios, recursos, configuraciones, etc, en diferentes tipos de formatos, o incluso almacenar imágenes (JPEG), en ningún caso está pensado para almacenar data tipo BLOB (Binary Large Object)

2.4.8. Historia de LDAP

A continuación una breve reseña histórica del protocolo LDAP.

1998 - Se establece el estándar X.500

1993 - Buscando un alternativa como versión más "liviana" del protocolo, se llevan a cabo diversas iniciativas que convergen finalmente en la RFC 1487 , en la que se establece el protocolo LDAP en su primera versión

1997 - LDAP es un protocolo que actualmente se encuentra en la versión LDAPv3 y fue definido originalmente en las RFC 2251, 2252, 2253, 2254, 2255, 2256, 2829, 2830 y 3377 [12].

2002 - RFC 3377. Esta última corresponde a la especificación técnica del LDAPv3.

2006 - Se reformulan *todas* las RFCs del protocolo LDAPv3 (RFCs 4510 a 4519).

Paralelamente surge la iniciativa OpenLDAP Foundation, proyecto open source (libre y de código abierto), que comienza sus actividades en agosto de 1998, y corresponde a una comunidad activa, cuyo objetivo declarado es desarrollar una suite robusta, de alto nivel, full.compliance, de aplicaciones y herramientas LDAP. La versión actual OpenLDAP 2 cumple con la especificación LDAPv3.

Además varios de los principales desarrolladores de software a nivel mundial han construido sus propias suites, de carácter comercial.

Si bien OpenLDAP es el proyecto open source más conocido, no es el único. También están los interesantes proyectos Apache Directory [13] (2002) del grupo Apache Software Foundation y OpenDS [14] de Sun/Oracle (2011).

En cualquier caso, el uso de herramientas LDAP, tanto open como comerciales, es cada vez más extendido.

De este modo, a medida que la comunidad empresarial ha ido adhiriendo al uso del protocolo, se ha producido una integración creciente de herramientas y productos de software al estándar LDAP.

- ▷ Integración vía APIs o librerías de diversos lenguajes de programación para comunicación vía protocolo LDAP (Java, Python)
- ▷ Frameworks de desarrollo de aplicaciones han implementado extensiones para comunicación vía LDAP (por ejemplo, Spring, Django)
- ▷ Servidores de aplicación han integrado APIs de comunicación siguiendo el protocolo LDAP (Tomcat usando las JNDI de Java)

Adicionalmente, aprovechando las características y ventajas del protocolo, se están explo-

[12] IBM-LDAP [10, capítulo 1 section 3.3]
[13] Apache [1, web]
[14] Oracle [15, web]

rando nuevas y diversas aplicaciones, en áreas tan relevantes como su uso, por ejemplo, para implementar DNS servers o la implementación de servicios DHCP basados en LDAP.

2.4.9. LDAPv3 versus las versiones anteriores de LDAP

La versión 3 de LDAP [15] fue desarrollada a fines de los 90. Su objetivo fue reemplazar a las versiones anteriores de LDAP, dando cobertura en aquellas áreas donde el protocolo aún estaba inmaduro.

LDAPv3 incorpora las siguientes características:

- ▷ Strong authentication and data security services via SASL
  Amplía los mecanismos de autenticación usando el mecanismo SASL (Simple Authentication and Security Layer)

- ▷ Certificate authentication and data security services via TLS (SSL)
  Encriptación de mensajería usando SSL/TLS.

- ▷ Internationalization a través del uso de Unicode
  Soporte de UTF-8 para caracteres.

- ▷ Referrals and Continuations
  Un servidor que no contiene un dato que está siendo requerido, puede referir o derivar el cliente hacia otro servidor (por ejemplo, en directorios que se encuentra particionados). Es una idea similar a la utilizada en los servidores DNS de la web.

- ▷ Schema Discovery
  Define mecanismos para recuperar el schema (definiciones de clases, tipos, etc) contenido en el directorio.

- ▷ Extensibility (object types, operaciones)
  Se puede ampliar el schema del directorio, definiendo nuevos object types y operaciones.

En la tabla 2.3 se muestra la lista de RFCs que han sido publicadas y que definían originalmente el núcleo de los estándares LDAP en su versión 3.

[15] OpenLDAP [13, capítulo 1.7]

Estándares LDAPv3 originales	
Protocol (RFC 2251)	
Mandatory Schema (RFC 2252)	User Schema (RFC 2256)
Distinguished Names (RFC 2253)	Authentication Methods (RFC 2829)
Search Filters (RFC 2254)	Transport Layer Security (RFC 2830)
LDAP URLs (RFC 2255)	Technical Specification (RFC 3377)

Tabla 2.3: Tabla de RFCs para LDAPv3 (originales)

Posteriormente, en 2006 aparecieron nuevas RFCs que han dejado obsoletas las anteriores. Tabla 2.4.

Nuevos Estándares LDAPv3 (2006)	
RFC 4510	Lightweight Directory Access Protocol: Technical Specification Road Map
RFC 4511	LDAP: The Protocol
RFC 4512	LDAP: Directory Information Models
RFC 4513	LDAP: Authentication Methods and Security Mechanisms
RFC 4514	LDAP: DN
RFC 4515	LDAP: Search Filters
RFC 4516	LDAP: URL
RFC 4517	LDAP: Syntaxes and Matching Rules
RFC 4518	LDAP: Internationalized String Preparation
RFC 4519	LDAP: Schema for User Applications

Tabla 2.4: Tabla de nuevas RFCs para LDAPv3 (2006)

2.4.10. Replicación y Control de Acceso

La versión 3 de LDAP implicó una significativa mejora sobre su predecesora. Sin embargo, no es explícita en temas de *replicación* de la data ni *control de acceso* sobre el directorio. En concreto, esto significa que cada proveedor ha implementado sus propios mecanismos para resolver estos aspectos.

Una práctica frecuente es la definición de *Access Control Lists* (ACLs) y *groups*, que limitan el acceso de los usuarios a ciertas acciones y sobre ciertos segmentos de la información

contenida en el directorio. Otros conceptos habituales son las Protected Object Policies (POP) que definen reglas de protección sobre determinados recursos, y que aplican a todos los usuarios del directorio.

En definitiva, LDAP sigue siendo un tema abierto. Y es probable que en el futuro los organismos que especifican los estándares vayan paulatinamente incorporando estas especificaciones de facto en nuevas RFCs de LDAP.

2.4.11. Otros servicios de Directorio

LDAP no es el único protocolo que provee servicios de directorio. Otros protocolos han sido populares en el pasado y muchos aún son utilizados en áreas específicas [16].

Se pueden mencionar el propio X.500 (que dio origen al protocolo LDAP), WHOIS, NIS, PH/QI, y algunos protocolos propietarios (de compañías como Novell, Banyan) y varios más.

2.4.12. Los cuatro Modelos de LDAP

Según hemos señalado, un LDAP se conforma por un conjunto de cuatro modelos, a saber:

Modelo de información que indica qué poner en el directorio

Namespace Un modelo de nomenclatura que indica cómo disponer y referir los datos en el directorio

Modelo funcional que describe qué se puede hacer con dichos datos

Modelo de seguridad que describe cómo proteger los datos de accesos no autorizados

En las siguientes secciones analizaremos cada uno de estos modelos.

2.5 LDAP - Modelo de Información

En lo que sigue veremos en detalle el modelo de datos de un directorio LDAP.

Según indica Donley en su libro LDAP [17], el modelo de datos de LDAP no es relacional, ni es tampoco completamente orientado a objetos. El modelo de datos de LDAP se puede describir de la siguiente forma:

[16] Donley [6, capítulo 1.1.3]
[17] Donley [6, capítulo 2.1]

- ▷ La información es representada como "objetos" denominados *Entry* (entrada en el directorio)
- ▷ Las entradas se componen de un conjunto de *atributos (Attribute)*
- ▷ Un atributo es un par *(Attribute Type, Attribute Value)*. El attribute type (tipo de atributo) es un identificador del tipo de información que contiene el atributo.
- ▷ Las características de un atributo quedan definidas por su Attribute Type
- ▷ Las entradas u objetos se identifican en forma única por su *distinguished name (DN)*.
- ▷ La organización de los objetos es jerárquica y está basada en sus DN. Esta organización se denomina *Directory Information Tree (DIT)*

Además,

- ▷ Las entradas pertenecen a uno o más *Object Class*
- ▷ Cada Object Class está definido en base a un conjunto de Attribute Types. Esta definición indica cuáles son los atributos que son obligatorios y cuáles son opcionales. Una entry pertenece a una Object Class si posee los atributos que la clase indica como obligatorios y posiblemente algunos de los opcionales.
- ▷ Las definiciones de Object Class y Attribute Types conforman un *Schema*
- ▷ Un Object Class puede extender una clase padre. Esto es, LDAP implementa el concepto de *herencia simple*. Esto conforma una jerarquía de clases, similar a la que encontramos en los lenguajes de programación. Por supuesto, esta es una jerarquía descriptiva o esquemática y no debe confundirse con el DIT, que corresponde a la jerarquía u organización de los objetos (entradas) en el directorio.

Podemos observar lo indicado en la figura 2.1. Aquí se muestra que un object class está compuesto de una lista de attribute types obligatorios (must) y una lista de attribute types opcionales (can).

Aunque no es exacto, si hacemos una analogía con los lenguajes de programación, los *object class* corresponden a las *clases* definidas en OOP, y las *entries* corresponde a los *objetos*. Además, la object class denominada *TOP* corresponde a la clase padre de todas las demás (similar a la clase *Object* de Java).

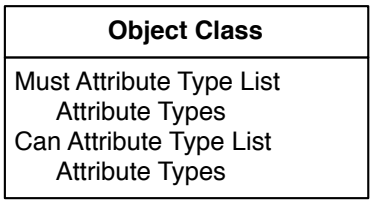


Figura 2.1: object Class

En la figura 2.2 se observa una jerarquía de Clases.

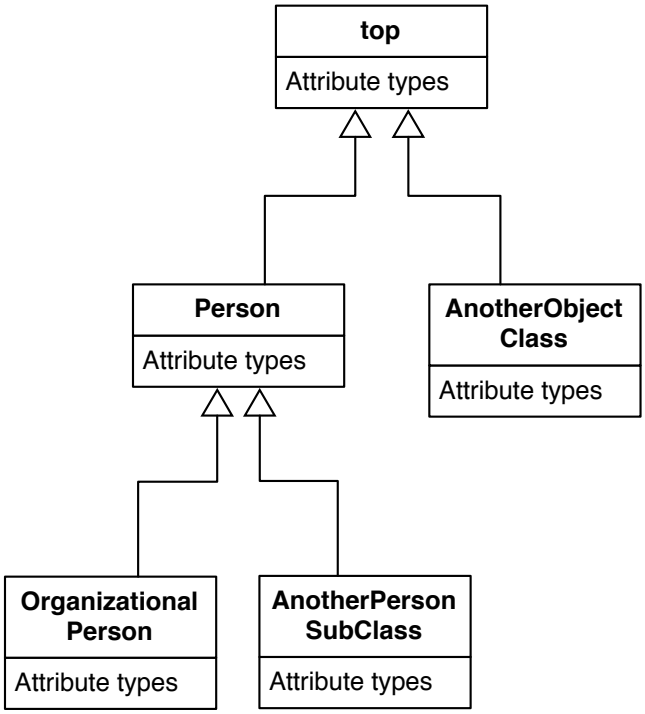


Figura 2.2: Jerarquía de Clases

En la figura 2.3 vemos un ejemplo de un conjunto de entries organizados en un DIT. Se indican, además, los nombres asociados a los objetos (el Namespace del Directorio).

El ejemplo es un posible DIT de representación de las universidades de nuestro país. Aparecen dos organizaciones, la Universidad Católica y la Universidad de Chile, ambas del país Chile. Las dos organizaciones tienen su nivel de detalle y descomposición en el DIT. En este caso sólo se muestra un detalle mayor y los elementos de una posible descomposición para la organización Universidad de Chile, llegando aquí hasta el nivel de los funcionarios que la integran.

2.5.1. Attributes y Entries

Las *Entries* están compuestas de pequeñas unidades denominadas *attributes*. Los atributos son similares a los campos en una Base de Datos, es decir, se conforman como un par *(attribute type, attribute value)*.

En la tabla 2.5 se observa una entry correspondiente a la objectclass *person*, y sus atributos asociados. Por ejemplo el atributo de tipo *sn (surname)* tiene el valor "Pérez".

Aparece también el atributo de tipo *objectclass*, cuyo valor es "person", y que sirve precisamente para indicar la objectclass a que pertenece la entry. Con ello se puede validar cuáles son los atributos que debe (y puede) tener la entry.

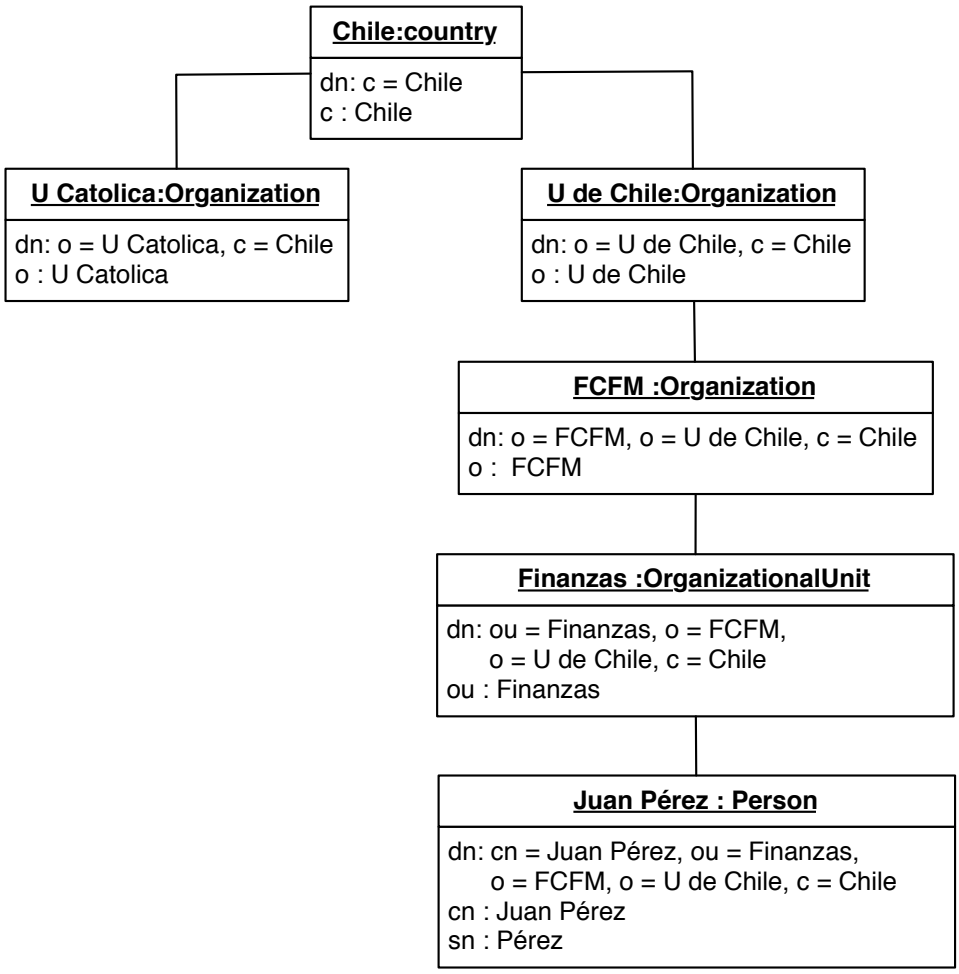


Figura 2.3: Ejemplo de DIT y Namespace

dn: cn=Juan Pérez, o=Universidad de Chile, c=Chile
objectclass: person
cn: Juan Pérez
sn: Pérez

Tabla 2.5: Ejemplo de Entry

En este ejemplo observamos:

- ▷ Un atributo es un par attribute type-attribute value.
- ▷ La entry tiene un atributo, el par objectclass: person, que indica a qué objectclass pertenece. Este atributo está compuesto del attribute type objectclass y del valor person.
- ▷ La entry se compone de cuatro atributos, cuyos atribute types son dn, objectclass, cn y sn
- ▷ Los atribute types son: dn por distinguished name, objectclass por su homónimo, cn por common name y sn por Surname [18]
- ▷ recordemos que el dn es un atributo que identifica a la entry en forma única
- ▷ en este caso el dn tiene por valor el string "cn=Juan Pérez, o=Universidad de Chile, c=Chile". Este valor corresponde a una secuencia de tres atributos, cn=Juan Pérez, o=Universidad de Chile y c=Chile, donde los atributte types son cn: Common Name, o:Organization y c:Country.
- ▷ El dn indica la ubicación de la entry en el árbol de información del directorio (DIT).
- ▷ En la entrada se identifican los atributos, que corresponden a los atributte types y sus respectivos valores. En este ejemplo, cn tiene el valor "Juan Pérez" y sn el valor "Pérez". Lo propio ocurre con los attribute types dn (ya explicado) y con objectclass.

Multivaluación de atributos

LDAP soporta atributos multivaluados. Por ejemplo, en la tabla 2.6 :

objectClass: person
cn: Juan Pérez
sn: Pérez
givenName: Juan
givenName: Juanito
givenName: Johnny

Tabla 2.6: Ejemplo de Entry con atributos multivaluados

Es relevante señalar que la multivaluación, aunque poderosa, debe usarse con discreción. Ello porque una query sobre el directorio retornará los valores de los atributos multivaluados de una determinada entry, en cualquier orden. No existe ninguna priorización de valores en los atributos multivaluados. Si tenemos algún valor destacado o distinto en la lista (por ejemplo, un givenName destacado, un teléfono con mayor prioridad dentro de una lista de teléfonos, etc) la única forma es almacenar dicho valor en un atributo distinto.

2.5.2. Attribute Types

La definición de los Tipos de Atributo contiene los siguientes elementos:

[18] Esta denominación corresponde a nomenclatura estándar en LDAP, lo que trataremos en los párrafos siguientes

- ▷ Name
- ▷ Object Identifier (OID)
- ▷ Syntax
- ▷ Matching Rules
- ▷ Inheritance

Veamos cada uno de ellos.

Name

Un *Attribute Type Name* es un string, case-insensitive, que puede contener letras, números, guiones (-). También se puede utilizar punto y coma (;) pero sólo en algunos casos especiales. Además, como práctica habitual se utiliza el estilo de nombre "camelcase", habitual en algunos lenguajes de programación. Este consiste en el uso de minúsculas, palabras pegadas, con uso de mayúsculas en la primera letra de cada palabra, a contar de la segunda palabra en adelante. Por ejemplo:

- · telephoneNumber
- · anotherTelephoneNumber
- · displayName
- · x509Certificate
- · test-data

Object Identifier (OID)

Además de su nombre, un attribute type posee un *Attribute Type Object Identifier*. Este se conforma como una secuencia de números separados por puntos, que siempre es única. En diferentes implementaciones LDAP se pueden tener nombres diferentes para el mismo attribute type, pero al asignar una determinado OID, es posible mapear los attribute types de una implementación a otra. Además, el OID sirve como identificador único para las APIs que utilicen la información del LDAP, y esto facilita la independencia de la implementación.

Los OIDs de alto nivel son asignados por la American National Standards Institute (ANSI) [19] de Estados Unidos. Una organización o empresa también puede solicitar disponer de un determinado OID.

Syntax

La *Sintaxis* del attribute type nos indica como manejar y operar los valores que contiene. Por ejemplo:

[19] http://www.ansi.org

- ▷ *DirectoryString*
  Strings Unicode codificados en UTF-8, generalmente case-insensitive
- ▷ *Binary*
  Datos codificados en binario
- ▷ *Certificate*
  Certificado digital, codificado en binario
- ▷ *Telephone Number*
  String que representa números telefónicos, donde los datos no numéricos son ignorados

Así como los atributos, las definiciones de sintaxis también poseen un OID. Por ejemplo:

- Binary - 1.3.6.1.4.1.1466.115.121.1.5
- Certificate - 1.3.6.1.4.1.1466.115.121.1.8
- Directory String - 1.3.6.1.4.1.1466.115.121.1.15

Adicionalmente a la sintaxis, un attribute type puede definir un tamaño y límites (*size and bounds*). Sin embargo, el chequeo de cumplimiento de estos rangos es dependiente de la implementación. Algunos LDAP servers verifican el size and bound, y otros no.

Matching Rules

Las *Matching Rules* permiten definir reglas de comparación que son utilizadas durante las búsquedas de información en el Directorio.

Cada attribute type define su propio mecanismo o regla de comparación, y esto define a priori la forma de operar de las búsquedas que serán realizadas. Se pueden utilizar cuatro tipos de reglas:

- ▷ *Equality*
  Permite definir la reglas para comparación por igualdad de dos valores del atributo. Por ejemplo, *caseIgnoreMatch* indica que la igualdad no debe diferenciar mayúsculas/minúsculas.
- ▷ *Greater or less than*
  Permite definir la reglas para comparación por mayor/menor entre dos valores del atributo. Por ejemplo, si un atributo de tipo string contiene números, se puede especificar *integerOrderingMatch* indicando que utilice el orden numérico por sobre la comparación de tipo alfanumérico.
- ▷ *Substring*
  Permite definir la reglas para la operación de subconjunto de un string en otro. Por ejemplo, *caseIgnore- SubstringsMatch* indica que no se deben considerar mayúsculas/minúsculas en la comparación.
- ▷ *Subschema*
  Permite definir la forma de operar para las búsquedas de información de la metadata del directorio.

Inheritance

Muchos attribute types comparten características comunes. LDAP implementa un mecanismo de herencia simple para definir estas características y heredarlas a los "sub" attribute types.

En la figura 2.4 se observa el attribute type de nombre *name*, cuya sintaxis es CIS (Case Ignore String) y cuya Matching Rule para igualdad es CIM (Case Ignore Match). Se observan además los attribute types *surName*, *commonName* y *givenName* que heredan del attribute type denominado *name*, y por tanto comparten estas características.

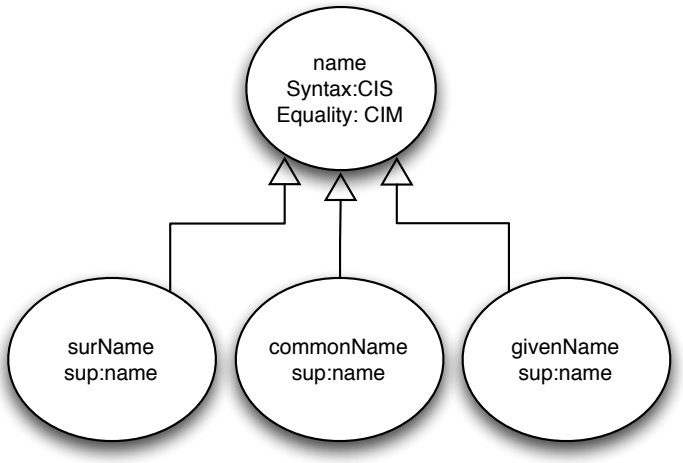


Figura 2.4: Ejemplo de herencia en Attribute Types

2.5.3. Object Classes

La definición de una object class nos pemite saber cuáles son los atributos requeridos y cuáles los permitidos. Esto es, todas las entries que pertenecen a dicha object class deben tener todos los atributos requeridos y posiblemente algunos de los permitidos. Adicionalmente la object class nos proporciona la información de la jerarquía de clases o herencia asociada.

En concreto, una object class queda definida por:

- ▷ Name
- ▷ Object Identifier (OID)
- ▷ Lista de attribute types requeridos (must)
- ▷ Lista de attribute types permitidos (can)
- ▷ Inheritance
- ▷ Class Type

Veamos estos elementos:

Name

Un *Attribute Type Name* es un string, case-insensitive, que puede contener letras, números, guiones (-) y punto y coma (;). También es frecuente el uso del estilo de nombre "camelcase".

Object Identifier (OID)

Los object class poseen un *Object Class Object Identifier*. Corresponde a una secuencia de números separados por puntos, que siempre es única. El OID sirve como identificador único para facilitar la compatibilidad entre APIs que utilicen la información del LDAP, la comparación de información entre diferentes directorios, etc.

Lista de attribute types

Los attribute types definen la información que contienen las entries. Los atributos obligatorios son aquellos que deben estar en la entry, y los atributos opcionales pueden aparecer, pero no son obligatorios.

Inheritance

LDAP permite implementar herencia simple. Habitualmente la herencia en una objectclass consiste en una especialización de una objectclass de nivel superior, por ejemplo agregando más atributos. Para especificar la herencia, en la definición de la clase "hija" se utiliza el atributo *sup* o *subclass of*, indicando el nombre de la clase "padre". Todas las objectclass heredan, al menos, de la clase *top*. Esta es la objectclass de más alto nivel en la jerarquía de clases de un LDAP (algo similar a la clase Object de Java). La objectclass top es la de mayor nivel y también es la más simple, pues consta de un único atributo obligatorio, el objectClass (la consecuencia lógica de esto es que, como todas las entries del LDAP pertenecen a algún objectclass, entonces obligatoriamente deben indicar dicha objectclass).

Es práctica habitual que en las entries de un LDAP, en el atributo objectclass se indique el nombre de la objectclass a que pertenece, pero además se agreguen líneas adicionales de objectclass indicando el nombre de las objectclass de nivel superior en la jerarquía de clases.

Un ejemplo puede verse en la tabla 2.7 :

dn: cn=Juan Pérez, o=Universidad de Chile, c=Chile
objectClass: organizationalPerson
objectClass: person
objectClass: top
cn: Juan Pérez
sn: Pérez
registeredAddress: Beauchef 850

Tabla 2.7: Ejemplo de entry, indicando todas las objectClass a que pertenece

Class Type

En LDAP existen tres tipos de clases:

▷ Abstract

Una objectclass es abstracta si sólo es usada como "superclase" de otras objectclass. Un uso habitual es definir una lista de atributos obligatorios que serán requeridos en las subclases que la heredan, Además, una objectclass abstracta nunca es usada para crear entries concretas. El mejor ejemplo de una objectclass abstracta es la objectclass top.

▷ Structural

Son las clases "concretas". Una entry puede pertenecer a varias object class (de acuerdo a la jerarquía), pero siempre pertenece a una única *structural object class*. En el ejemplo de la tabla 2.7, la entry pertenece a las object class organizationalPerson, person y top, pero la structural object class es organizationalPerson.

▷ Auxiliary

Permiten agregar algunos elementos o atributos secundarios a una entry. Un buen ejemplo es la protectedObject, que se puede incorporar como objectclass de cualquier entry. Esta object class actúa como una suerte de "marca" (indicando que la entry es un "objeto protegido"), y puede ser útil para algún determinado objetivo.

Es claro que no tiene sentido utilizar una auxiliary object class sin una structural object class, y que el uso de una auxiliary object class no cambia en nada la structural object class de las entry a la cual se agrega.

2.5.4. LDAP Schema

El *schema* de un LDAP server corresponde a la definición de la información que dicho LDAP server puede manejar. Esto es, los object class, la jerarquía de clases, los attribute

types, su sintaxis y herencia, etc.

En la práctica, un esquema se especifica en términos de definir completamente sus:

- ▷ object classes
- ▷ attribute types

En la figura 2.5 vemos un diagrama de un schema de un LDAP, que contiene la definición de los object class A y B y los attribute types 1, 2, hasta 4, que soporta este LDAP.

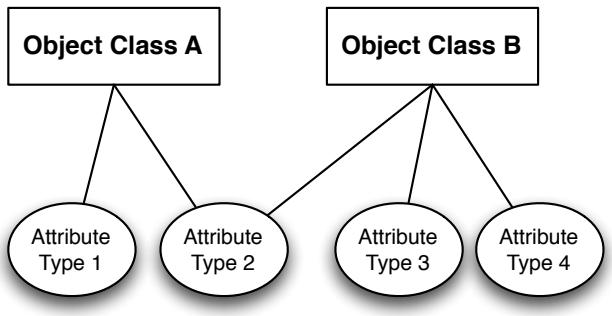


Figura 2.5: Ejemplo de schema de un LDAP

2.5.5. LDAP Schema estándar

¿Porqué definir un schema estándar?

La respuesta clave es una sola: *Compatibilidad*. Si bien es cierto que LDAP soporta la creación arbitraria de object classes y attribute types, disponer de un schema estándar predefinido va en directo beneficio de la compatibilidad y la consistencia de las aplicaciones clientes, el pareo y comparación de información entre distintos directorios, etc.

Por esta razón, antes de crear un nuevo schema, es importante primero averiguar qué está disponible en el schema estándar, y cuánto de allí es directamente utilizable en mi diseño en particular.

¿Dónde encontrar estos estándares?

La Internet Engineering Task Force (IETF) ha definido:

- ▷ RFC 2252 define el esquema operacional de LDAP
- ▷ RFC 2256 define varios object classes estándar para personas, grupos y organizaciones

Nombre	sup	must	may
person	top	sn, cn	userPassword, telephoneNumber, seeAlso, description
organizationalPerson	person		title, x121Address, registeredAddress, destinationIndicator, preferredDeliveryMethod , telexNumber, teletexTerminalIdentifier, telephoneNumber, internationaliSDNNumber, facsimileTelephoneNumber, street, postOfficeBox, postalCode, postalAddress, physicalDeliveryOfficeName, ou, st, l
organization	top	o	userPassword, searchGuide, seeAlso, businessCategory, x121Address, registeredAddress, destinationIndicator, preferredDeliveryMethod, telexNumber, teletexTerminalIdentifier, telephoneNumber, internationaliSDNNumber, facsimileTelephoneNumber, street, postOfficeBox, postalCode, postalAddress, physicalDeliveryOfficeName, st, l, description

Tabla 2.8: ejemplo schema estándar

Por ejemplo, las object classes más frecuentemente usadas son *person*, *organizationalPerson*, *organization*, etc. [20]

Por supuesto en otras RFCs existen definiciones estándares adicionales.

En la tabla 2.8 puede verse un ejemplo de algunas object class del schema estándar:

Además, cabe señalar que cada proveedor de LDAP efectúa adiciones/ajustes al schema estándar. De modo que la información exacta debe ser consultada en la documentación del proveedor respectivo.

¿Entonces, cuál es la forma de modelar en LDAP?

Analicemos el siguiente ejemplo. Deseamos modelar una objectclass donde almacenar información de las personas de nuestra organización. Como observamos en la tabla de schema estándar (tabla 2.8), disponemos de la clase Person, que es una clase estructural. ¿Qué ocurre si los atributos que contiene no son suficientes? Busquemos otra objectclass. Podría servir la OrganizationalPerson, que hereda de Person y dispone de más atributos. Pero también podría ocurrir que no fuese suficiente.

Obviamente, si no existe una objectclass adecuada siempre podemos crear una nueva. Pero en este caso, tenemos una mejor alternativa.

[20] Muchas de estas clases están basadas en las definiciones del protocolo X.500

En el ejemplo, disponemos de dos clases que se asemejan al tipo de entries que queremos almacenar. Las buenas prácticas nos sugieren dos mecanismos posibles:

- ▷ Generar una subclase de la clase Person (o de OrganizationalPerson, la que más se asemeje) para crear una nueva clase estructural que incluya aquellos atributos adicionales que necesitamos
- ▷ Definir una clase auxiliar que contenga todos los atributos adicionales que necesitamos, y agregarlos en cada entry de la clase Person, indicando en cada una de esas entries que su objectclass es Person y que además su objectclass también es el de esta nueva clase auxiliar (esto es lo que nos permite agregar los nuevos atributos)

¿Cuál es la mejor opción? Ambas son válidas, y ambas cumplen con el requerimiento. La decisión final depende del criterio y experiencia del modelador LDAP. Pero la directriz es clara: Usar el schema estándar mientras nos sea posible.

2.5.6. LDAP Data Interchange Format (LDIF)

Los LDAP server implementan diversos mecanismos para ingresar información al directorio. Habitualmente proporcionan interfaces interactivas o pantallas donde digitar la información de cada entry. Sin embargo, muchas veces es necesario intercambiar información entre directorios o simplemente efectuar cargas masivas de datos.

Para estos efectos se utiliza una especificación estándar de intercambio y carga de datos, en formato de archivos planos. Estas son:

- ▷ LDIF (LDAP Data Interchage Format)
- ▷ DSML (Directory Services Markup Language)

LDIF - LDAP Data Interchage Format

Es un formato de intercambio de información, sencillo y de uso frecuente. Existen tres variantes [21]:

- ▷ slapd.conf Schema
- ▷ ASN.1
- ▷ LDAPv3

Aunque presentan pequeñas diferencias, el más común es el slapd.conf. En la tabla 2.9 vemos una entry escrita en LDIF, en formato slapd.conf:

[21] Howes *et al.* [8, capítulo 7]

dn: cn=Juan Perez, dc=udechile, dc=org cn: Juan Perez sn: Perez objectClass: person objectClass: organizationalPerson

Tabla 2.9: Ejemplo de Entry en formato LDIF

DSML - Directory Services Markup Language

Es un formato de intercambio, que representa la información del directorio en XML. En la tabla 2.10 vemos el ejemplo anterior, ahora en este formato.

```
<dsml:dsml xmlns:dsml="http://www.dsml.org/DSML">
    <dsml:directory-entries>
        <dsml:entry dn="cn=Juan Perez, dc=udechile, dc=org">
            <dsml:objectclass>
                <dsml:oc-value>person</dsml:oc-value>
                <dsml:oc-value>organizationalPerson</dsml:oc-value>
            </dsml:objectclass>
            <dsml:attr name="cn">
                <dsml:value>Juan Perez</dsml:value>
            </dsml:attr>
                <dsml:attr name="sn">
                <dsml:value>Perez</dsml:value>
            </dsml:attr>
        </dsml:entry>
    </dsml:directory-entries>
</dsml:dsml>
```

Tabla 2.10: Ejemplo de Entry en formato DSML

LDIF - modificando el directorio

Es frecuente que los LDAP server permitan mantener el directorio a través de pantallas, vía comandos. etc. Además, siempre está presente el mecanismo batch usando LDIF. La sintaxis es muy sencilla y permite efectuar CRUD (Create, Read, Update and Delete) sobre los datos del LDAP.

En la tabla 2.11 se muestra un ejemplo para agregar una entry a un directorio LDAP.

```
dn: cn=Edmundo Gonzalez, dc=udechile, dc=org
changetype: add
cn: Edmundo Gonzalez
sn: Gonzalez
objectclass: person
objectclass: organizationalperson
```

Tabla 2.11: Agregando una Entry en formato LDIF

LDIF - representación de Schemas

LDIF también permite representar la información del schema del directorio LDAP. En la tabla 2.12 se muestra la información de un object class de nombre *person*, que hereda del object class *top*. Sus atribute type obligatorios son *sn* y *cn*. Y sus attribute type opcionales son *userPassword*, *telephoneNumber*, *seeAlso* y *description*.

```
objectclasses: ( 2.5.6.6 NAME 'person' SUP top
     MUST ( sn $ cn )
     MAY ( userPassword $ telephoneNumber
          $ seeAlso $ description ) )
```

Tabla 2.12: Ejemplo de Schema en formato DSML

En la tabla 2.13 vemos un ejemplo de representación LDIF de un attribute type:

```
attributetypes: ( 2.5.4.35
    NAME 'userPassword'
        EQUALITY octetStringMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.40
    USAGE userApplications )
```

Tabla 2.13: Ejemplo de attribute type en formato LDIF

Si queremos modificar el Schema, también es posible. En la tabla 2.14 se muestra como agregar un attribute type a un schema LDAP.

```
dn: cn=schema
     changetype: modify
     add: attributetypes
     attributetypes:
         ( 1.2.3.4 NAME 'myAttribute'
         DESC 'My Own Attribute' )
```

Tabla 2.14: Agregando un attribute type al LDAP schema, en formato LDIF

2.6 LDAP - Modelo de Nomenclatura, Namespace

Cada entry en un directorio LDAP tiene un identificador único, su *dn* o *distinguished name*. El modelo de nombres o *namespace* de un directorio LDAP corresponde precisamente a los nombres usados y permitidos dentro de dicho directorio.

Los nombres de las entries en el directorio son organizados jerárquicamente. La estructura y organización de estos nombres es lo que se denomina *Directory Information Tree* o simplemente DIT.

Ahora bien, ¿Cuál es el sentido de esta organización jerárquica? Esto obedece a la funcionalidad que LDAP proporciona y se encuentra estrechamente relacionada con esta forma de estructura. Lo veremos en los siguientes párrafos.

2.6.1. Funciones del DIT

▷ El DIT permite que los nombres de las entidades sean únicos en toda la Organización.

En la Universidad de Chile pueden existir dos alumnos de nombre Juan Pérez. Pero sus *dn* en el LDAP serán distintos y perfectamente identificables. Por ejemplo, el primer Juan Pérez tendrá un "dn: uid=JPerez001, dc=udechile, dc=org" y el segundo Juan Pérez tendrá un "dn: uid=JPerez002, dc=udechile, dc=org".

▷ El DIT puede ser distribuido

Partes del DIT pueden ser alojados en diferentes servidores, obedeciendo a criterios organizacionales o de arquitectura. Es frecuente que los LDAP server permitan realizar

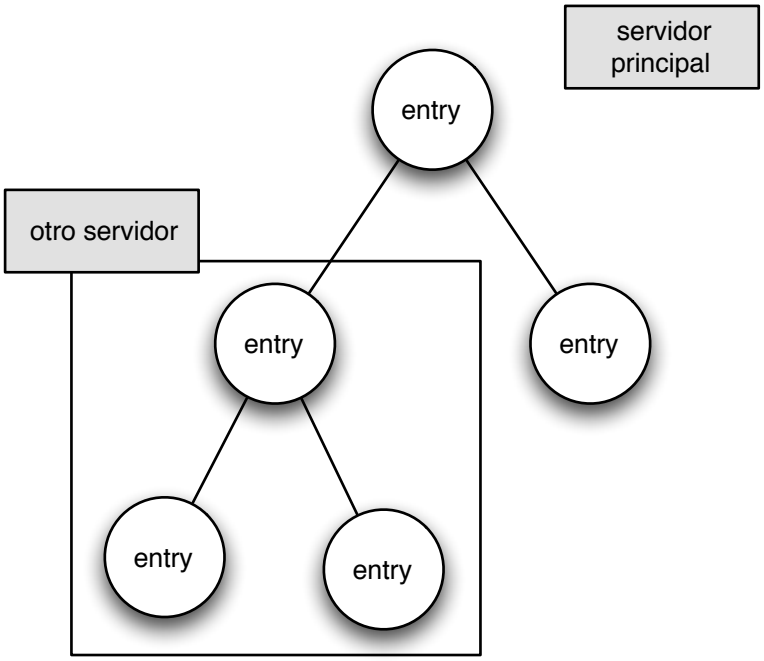


Figura 2.6: Distribución de un DIT

esta distribución, a través de la jerarquía del DIT (el árbol bajo un nodo determinado), tal como se aprecia en la figura 2.6.

▷ El DIT permite definir elementos de Seguridad

Un ACL o *Access Control List* define una regla de seguridad de acceso sobre el segmento donde se aplica. Aunque aún no es parte del estándar, en el caso de LDAP la aplicación de un ACL habitualmente se define sobre una entry específica o sobre la entry y todo su subárbol. Esto puede verse en la figura 2.7.

2.6.2. Distinguished Names (DN) y DIT

El identificador único de una entry es su distinguished name o simplemente dn. El dn se compone de dos partes:

▷ Relative Distinguished Name (RDN)

▷ La base

En la figura 2.8 vemos un ejemplo de un DIT, donde se muestra el RDN y la base de cada entry (nodo) del DIT.

En este ejemplo, cn identifica a la entry con el valor "Juan Pérez", y los otros dos atributos de su dn corresponden a los nodos o entries de niveles superiores en el DIT. Cabe consignar que, en este ejemplo, el atributo cn hace precisamente que la entry sea única dentro del

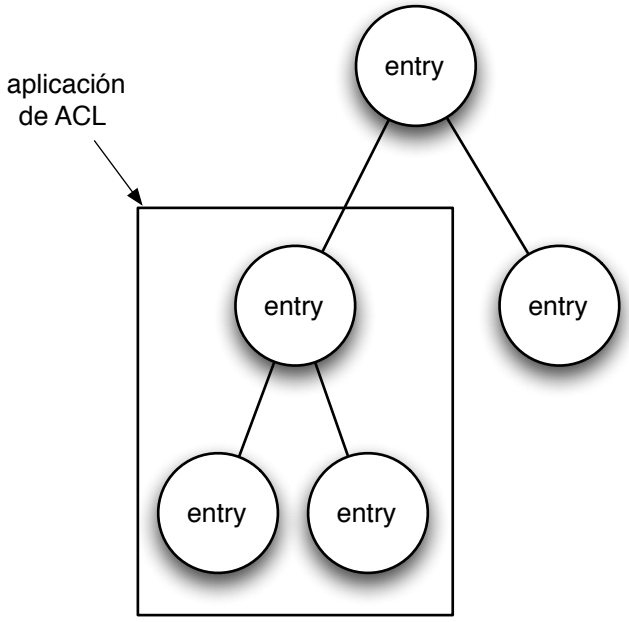


Figura 2.7: ACL sobre un DIT

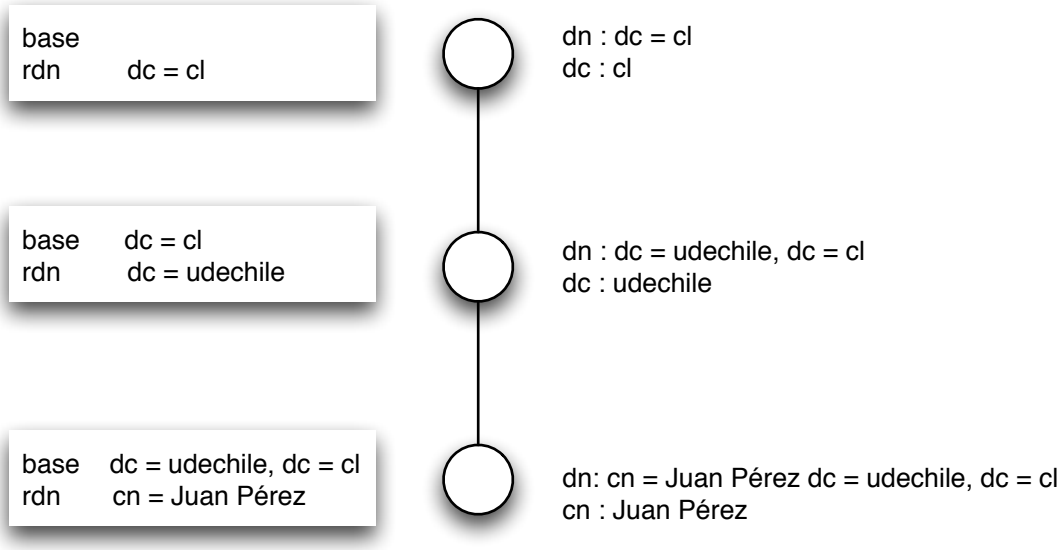


Figura 2.8: RDN y base en DIT

subárbol de información del directorio (y por tanto, única dentro de todo el DIT).

Al atributo (o conjunto de atributos) que genera esta unicidad dentro del dn se le denomina *relative distinguished name* (RDN). Por supuesto, como parte del diseño del DIT debe efectuarse una adecuada elección de RDNs, de forma de asegurar que se cumpla esta propiedad.

Candidatos a RDN

En la tabla 2.15 podemos ver ejemplos de atributos estándares, que habitualmente se usan como RDN. Es relevante recordar que puede utilizarse un único atributo o una combinación de ellos para generar el RDN de las entries.

Attribute Type	Contenido
cn	common name
l	locality name
st	state o province name
o	organization name
ou	organizational unit name
c	country name
street	street address
dc	domain component
uid	user identity

Tabla 2.15: Attribute types comúnmente usados como RDN

2.6.3. Server's Root Naming Context

La entry de más alto nivel en el DIT se denomina *server's root naming context*. Habitualmente se usan dos estilos para nombrarla:

- ▷ Traditional style
- ▷ Domain component style

Traditional style

En este caso para dar nombre a la raíz del DIT, se utiliza *o = company, c = country*, donde company corresponde al nombre de la empresa u organización y country es un código ISO estándar de dos caracteres, tal como US (Estados Unidos) o CL (Chile).

Esta convención es heredada del protocolo X.500, pero presenta un problema. No todas las empresas tienen registrado su nombre ante alguna entidad como la ANSI o la ISO, que

asegure que su nombre sea único a nivel global. Es posible que ocurran coincidencias de nombre con otras empresas, lo que podría resultar inconveniente en algunos casos.

Domain component style

La práctica más habitual hoy en día, en este ámbito, es usar *dc = company , dc = domain*, donde company y domain son dos o más partes de un nombre de dominio DNS de la organización. Por ejemplo, uchile.cl tendría un root naming context de dc=uchile, dc=cl.

2.6.4. Alias

Algunos directorios soportan entries de tipo *alias*. Esto es, una entry que referencia o apunta a otra entry dentro del directorio.

Una *alias entry* debe tener un objectclass alias y un atributo de nombre aliasedObjectName, cuyo valor contiene el dn de la entry a la cual se desea apuntar.

El uso de alias, de alguna forma vulnera el principio de ordenamiento jerárquico del DIT, y puede tener un alto impacto en los procesos de búsqueda, de modo que en general debe usarse con discreción.

2.6.5. Diseño del DIT

Diferentes diseños de DIT pueden presentar ventajas y desventajas. Por supuesto, esto depende de las características de la organización y del o los problemas que se quieren resolver.

En general, los DIT puede ser de tipo *flat (plano)* o *deep (profundos)*. Los árboles flat son aquellos que presentan poca jerarquía, mientras que los deep tienen muchos niveles jerárquicos. Es una buena práctica mantener el DIT lo más plano posible, manejando un nivel "adecuado" de jerarquía, teniendo en mente la distribución de la información y los niveles de control que se requieren.

Relevancia de un buen diseño de DIT

En definitiva, el DIT permite estructurar y organizar las entries que representan a las diferentes entidades dentro de una organización. Esto posibilita ordenar las búsquedas de información, definir reglas de accesabilidad y autorización, segmentar el directorio y delegar funcionalidad, definir la arquitectura y criterios de distribución de los servidores, etc.

Por ello el diseño del namespace y el DIT constituye una importante decisión dentro de las actividades de desarrollo de un LDAP.

2.7 LDAP - Modelo Funcional

Habiendo descrito la forma en que LDAP organiza y almacena su información podemos ahora mostrar cómo opera LDAP y cómo interactúa con los usuarios y sistemas que requieren sus servicios y operaciones.

En la tabla 2.16 se muestran estas operaciones [22] agrupadas por categoría.:

Categoría	Operaciones LDAP
connect session	Bind, unbind, and abandon
queries	Search and compare
update	Add, modify, modifyRDN, y delete
extended	Extended operations, LDAP controls

Tabla 2.16: Operaciones LDAP

En los párrafos siguientes describiremos estas operaciones [23].

2.7.1. Connect Session

LDAP tiene dos operaciones de autenticación (bind y unbind) y una operación de control (abandon).

Conexión con el servidor LDAP

El mecanismo esquemático de comunicación entre un sistema usuario de los servicios LDAP (cliente LDAP) y un servidor LDAP sigue los siguientes pasos:

- ▷ El cliente establece una sesión con un servidor LDAP. Esto se denomina *binding* con el servidor. La conexión habitual es sobre protocolo TCP/IP.
- ▷ El cliente puede proporcionar un nombre de usuario y password para autenticarse frente al servidor, o bien establecer una sesión anónima (DN y password vacíos).
- ▷ El cliente efectúa operaciones sobre el directorio. Estas pueden ser de lectura o modificación y sujeto a los permisos que ostente el cliente, de acuerdo a su autenticación efectuada en el paso anterior. Habitualmente las operaciones son de búsqueda sobre una o varias entradas del Directorio.
- ▷ Finalmente, al terminar las operaciones, el cliente cierra o finaliza la sesión frente al servidor. Esto se denomina *unbinding*.

[22] Arkills [2, capítulo 1]
[23] Howes *et al.* [8, capítulo 2]

Bind

Mediante la operación de *bind* el cliente se conecta al servidor. El cliente provee un DN y credenciales y el servidor puede verificar si éstas corresponden al DN entregado. Si es el caso, el cliente continúa autenticado mientras la sesión permanezca abierta o hasta que el cliente efectúe una nueva autenticación. El servidor asigna privilegios al cliente en base a su identidad.

Existen varios mecanismos de seguridad asociados al bind. Desde el intercambio de información en texto claro, hasta el uso de encriptación de conexiones usando Secure Sockets Layer (SSL) o Transport Layer Security (TLS), o bien utilizar SASL, que es un framework para efectuar negociación e intercambio de parámetros y credenciales de seguridad.

El cliente puede efectuar un bind, realizar algunas operaciones con esa identidad, efectuar un nuevo bind y luego realizar operaciones adicionales con la nueva identidad.

Unbind

La segunda operación de autenticación es el *unbind*. Esta operación no tiene parámetros. Cuando el server recibe una instrucción de unbind, procede a desechar toda la información relativa al cliente LDAP y su conexión, termina cualquier operación que esté en curso y desconecta al cliente, terminando la conexión TCP.

Abandon

Se trata de un mecanismo de control. Esta operación tiene un único parámetro: el message ID de la operación LDAP que se desea abandonar. El cliente LDAP efectúa esta operación cuando ya no está interesado en los resultados de dicha operación. Cuando el server recibe la instrucción, aborta o termina el procesamiento de la operación correspondiente al message ID. La operación *abandon* es usada tipicamente cuando el usuario cancela un search request que está demorando demasiado.

2.7.2. LDAP queries

Search

Esta operación permite buscar entries en el DIT que cumplen algún criterio. En LDAP no existe una operación "LDAP read" para obtener una entry específica. En su lugar se debe utilizar un search indicando un criterio que restringa la búsqueda a la única entry que se desea recuperar.

El LDAP search utiliza los siguientes parámetros:

1. Base object para el search

   Es un DN e indica la entry u objeto base desde donde se iniciará la búsqueda.

2. Search scope

   Hay tres tipos de scope. Un scope *sub* indica que la búsqueda se debe efectuar en todo el subárbol cuya raiz es el objeto base (incluido). Si el scope es *onelevel* la búsqueda debe efectuarse sólo en los hijos inmediatos al objeto base. Por último, un scope *base* indica que la búsqueda se limita al objeto base exclusivamente.

3. Alias dereferencing options

   Indica al servidor cómo tratar los alias durante el proceso de búsqueda. Recordemos que un alias es una suerte de puntero a otra entry en el DIT. Las opciones son *neverDerefAliases* (no buscar en los alias), *derefInSearching* (buscar en todos los alias excepto el objeto base), *derefFindingBaseObj* (usar los alias sólo en el objeto base) y *derefAlways* (mirar en todos los alias).

4. Size limit

   Indica la cantidad máxima de resultados (entries) que se desea recibir. Si lleva el valor cero, indica al server que entregue todos los resultados que encuentre.

5. Time limit

   Máximo tiempo en segundos que el server debe dedicar a efectuar el search. Si el tiempo es excedido el server detendrá el proceso y entregará un mensaje de error. Por supuesto, dependiendo de la implementación, un servidor puede manejar sus propios límites de time-out, independientemente de lo indicado en este parámetro.

6. Attributes-only parameter

   Si es seteado en *true* indica que sólo se requieren los attribute types como respuesta, pero no los attribute values. Si su valor es *false* el server debe entregar tanto los types como los valores de los atributos.

7. Search filter

   El *search filter* es una combinación booleana de atributos y valores, que indica cuáles son las condiciones que las entries deben cumplir.

   La sintaxis más simple es "(attributetype operator value)". Los operadores son =, ∼=, :=, >= y <= (las opciones < y > se obtienen por negación). El operador ∼= busca valores "aproximados" (usando algún algoritmo de soundex). Además se puede usar el asterisco como wilcard (por ejemplo (cn=*uan*), que busca las combinaciones del

nombre Juan, Juanito, etc). Por último, el operador := que está disponible en LDAPv3, permite extender las reglas de matching (por ejemplo, para buscar información dentro del DN).

Estos filtros se pueden combinar usando &, | y ! (and, or y not), en notación de precesión (por ejemplo, !(cn = Juan)). Esto es similar a la notación que usa Lisp o la notación polaca.

8. Lista de attributes a retornar

   Permite indicar una lista de atributos a retornar en las entries resultantes. Si se deja en blanco, el server entregará todos los atributos de dichas entries.

Compare

Esta operación permite verificar si una entry contiene un cierto valor en un determinado atributo. El cliente envía el request al server, indicando el DN, el attribute name y el valor. El server efectúa la operación y entrega un resultado afirmativo o negativo según corresponda.

Es claro que esta operación podría efectuarse con un search con base igual al DN, un scope base y un filtro que verifique la igualdad. Sin embargo, existe una pequeña diferencia: En el caso que la entry no contenga el atributo, la operación *compare* retorna una indicación de tal hecho, en cambio el search simplemente no retorna resultados. Esto podría ser útil en algunas circunstancias, así como el que la operación *compare* es más compacta en términos de la cantidad de información intercambiada.

2.7.3. LDAP Update

LDAP tiene cuatro operaciones de update: add, delete, modify y rename (modificar el DN). Estas operaciones definen la forma en que se manipula la información en el directorio.

Add

La operación *add* permite crear nuevas entries en el directorio. Utiliza dos parámetros: el DN de la entry a ser creado, y una lista de atributos (attribute type, value). Para que la operación sea exitosa se deben cumplir las siguientes condiciones:

- ▷ El padre de la nueva entry debe existir en el directorio
- ▷ No debe existir otra entry con el mismo dn
- ▷ La nueva entry debe cumplir con el schema del directorio (la o las object class a que pertenece)
- ▷ El control de acceso para el cliente solicitante debe permitir dicha operación

Si las condiciones se cumplen, la entry es agregada en el directorio.

Delete

La operación *delete* elimina una entry del directorio. Tiene un único parámetro: el DN de la entry a ser borrada. Para que el delete sea exitoso se deben cumplir las siguientes condiciones:

- ▷ La entry debe existir
- ▷ La entry no debe tener hijos en el DIT
- ▷ El control de acceso debe permitir el borrado para dicha entry

Modify

La operación *modify* permite efectuar cambios sobre los atributos de la entry. Tiene dos parámetros: El DN de la entry a ser modificad y el conjunto de modificaciones a ser aplicado. Estas modificaciones pueden consistir en modficar los valores de un atributo, agregar atributos (attribute type, attribute value) e eliminar atributos de la entry. La operación es exitosa si:

- ▷ La entry existe
- ▷ Todas las modificaciones de atributos deben ser exitosas
- ▷ La entry resultande debe seguir cumpliendo con el schema del directorio
- ▷ El control de acceso debe permitir esta operación

Rename (Modify DN)

Esta operación es usada para renombrar y/o mover entries dentro del DIT. Tiene cuatro parámetros. El DN de la entry a ser renombrada, el nuevo RDN para la entry, un argumento opcional indicando el nuevo padre de la entry y el delete-old-RDN flag.

Si la entry a ser renombrada conservará su padre, entonces el nuevo padre se deja en blanco. Si el nuevo padre tiene valor, la entry será movida a esa nueva locación en el DIT. Por último, delete-old-RDN flag sirve para indicar si el RDN original será retenido o no como atributo en la entry.

Para que la operación sea exitosa, se deben cumplir las siguientes condiciones:

- ▷ La entry renombrada debe existir
- ▷ El nuevo nombre de la entry no debe estar en uso por otra entry
- ▷ El control de acceso debe permitir la operación

2.7.4. LDAP Extended

Las extensiones LDAP permiten que el usuario defina nuevas operaciones o efectúe ajustes sobre las operaciones ya existentes. Las operaciones extended son el contexto ideal para que las implementaciones específicas de LDAP agreguen su propia funcionalidad. Operaciones de diversa índole, funcionalidades adicionales de seguridad, son un buen ejemplo.

Respecto de las modificaciones a las operaciones estándares, se hace uso de los *LDAP Controls*. Se trata de parámetros que permiten a un cliente solicitar al server que ejecute una operación estándar en alguna forma sutilmente diferente. Un ejemplo es el *paged search control* que indica al server que entregue los resultados de las queries en bloques de páginas (en vez de todas al mismo tiempo).

Los server LDAP debe indicar en el *supportedControl* attribute (que se puede obtener al recuperar la información del schema) cuáles controles tiene implementados, de forma que el cliente pueda hacer uso de ellos.

Los LDAP controls constituyen un área en permanente desarrollo y revisión en los estándares LDAP.

2.8 LDAP - Modelo de Seguridad

La seguridad en LDAP está basada en el hecho que se trata de un protocolo orientado a conecciones. Es decir un cliente LDAP abre una conección en un LDAP server y efectúa varias operaciones sobre la misma conexión. En algún punto el cliente LDAP debe autenticarse y con ello establecer el nivel de permisos y accesos que tendrá durante esa sesión. Durante toda esta actividad, se han efectuado acciones de autenticación y autorización.

2.8.1. Autorización

Como hemos señalado, la autorización es un tema aún abierto en las especificaciones LDAP. Esto significa que la implementación exacta depende de cada proveedor, aunque resulta frecuente el uso de ACLs, groups y POPS.

2.8.2. Modelos de autenticación

Tal como hemos visto, el proceso de autenticación ante el server se fundamenta en el binding. Una entidad está enlazada a una conexión si ha logrado una exitosa autenticación. Si el cliente no se autentica, o si se autentica sin credenciales, entonces el cliente se encuentra enlazado anónimamente a dicha conexión.

En La versión 3 del protocolo, se reconoció la necesidad de definir un mínimo de especificaciones que fueran provistas por los servers. En este contexto, los LDAP servers se dividen en tres grupos con los siguientes requerimientos:

- ▷ Los LDAP servers de tipo read-only public pueden permitir autenticación anónima (sin password).
- ▷ Los servers que admiten autenticación basada en password deben soportar the DIGEST-MD5 SASL usando Digest Authentication como mecanismo SASL.
- ▷ Los servidores que requieren protección de sesión (encriptación) y autenticación deben implementar el StartTLS extended operation. Esta extensión para Transport Layer Security (TLS) permite a un cliente LDAP requerir encriptación del flujo de información compartido con el server, y permite al cliente y al server autenticarse recíprocamente utilizando certificados de clave pública.

En su redbook de LDAP, IBM se refiere a esta materia [24]. Veamos estos mecanismos.

Anónima o Sin autenticación

Se trata del método más simple de autenticación. Este mecanismo debe ser usado sólo cuando la seguridad de la data no es relevante. Un ejemplo podría ser el caso de una libro de información con direcciones de utilidad pública.

Autenticación básica

En este caso el LDAP client se identifica ante el server por medio de un DN y password. La password es enviada al server en texto claro (en algunas implementaciones se usa Base64). El server considera que el cliente está autenticado si el DN/password recibido coincide con el DN/password almacenado en el directorio.

SASL

SASL es un framework para agregar mecanismos de autenticación adicionales en protocolos orientados a conecciones, en ambientes Internet. SASL fue originalmente diseñado para el protocolo IMAP, pero ha evolucionado en un mecanismo más general absorviendo nuevos protocolos, en particular, LDAP. Opcionalmente, luego de la autenticación, es posible agregar un security layer para encriptación de la data intercambiada, como por ejemplo TLS.

El uso de SASL permite que métodos adicionales de autenticación, tales como smart cards o autenticación biométrica, sean fácilmente incorporados.

[24] IBM-LDAP [10, capítulos 2.5 y 2.6]

SSL y TLS

Secure Socket Layer (SSL) fue diseñado para proveer tanto autenticación como privacidad. TLS es el sucesor de SSL y es un estándar Internet.

SSL/TLS soporta autenticación del server (autenticado por el cliente), autenticación del cliente (por el server) y autenticación mutua. Además provee mecanismos para encriptar el socket TCP/IP, para proteger la data intercambiada sobre la red. SSL/TLS usa un método de clave pública para asegurar la comunicación y autenticar la contraparte de la sesión. Esto se logra con un par clave pública/privada. Cada parte encripta con su clave privada propia (y la clave pública de la contraparte), envía la información sobre la red, y la contraparte desencripta con la clave pública (del origen y la clave privada propia).

2.8.3. La relación entre seguridad y los directorios

La seguridad tiene que ver con las funciones de Autenticación, Autorización y Accounting. ¿Cuál es entonces la relación con LDAP?

LDAP no es un servicio de autenticación, ni tampoco de autorización. Sin embargo permite almacenar la información de identidades y credenciales utilizadas por esos servicios, y es posible usar LDAP directamente como un mecanismo de autenticación, a través de las verificación exitosa o fallida de las operaciones de binding. Además, LDAP puede ser usado como agente de autorización, a través de las listas de control de acceso (ACLs) y *policies* que son habituales como mecanismo para autorizar acceso a los recursos del directorio. Por último, todas las transacciones y acciones efectuadas sobre LDAP pueden ser registradas como parte del mecanismo de accounting. Ver la figura 2.9.

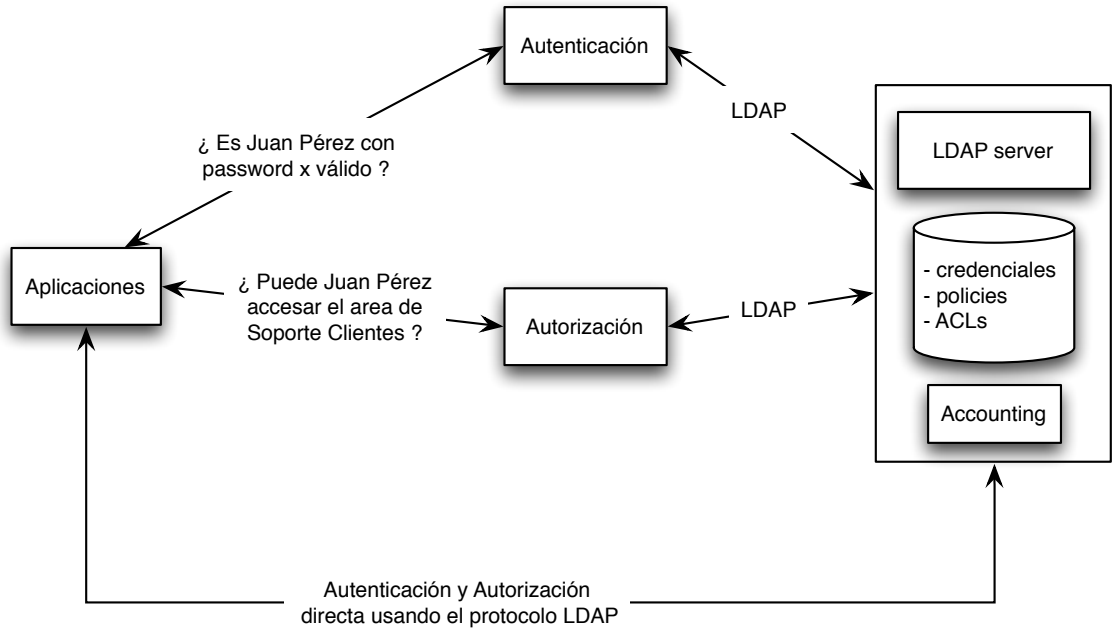


Figura 2.9: usando LDAP en autenticación y autorización

2.9 Conclusiones

Desarrollar un sistema de Gestión de Usuarios no se limita simplemente a construir una Base de Datos con cuentas y passwords. Se requiere una cantidad y variedad de elementos mucho mayor y conceptualmente sus objetivos son mucho más ambiciosos: En suma, se trata de crear una componente de software con altos estándares de calidad y que colabore activamente en garantizar la seguridad al interior de la Organización.

Los elementos que debe considerar son:

- Administración
- Autenticación
- Autorización
- Accounting (Auditoría, Logging, Tracing)

Los atributos de calidad que deben estar presentes son:

- Disponibilidad y Tolerancia a Fallas
- Integridad
- Rendimiento (Performance)

Los ambientes de seguridad a considerar son:

- Acceso Local
- Red
- InterSite
- Web

La utilización de Patrones de Seguridad permite aprovechar experiencias exitosas anteriores y define una guía para asegurar la calidad y completitud de los diferentes aspectos de diseño e implementación del sistema Gestión de Usuarios.

Por su parte, el uso de Tecnologías nos permite acelerar el proceso de desarrollo del sistema Gestión de Usuarios, aportando paralelamente varios de los requisitos de calidad que son necesarios. En este sentido, la tecnología LDAP resulta una valiosa contribución, por cuanto nos permite contar desde un comienzo, con una plataforma segura, estándar y con procedimientos de uso y conectividad completamente definidos.

Capítulo 3

El Problema

Según se ha señalado, Fonasa se encuentra llevando a cabo un proyecto de cambio de todas sus plataformas de servicio. Se trata un proyecto de alto impacto en la organización y afecta no sólo los sistemas computacionales, sino también sus procesos y procedimientos.

Este proyecto ha sido dividido en fases o etapas (bloques de sistemas) y la Gestión de Usuarios -el tema de la presente Memoria- es parte del primer bloque de sistemas a desarrollar.

En este capítulo, a modo de contexto, se describirá en líneas generales el proyecto Global de Fonasa, y luego profundizaremos, en forma específica, en el sistema Gestión de Usuarios.

Primero se mostrará la situación de la administración y gestión de usuarios en los sistemas Fonasa, existente con anterioridad a la implantación del nuevo sistema Gestión de Usuarios. Luego se describirán los elementos que debe considerar el nuevo sistema y, como consecuencia natural, se documentará el levantamiento formal de requerimientos efectuado.

Finalmente, se entregará la planificación o Plan de Trabajo que permitió efectuar el desarrollo e implantación del sistema, y que sirvió de guía durante la ejecución de todos los trabajos asociados.

3.1 El Proyecto Global de Fonasa

El Fondo Nacional de Salud (Fonasa) es un organismo del Estado de Chile cuya función es proporcionar a sus beneficiarios una serie de prestaciones, cumpliendo de esta forma con la legislación vigente [1].

La modernización del aparato estatal y sus reparticiones, es un objetivo recurrente de la Agenda Digital de nuestro país. Fonasa no está ajeno a este objetivo, y como tal ha iniciado un Proyecto Global de gran envergadura, consistente en el cambio, mejora y ampliación de todos sus sistemas corporativos. Esto incluye la totalidad de sus plataformas, procesos y todos los sistemas computacionales asociados.

Los objetivos son:

- ▷ Agilizar, modernizar y mejorar sus servicios
- ▷ a nivel central y regional
- ▷ tanto para usuarios y beneficiarios, así como su gestión interna.

Se trata de un proyecto de largo aliento, que dada su cobertura y alcance, tendrá un gran impacto a nivel de la organización, así como en sus beneficiarios a lo largo de todo el país.

Para cumplir con este objetivo, Fonasa llamó a una licitación, en la cual se considera el cambio de todos sus sistemas, entre los cuales se encuentra el sistema Gestión de Usuarios.

La nómina de sistemas considerados por Fonasa en su Proyecto Global es:

- ▷ Acreditación
- ▷ Cotizaciones
- ▷ Prestadores y Convenios
- ▷ Prestaciones
- ▷ Percápita
- ▷ Fiscalización de Cotizaciones
- ▷ Fiscalización de Prestadores
- ▷ Varios Procesos de Apoyo (Gestión de Usuarios y otros)

Fonasa se refiere globalmente a su nueva plataforma como *SCI, Sistema Corporativo de Información.*

3.1.1. Fases del Proyecto Global de Fonasa

Dada la cantidad y variedad de sistemas a desarrollar, el proyecto global ha sido dividido en fases o etapas (bloques de sistemas). De esta forma, habrá una primera fase con un grupo

[1] mayores antecedentes de Fonasa como organización en Anexo 1

de sistemas que se desarrollarán e implementarán en conjunto, y luego fases adicionales donde se irán incorporando el resto de los sistemas requeridos.

En el contexto de esta nueva plataforma, y desde un comienzo, se requiere la implementación de un sistema que gestione el uso y acceso de los usuarios a las nuevas aplicaciones: El sistema Gestión de Usuarios.

Las fases son:

- ▷ Primera Fase (Etapa)
  - Gestión de Usuarios
  - Acreditación
  - Cotizaciones
- ▷ Siguientes Fases (Etapa)
  - Fase o etapa 2 de Gestión de Usuarios (integración a RRHH [2] y funcionalidades avanzadas)
  - Resto de las aplicaciones

3.1.2. Levantamiento de Requisitos Gestión de Usuarios

Ingeniería de Requisitos

En esta etapa del desarrollo de la Gestión de Usuarios, que es previa al diseño, se buscó definir con el mayor detalle posible, todos los requerimientos surgidos de los diferentes stakeholders del Sistema.

Las actividades son:

- ▷ Especificación inicial
- ▷ Reuniones con usuarios para levantamiento de requisitos
- ▷ Especificaciones formales y validación
- ▷ Ajustes
- ▷ Especificación final, sancionada por los stakeholders

Stakeholders

Como tal identificamos a los diferentes usuarios y actores que participan y son afectados por las decisiones de diseño del nuevo sistema Gestión de Usuarios. En consecuencia, se trata

[2] Recursos Humanos

de usuarios cuyas ideas y opiniones deben ser consideradas en los requerimientos y atributos de calidad del sistema.

Nos referimos a:

- ▷ Usuarios funcionales del sistema Gestión de Usuarios
- ▷ Usuarios técnicos
- ▷ Usuarios auditores del Proyecto Global
- ▷ Desarrolladores de otros sistemas/plataformas asociados al nuevo Proyecto Global
- ▷ Desarrolladores de sistemas legacy que requieren integración

3.2 Situación Anterior

3.2.1. Estado de la Gestión de Usuarios antes del nuevo sistema

El Proyecto Gestión de Usuarios es parte de la primera etapa del Proyecto Global de Fonasa.

Como tal, este sistema debe cubrir todos los aspectos relacionados con el manejo, administración y gestión de los diferentes usuarios de Fonasa [3] y el control de los accesos y permisos asociados.

En el entorno actual existe una variedad de plataformas y sistemas, cada uno con sus propios mecanismos de seguridad y control, con múltiples esquemas de manejo de usuarios, con cuentas de acceso duplicados, con escasa correlación entre unos sistemas y otros y sin la posibilidad de efectuar chequeos cruzados de consistencia, ni tampoco realizar una administración centralizada de usuarios y permisos.

Los usuarios poseen diferentes cuentas de acceso, con diferentes passwords en diferentes sistemas, con más de un mecanismo de identificación (rut, código, correlativo, siglas, etc). Existen disímiles niveles de seguridad en los datos de usuario (protegidos, no protegidos, en diferentes mecanismos de almacenamiento) y con frecuentes niveles de inconsistencias entre una y otra plataforma de aplicación.

Esto refleja simplemente un crecimiento inorgánico, no controlado, de los mecanismos de gestión de usuarios, que han ido adoptando diferentes formas y mecanismos a medida que ha ido creciendo la cantidad y complejidad de las aplicaciones de Fonasa.

El nuevo sistema Gestión de Usuarios busca proporcionar un mecanismo único, de control centralizado y acceso distribuido, de tipo single-sign-on [4], con altos niveles de confidencia-

[3] entendiendo como tal, en un sentido amplio, a todo ente o individuo que tenga la facultad de interactuar de alguna forma y en determinado momento con los sistemas o plataformas de Fonasa

[4] Mecanismo de conexión o logon único hacia a los sistemas

lidad en la protección de los datos, y con altísimos niveles de rendimiento, disponibilidad y consistencia.

3.2.2. Problemas asociados

- ▷ Múltiples mecanismos de conexión
- ▷ Dificultad para los usuarios para recordar sus usuarios/passwords
- ▷ Inconsistencia entre plataformas
- ▷ Dificultad de mantención de usuarios/passwords
- ▷ Dificultad de mantención de permisos y atribuciones de los usuarios
- ▷ Laboriosos e inadecuados mecanismos de rastreo y log de actividades

3.3 Gestión de Usuarios

3.3.1. Definiciones

Usuario

Como Usuario se define toda aquella persona/entidad que debe interactuar de alguna manera con las plataformas/sistemas de Fonasa. De esta forma, un Usuario puede ser:

- ▷ Una persona/individuo
- ▷ Un sistema/plataforma interno de Fonasa
- ▷ Un sistema/plataforma externo a Fonasa

Persona/Individuo

El caso general de personas/individuos contempla tanto:

- ▷ Usuarios Internos (típicamente funcionarios). Los usuarios están asociados a una sucursal, o a las unidades centrales.
- ▷ Usuarios de otras reparticiones vinculadas a Fonasa, autorizados a ejecutar determinadas acciones sobre el sistema (esta vinculación es de tipo operativo o para fines de fiscalización por entes reguladores). Específicamente se consideran usuarios de los Establecimiento de Atención Primaria de Salud (EAPS).
- ▷ Público en General (aunque NO son requerimientos de esta fase)

Recurso

Los Recursos corresponden a:

- ▷ Sistemas/plataformas
- ▷ Funcionalidad específica dentro de sistemas/plataformas (diferentes niveles de consultas, ingreso de información, generación de transacciones, etc.)
- ▷ Servicios
- ▷ Bases de datos

Rol

Los Roles corresponden a grupos de tareas que deben ser realizadas por los usuarios. Para ejecutar estas tareas, un Rol debe disponer de acceso a determinados Recursos.

Si un usuario pertenece a un Rol, entonces debe tener acceso a los Recursos asociados a dicho Rol.

Cuenta

Las cuentas corresponden a identificadores de acceso de los usuarios en los diferentes Sistemas de Fonasa.

En otras palabras, una persona/funcionario/entidad:

- ▷ tiene asociado un único usuario
- ▷ tiene asociado múltiples cuentas, una cuenta por cada sistema de la organización

3.3.2. Tipos de Usuario

Los usuarios creados por el sistema Gestión de Usuarios, corresponden a alguno de los siguientes tipos:

Usuario normal o usuario general

Es un usuario que -desde el punto de vista de este sistema- no presenta ninguna característica especial. Considera tanto usuarios internos como de los EAPS.

Usuario solicitador

Es un tipo de usuario, que opera a nivel nacional, conformado por Jefes de Sucursal y Directores Regionales, que tiene la potestad de:

- ▷ solicitar creación/eliminación de usuarios y desbloqueo de contraseñas (reset de passwords). Los usuarios de este grupo efectúan la solicitud, pero la acción es derivada al *Grupo Resolutor* que es el que resuelve finalmente.
- ▷ solicitar asignación/eliminación de funciones (recursos) para un usuario determinado. Los usuarios de este grupo efectúan la solicitud, pero si la funcionalidad tiene un *Dueño de Recurso*, la acción es derivada a dicho dueño(s).
- ▷ solicitar asignación/eliminación de Roles para un usuario determinado. En este sentido, los Roles también pueden ser vistos como una suerte de "macro Recurso". Los usuarios de este grupo efectúan la solicitud, pero si el Rol tiene un *Dueño de Recurso*, la acción es derivada a dicho dueño(s).
- ▷ las funciones así como los roles pueden tener un dueño (o más de uno). Si existe algún dueño, cualquiera de ellos resuelve si aprueba o rechaza la solicitud. Si el recurso no tiene dueño la solicitud es autorizada automáticamente.
  Los permisos de recursos asociados a un rol son otorgados al usuario apenas se autoriza la solicitud (de inmediato si el rol no tiene owner, o bien en el momento que el owner autorice).

Usuario resolutor

Es un tipo de usuario que pertenece a un grupo definido por Fonasa, que opera centralizadamente, pero con ámbito de acción nacional. Sus funciones son autorizar/rechazar y ejecutar las creaciones/eliminaciones de usuarios y los desbloqueos de contraseñas.

Es un grupo conformado por personas del área Fonasa GTI [5].

Usuario dueño de recurso (Usuario owner)

Cada recurso (funciones y/o roles) puede eventualmente tener un dueño (o más de uno). Esto da lugar a grupos de dueños de recurso que operan y resuelven a nivel nacional y que autorizan/rechazan las solicitudes de permisos/accesos a sus respectivos recursos.

Los usuarios dueños de recurso habitualmente son personas de las áreas de Operaciones/Negocio de Fonasa. Los usuarios owner son usuarios de cualquiera de las categorías anteriores, pero además son *dueños (owner)* de algún Recurso en particular.

Aunque estamos adelantándonos en el modelamiento, en lo que sigue usaremos la siguiente denominación:

Rol
grupo de tareas o funciones en que debe desempeñarse un usuario (Ejemplo: Cajero, Supervisor, Ejecutivo, etc)

Recurso
grupo o unidad funcional dentro de un sistema, a través de las cuales se pueden realizan tareas específicas asociadas con dicha unidad funcional (Ejemplo: efectuar Acreditaciones, ingreso de Cotizaciones, consultas de beneficiarios, etc)

[5] Área de Sistemas de Fonasa, designada así por su función: Gestión de Tecnologías de Información

Matriz de Permisos

Existe una correlación entre los *roles* y los *recursos*. Los roles deben tener acceso a determinados recursos para poder efectuar las tareas de las cuales son responsables. Este concepto lo desarrollaremos más adelante durante el modelamiento de la solución, pero por ahora le daremos la denominación indicada,

3.3.3. Alcance de la Gestión de Usuarios en la Primera Fase

Alcance de esta fase - Usuarios

En esta primera Fase del desarrollo del Sistema Gestión de Usuarios se cubrieron las necesidades de usuarios internos y usuarios EAPS. Los usuarios del tipo "Público en General" NO son requerimiento de esta fase y por ende no fueron considerados.

Alcance de esta fase - Sistemas/Funcionalidad/Servicios de la Primera Fase

En esta primera Fase del Proyecto se ingresó la información (recursos/permisos) asociada a los sistemas/funcionalidades y servicios que entraron a producción en la primera Fase del Proyecto Global de Fonasa. No obstante, el sistema quedó preparado para recibir el ingreso de información del resto de los sistemas de las siguientes Fases, lo que ocurrirá naturalmente a medida que estos sistemas vayan terminando sus respectivos desarrollos e ingresando a operación normal.

3.3.4. ¿Qué es la Gestión de Usuarios?

La Gestión de Usuarios forma parte activa de la seguridad en la organización. No sólo debe administrar los elementos asociados, sino que también debe entregar dinámicamente servicios de autenticación/autorización por cada ente que lo requiera, así como llevar un registro detallado de todas las actividades efectuadas, a objeto de efectuar auditing, tracing y logging, analizar rendimiento y comportamiento, y aportar con los necesarios antecedentes para la prevención y detección de incidentes de seguridad.

Para ello debe considerar los siguientes elementos:

- ▷ Usuarios
- ▷ Cuentas
- ▷ Recursos
- ▷ Accesos (permisos)

La Gestión de Usuarios tiene por funciones:

- ▷ Administrar el Ciclo de Vida de los elementos del Sistema

- ▷ Entregar servicios de Autenticación y Autorización para los usuarios/plataformas que lo requieran
- ▷ Llevar un registro detallado de todas las actividades efectuadas sobre el sistema

3.3.5. Ciclo de Vida

El *ciclo de vida* de un elemento del Sistema Gestión de Usuarios corresponde a:

- ▷ conjunto de eventos/sucesos asociados a dicho elemento
- ▷ actividades y tareas que se deben efectuar ante cada uno de estos eventos

Veamos estos ciclos de vida en detalle:

Ciclo de vida de un usuario

- ▷ Surge un nuevo usuario en la organización. Esto se origina habitualmente en la contratación de un funcionario, en modalidad Planta o "a Honorarios".
- ▷ Creación del usuario y sus atributos básicos (demográficos y otros). Asignación de un rol (básico o específico)
- ▷ Creación de cuenta(s) básica(s) (por ejemplo, el nivel mínimo es Autoatención de Usuarios)
- ▷ Usuario en operación normal
  - Autoatención (cambio password)
  - Asignación/revocación roles/permisos
  - Creación/eliminación de cuentas asociadas al usuario
  - Inhabilitación/habilitación (por ejemplo, vacaciones)
- ▷ Cese de funciones (por desvinculación u otros eventos)
  - Eliminación/desactivación de cuentas
  - Eliminación/desactivación de roles/permisos

Ciclo de Vida de Cuentas

- ▷ Creación de cuentas. Ocurre en forma automática, asociado al momento de la creación de un usuario. Cabe señalar que la creación de una cuenta no otorga en ese momento ningún privilegio al usuario. Los privilegios sólo son establecidos en la asignación de roles o permisos.
  También puede ocurrir con posterioridad a la creación de un usuario, cuando surgen nuevos sistemas o plataformas para los que se requiere que el usuario tenga acceso, o bien, ante cambio de funciones, en los cuales sea necesario incorporar nuevas cuentas al usuario.

- ▷ Eliminación de cuentas. Ocurre en forma automática en el momento del cese de funciones de un usuario.
  También puede ocurrir en el cambio de funciones de un usuario, cuando ya no sea requerido su acceso a un sistema/plataforma determinado.

Ciclo de Vida de recursos

Los recursos corresponden a los sistemas o subconjuntos de funcionalidad dentro de los sistemas. Su ciclo de vida está asociado con la liberación de nuevos sistemas o funcionalidades, o la extinción y/o reemplazo de sistemas.

Ciclo de Vida de accesos/permisos

- ▷ Otorgamiento de acceso/permiso. Ocurre en forma automática, asociado al momento de la asignación de rol a un usuario.
- ▷ También puede ocurrir como resultado de una solicitud de permiso a un recurso que esté fuera del rol del usuario, en cuyo caso el permiso:
  - Es otorgado automáticamente si el recurso solicitado no tiene owner (dueño).
  - Queda a la espera de la autorización/rechazo del owner (o de alguno de los owner)
- ▷ Eliminación de acceso/permiso. Ocurre en forma automática, asociado al momento de la eliminación/cambio de rol de un usuario.
- ▷ También puede ocurrir como la revocación de un permiso otorgado expresamente y fuera del ámbito del rol, en cuyo caso la eliminación del permiso es inmediata.

3.3.6. Medioambiente e Interfaces

Corresponde al entorno sistémico y ambiental en que opera el sistema. Esto incluye procesos y procedimientos que lo afectan, así como normativas y marco regulatorio vigente.

- ▷ Usuarios interactuando en forma directa
- ▷ Sistemas Actuales
- ▷ Sistemas Nuevos del Proyecto Global en su primera Fase
- ▷ Sistemas Nuevos del Proyecto Global en sus restantes Fases
- ▷ Sistemas/Plataformas externas a Fonasa
- ▷ Normas internas de Fonasa
- ▷ Marco Regulatorio

3.3.7. Entradas de Información

- ▷ CRUD [6] de usuarios/cuentas
- ▷ Desbloqueo de contraseñas
- ▷ Administración de grupos de usuarios propietarios de recursos (listas de owner)
- ▷ Solicitud de asignación de rol o recurso a un usuario
- ▷ Aceptación/rechazo de solicitud de asignación de rol/recurso
- ▷ Autoatención de usuarios (cambio password/preguntas de autenticación [7])
- ▷ Solicitud de servicios de Autenticación
- ▷ Solicitud de servicios de Autorización

3.3.8. Salidas de Información

- ▷ Actualización de información del LDAP en base a la información ingresada:
  - CRUD de usuarios/cuentas
  - desbloqueo de contraseñas
  - administración de listas de owner de recursos
  - solicitud de asignación de rol/recurso
  - aceptación/rechazo de solicitud de asignación de rol/recurso
  - autoatención de usuarios
- ▷ Respuesta a la solicitud de servicios de Autenticación
- ▷ Respuesta a la solicitud de servicios de Autorización
- ▷ Informes de actividad

3.4 Requisitos

El desarrollo del Sistema Gestión de Usuarios de Fonasa debe tomar en consideración:

- ▷ tamaño de la estructura de Fonasa, la distribución geográfica de sus oficinas, sucursales y diversas reparticiones a lo largo de todo el país
- ▷ cantidad y variedad de usuarios
- ▷ cantidad de sistemas, y la coexistencia durante el proyecto de plataformas antiguas y nuevas
- ▷ proporcionar una base sólida para el ingreso de las nuevas aplicaciones que pasarán a producción paulatinamente, a medida que el proyecto global vaya avanzando

En definitiva, el sistema Gestión de Usuarios debe cumplir con los siguientes requisitos:

[6] Create, Read, Update and Delete
[7] preguntas utilizadas para recuperación de contraseña

3.4.1. Funcionales

1. Integración con los Sistemas de la Primera Fase del Proyecto Global
2. Capacidades básicas de Administración de Usuarios
   - ▷ Altas y bajas de usuarios / cuentas
   - ▷ Desbloqueo de contraseñas
   - ▷ Administración de grupos de owners
   - ▷ Solicitud/Autorización roles/recursos
   - ▷ Autoatención de usuarios (primera conexión, cambio de password/preguntas de autenticación)
3. Proveer servicios de
   - ▷ Autenticación
   - ▷ Autorización
4. Capacidades Avanzadas de Administración (según se indica más adelante, este requisito está fuera del alcance de esta primera etapa del proyecto)
   - ▷ Monitoreo de Accesos
   - ▷ Administración de flujos y correlación con sistema de Recursos Humanos (RRHH)
5. Migración de Datos
   - ▷ Definición de datos válidos y vigentes
   - ▷ Recolección, organización y carga inicial de datos de usuarios actuales

3.4.2. No Funcionales

1. Rendimiento o Performance
   - ▷ Rendimiento en ambientes de alto volumen transaccional [8]
   - ▷ Rendimiento garantizado en horarios peak
2. Seguridad
   - ▷ Controles de acceso (autenticación de usuarios)
   - ▷ Control accesos no autenticados
   - ▷ Autorización de accesos a recursos (autorización en base a permisos)
   - ▷ Control intentos de acceso no autorizados
   - ▷ Log y Auditoría de operaciones
   - ▷ todo esto corresponde a la conocida sigla AAA de seguridad (Autenticación/ Autorización/ Accounting)
3. Alta Disponibilidad y capacidad de Operación en Site Secundario
   - ▷ Alta disponibilidad propia (stand alone). Mecanismos de restauración.

[8] Los tiempos de respuesta de las transacciones deben dar cumplimiento a los niveles de SLA exigidos por el cliente. El sistema Gestión de Usuarios aporta una parte del tiempo en estas transacciones. El tiempo total esperado (que incluye este tiempo), aún cuando depende del tipo de transacción, en términos globales no debe ser superior a 1 segundo.

- ▷ Operación en alta disponibilidad para servicios brindados a sistemas y plataformas usuarias
- ▷ Operación controlada en Site Secundario

4. Integridad
   - ▷ Mínimos niveles de latencia en red para propagación de cambios
   - ▷ Consistencia en replicaciones en ambientes distribuidos
   - ▷ Consistencia con registros y Auditorías de operaciones
5. Interoperabilidad o Convivencia
   - ▷ Soportar operación concurrente con sistemas legacy de Fonasa así como con la nueva plataforma
6. Evolucionabilidad
   - ▷ Habilidad de crecimiento para soportar nuevas funcionalidades
   - ▷ Capacidad de admitir la incorporación de nuevos sistemas y plataformas usuarias

3.4.3. Procesos

Se muestran los diagramas con la descripción del flujo de procesos levantado. La intervención de Help-Desk [9] en esta etapa simplemente aporta como elemento de control y registro del flujo y para apoyo a las áreas usuarias.

- ▷ Creación usuario y Cese de funciones de usuario
  En la figura 3.1 observamos el proceso de creación, modificación y eliminación (cese) de Usuarios (CRUD).
- ▷ Desbloqueo de contraseña (password)
  Ver figura 3.2.
- ▷ Solicitud de roles/permisos
  Ver figura 3.3.
- ▷ Autoatención de usuarios (primera conexión)
  Ver figura 3.4.

3.4.4. Prioridades

Fonasa ha definido que el Proyecto Gestión de Usuarios debe cumplir con las siguientes prioridades:

1. Integración con los Sistemas de la Primera Fase del Proyecto Global
2. Cumplimiento estricto de todos los Requisitos No Funcionales
3. Capacidades básicas de Administración

En consecuencia, el requisito de Capacidades avanzadas de Administración queda fuera de esta etapa.

[9] Mesa de Ayuda

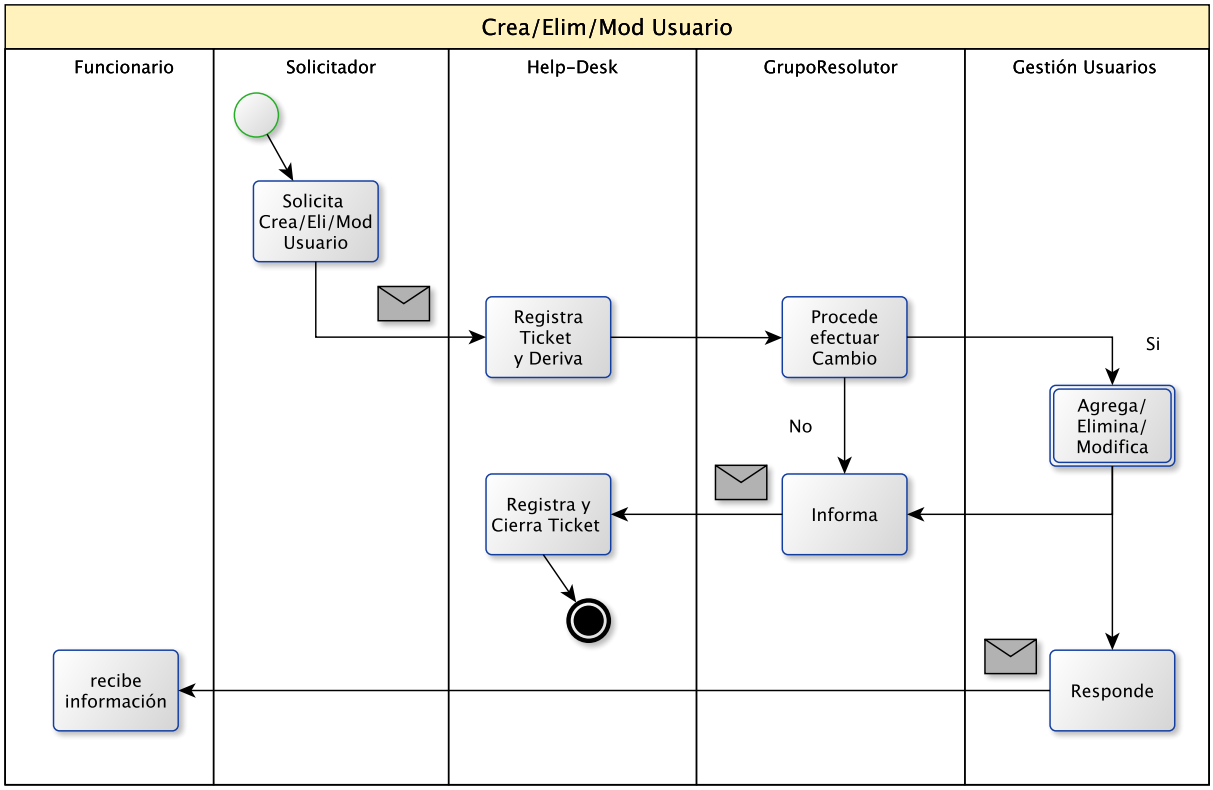


Figura 3.1: CRUD Usuarios

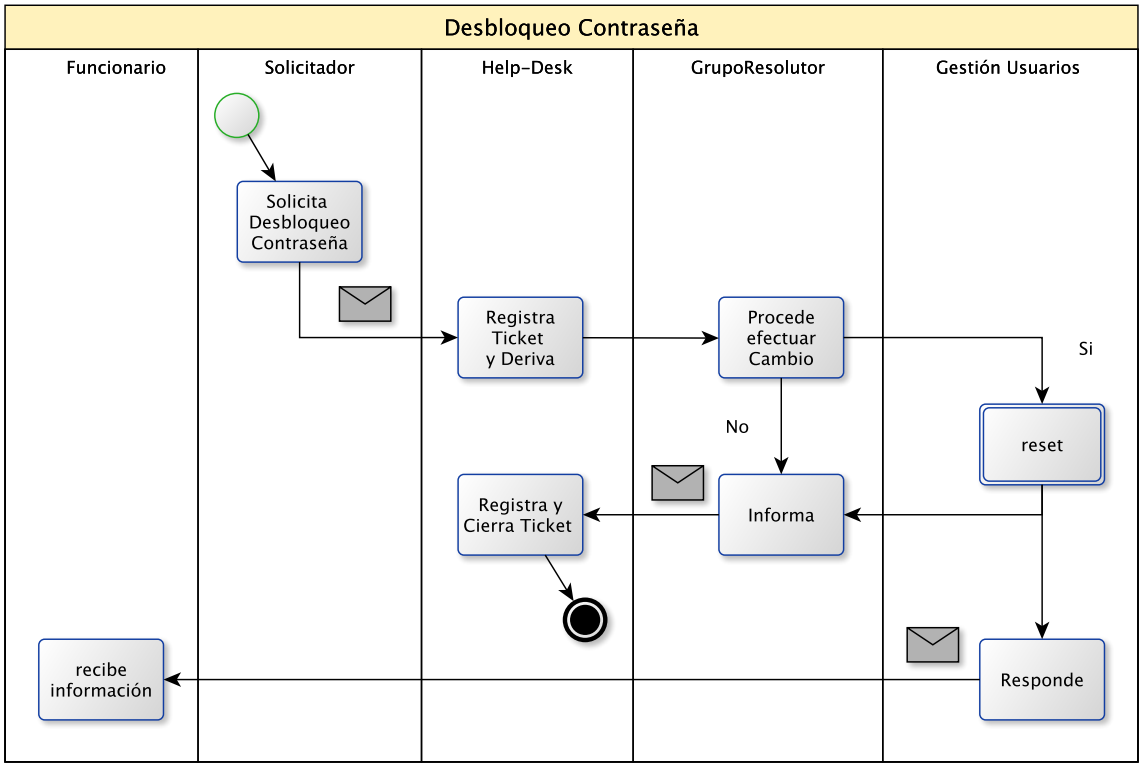


Figura 3.2: Desbloqueo de contraseña

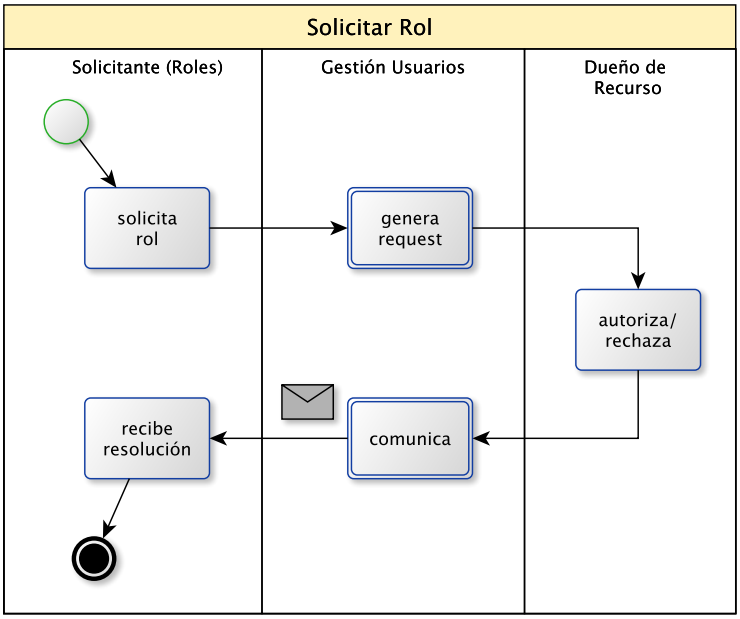


Figura 3.3: Asignación de Roles

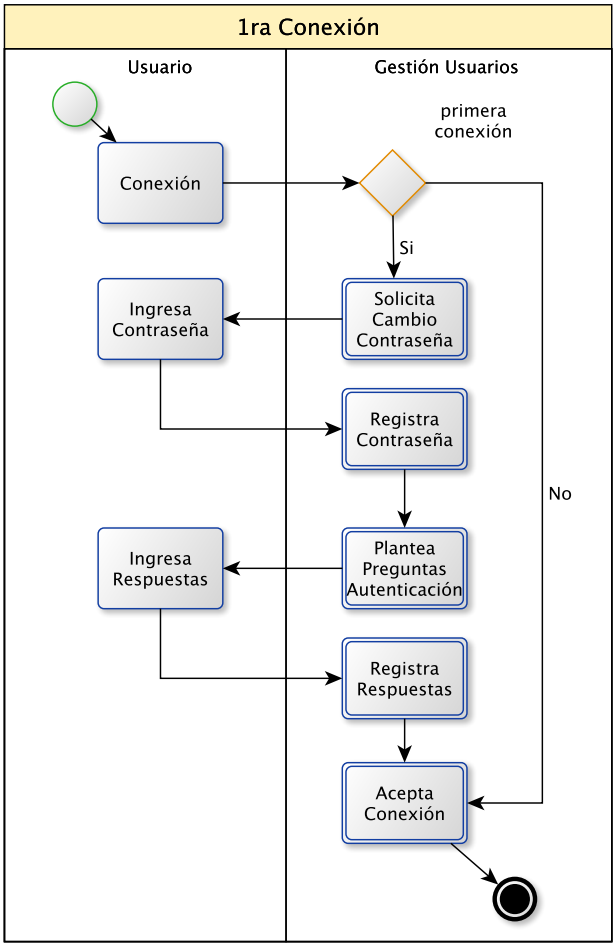


Figura 3.4: Primera conexión del usuario

3.4.5. Estrategia - Fases y Alcance del Proyecto

Alcance del Proyecto

Se logró un acuerdo con Fonasa, en el sentido de dividir el proyecto Gestión de Usuarios en dos fases/etapas, sincronizadas con las fases del Proyecto Global.

Primera Fase, que corresponde al tema de la presente Memoria.

1. Cumplimiento de requisitos funcionales de Primera Prioridad (integración con sistemas de la primera Fase)
2. Cumplimiento completo de requisitos No funcionales
3. Coexistencia con sistemas actuales de Fonasa
4. Entregable completo y puesta en producción sincronizada con resto de los sistemas de la primera Etapa

Segunda Fase (que no forma parte de esta Memoria).

1. Cumplimiento de Requisitos Funcionales adicionales
2. Crecimiento natural hacia Sistema de Recursos Humanos

3.5 Resultados del Levantamiento de Requisitos

3.5.1. Modelo Gestión de Usuarios

Los usuarios se agrupan en diferentes categorías. Existirán usuarios a nivel central así como regional. Cada usuario tiene definido un cargo. Los usuarios tienen asociadas funciones o tareas de acuerdo a sus cargos, y deben adquirir permisos según las tareas que deban realizar.

Los cargos no determinan unívocamente las funciones, pues también, según las dependencias donde desarrollen su trabajo, en algunos casos un mismo cargo puede desarrollar más funciones. También, de acuerdo al período del año o por contingencias (es decir, vacaciones, reemplazos, etc), un funcionario puede necesitar reemplazar las funciones de algún compañero.

La asignación de tareas (y por ende los permisos) es descentralizada, de manera que el sistema debe ser capaz de permitir la delegación de la función de otorgamiento de permisos.

Lo anterior nos lleva a que las funciones o tareas deberán recibir una primera clasificación global, en cuanto a si son funciones propias del negocio o funciones asociadas directamente al sistema gestión de usuarios.

En este punto es conveniente separar el concepto de cargos del concepto de permisos. Para ello, tal como lo hemos señalado anteriormente, definiremos roles, como una abstracción más genérica, flexible y transversal. En otras palabras, haremos uso del patrón *RBAC (Rol-Based Access Control)*.

En definitiva, las directrices del modelamiento fueron:

- ▷ Definición de roles
- ▷ Segmentación de usuarios de acuerdo a roles
- ▷ Definición de permisos y segmentación en funciones asociadas al negocio (*recursos*) y funciones asociadas a la gestión de usuarios
- ▷ Definición de permisos (acceso a recursos) asociados a roles
- ▷ Definición de mecanismos de otorgamiento de permisos asociados a roles
- ▷ Definición de procedimientos para creación de usuarios y cuentas.
- ▷ Mapeo entre permisos albergados en el LDAP y su asociación con las funciones definidas al interior de los diferentes sistemas y plataformas (los recursos)

3.5.2. Casos de Uso

1. CRUD usuarios/cuentas (creación usuario, cese de funciones de usuario, creación/cese cuentas)
2. Desbloqueo de contraseñas
3. Administración de listas de owners de recursos
4. Solicitud de asignación roles/permisos
5. Autorización/rechazo de solicitud roles/permisos
6. Autoatención
7. Servicio Autenticación
8. Servicio Autorización
9. Otros Servicios
10. Generación de Estadísticas y Reportes
11. Migración / Carga inicial

En la figura 3.5 observamos el diagrama UML de casos de uso del sistema Gestión de Usuarios.

3.6 Plan de Trabajo

Para llevar a cabo exitosamente el desarrollo del Sistema Gestión de Usuarios se elaboró un plan de Trabajo, que se expone a continuación.

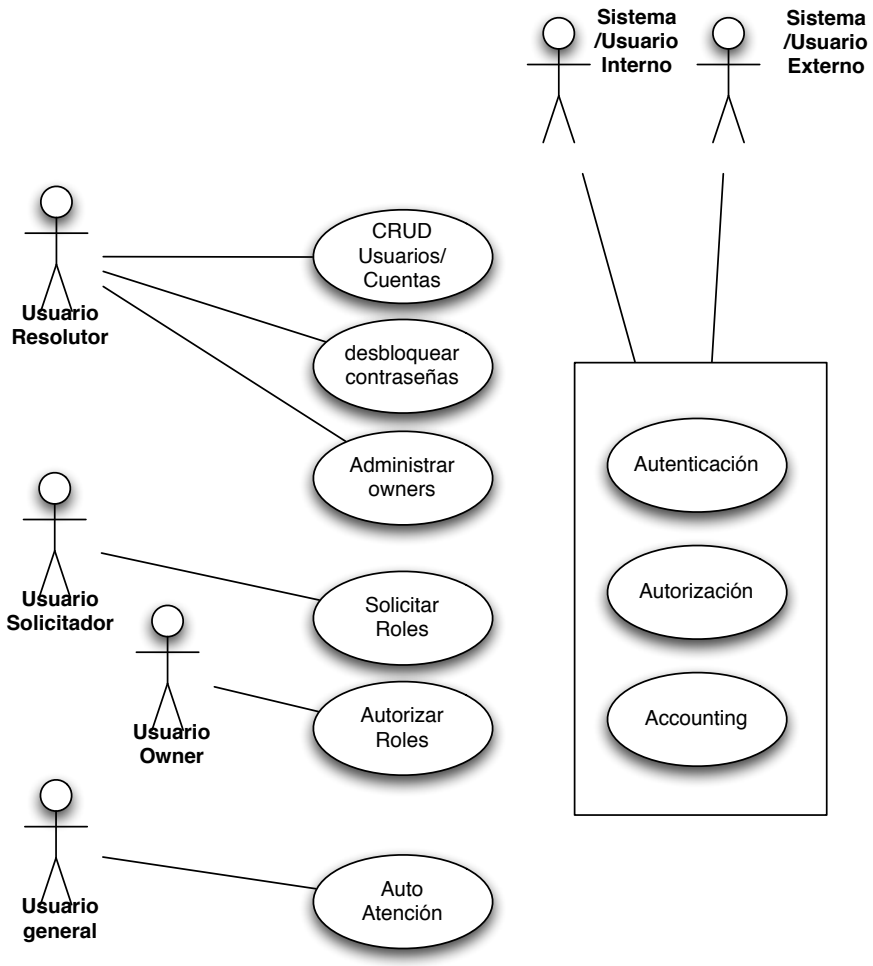


Figura 3.5: Casos de Uso

3.6.1. Elementos del Plan

1. Informar y establecer Metodología de Control de Proyecto [10]
2. Establecimiento de acuerdos de procedimientos (desarrollo, pruebas, aprobaciones, paso a producción, etc)
3. Apoyo en Gestión del Cambio
4. Levantamiento detallado de requerimientos (Modelamiento y establecimiento de acuerdos con los usuarios)
5. Análisis y selección de alternativas de LDAP
6. Construcción: Adaptación/Desarrollo
7. Diseño y ejecución de pruebas funcionales (Unitarias, Integradas)
8. Diseño y ejecución de pruebas No funcionales
9. Migración de datos
10. Marcha blanca y producción

3.6.2. Descripción Elementos del Plan y definición de Hitos y Entregables

1. Metodología
   - ▷ Revisión de metodología global y establecimiento de acuerdos específicos para el desarrollo del proyecto Gestión de Usuarios
   - ▷ Definición de usuarios y líderes estratégicos para sistema Gestión de Usuarios
   - ▷ Definición de hitos y entregables, mecanismos de certificación y de aprobación
2. Acuerdos de Procedimientos
   - ▷ Definición específica de mecanismos, procedimientos y formatos
   - ▷ Acuerdos para revisiones periódicas y estados de avance
3. Apoyo en Gestión del Cambio
   - ▷ Definición de alcance y grado de participación
   - ▷ Definición de compromisos
4. Levantamiento de Requerimientos
   - ▷ Análisis detallado (modelo general y específico, tipos de usuarios, niveles y permisos, funciones, mecanismos de altas y bajas, mecanismos de reasignación de funciones, niveles centrales y distribuidos de autorizaciones, etc).
   - ▷ Levantamiento de reglas de negocio asociadas a los procesos y actividades de Gestión de Usuarios
   - ▷ Especificación detallada y documentación
   - ▷ Ajustes y modificaciones
   - ▷ Aprobaciones formales

[10] La Metodología de Control de Proyecto está definida al origen del proyecto Fonasa, y se informa como parte de cada subproyecto, dado que aparecen usuarios adicionales en cada tema.

5. Análisis y selección de Alternativas de LDAP
    - ▷ Análisis de características
    - ▷ Nivel de adherencia al estándar LDAP
    - ▷ Evaluación de ventajas y desventajas
6. Construcción
    - ▷ Definición de Modelo de Datos
    - ▷ Definición de necesidades de Construcción/Adaptación
    - ▷ Definición de necesidades de Integración y generación de especificaciones en términos de formatos y restricciones para uso de protocolo LDAP entre las aplicaciones
    - ▷ Definición de mecanismos de Paso a Test y Producción
7. Diseño y ejecución de Pruebas
    - ▷ Definición de Plan de Pruebas
    - ▷ Definición de escenarios de Pruebas (operación normal, en contingencia, mecanismos de restauración)
    - ▷ Definición de casos de Prueba
    - ▷ Especificación de mecanismos de medición
    - ▷ Definición de criterios de aceptación y márgenes de tolerancia
    - ▷ Pruebas funcionales unitarias
    - ▷ Pruebas funcionales de sistema
    - ▷ Pruebas funcionales integradas (comunicación con otros sistemas/plataformas)
    - ▷ Pruebas no funcionales (Rendimiento, disponibilidad, seguridad, etc)
8. Migración de Datos
    - ▷ Definición de mecanismos de recepción de la data actual
    - ▷ Definición de requerimientos de validación y transformación de la data
    - ▷ Definición de procedimientos y mecanismos de carga y chequeo de resultados (calidad y completitud)
    - ▷ Acuerdos de niveles de éxito/error objetivos para cumplimiento y aceptación
9. Marcha Blanca y Producción
    - ▷ Definición de estrategia (Piloto - Big Bang)
    - ▷ Plan de puesta en marcha
    - ▷ Opción Piloto - Manejo coexistencia sistemas antiguos/nuevos
    - ▷ Mecanismos de roll-back ante fallas
    - ▷ Plan detallado

3.6.3. Cronograma

En la tabla 3.1 podemos ver el cronograma de este plan de trabajo.

Etapa	Descripción	Plazo Estimado
Detalle de Requerimientos	Especificación detallada y aprobaciones formales. Definición Apoyo Plan Gestión del Cambio	1 mes
Construcción	Definición de modelo de datos, especificación de formatos, Prototipos y adaptación/customización	1 mes
Pruebas y Ajustes	Unitarias, de sistema, integradas y pruebas No funcionales	1.5 mes
Migración de Datos	Definición de reglas, desarrollo aplicaciones de migración y pruebas	1 mes (paralelo)
Documentación	Usuario, Administración y Tema de Memoria	durante proyecto
Implantación	Marcha Blanca y puesta en producción (piloto, big-bang)	1 mes

Tabla 3.1: Cronograma del Proyecto

Capítulo 4

Solución

Como ya se ha mencionado, Fonasa se encuentra llevando a cabo un proyecto de cambio de todas sus plataformas de servicio. Este es un proyecto de alto impacto en la organización y afecta no sólo los sistemas computacionales, sino también sus procesos y procedimientos.

La Gestión de Usuarios, tema de la presente Memoria, es uno de los sistemas que se encuentran dentro de la nómina de aplicaciones que Fonasa desea modernizar. Al cambiar todas sus plataformas, surge en forma natural la necesidad de un nuevo sistema de gestión y administración de los usuarios de Fonasa, que por un lado unifique los diferentes mecanismos de autenticación que existían, y por otro que asegure la integración y consistencia con las nuevas plataformas.

En el capítulo anterior se describieron los requerimientos asociados. Corresponde ahora explicar en detalle la solución adoptada para el desarrollo e implementación del nuevo sistema Gestión de Usuarios de Fonasa.

Para ello, en primer lugar se explica la estructura, metodología y visión global de la evolución del proyecto, y luego se profundiza en los aspectos más técnicos de la solución.

4.1 El Proyecto

Para llevar a cabo su proyecto Fonasa llamó a una licitación en la que considera el cambio y modernización de todas sus plataformas. La empresa que se adjudicó este proyecto es Adexus S.A. Adexus es una empresa chilena, de larga trayectoria en los ámbitos de outsorcing, desarrollo y servicios de software.

Adexus enfrentó este proyecto tanto con recursos propios como con apoyo de una variedad de empresas consultoras.

Lo propio realizó Fonasa, que también utilizó recursos internos designados para este proyecto, así como profesionales y empresas consultoras externas.

En definitiva, el Proyecto Global contempla tanto los desarrollos de las aplicaciones como el outsorcing (explotación, housing, soporte, etc.) de los servicios asociados.

4.1.1. Organización del Proyecto Global

En la figura 4.1 se observa una diagrama esquemático de la estructura orgánica del Proyecto Global.

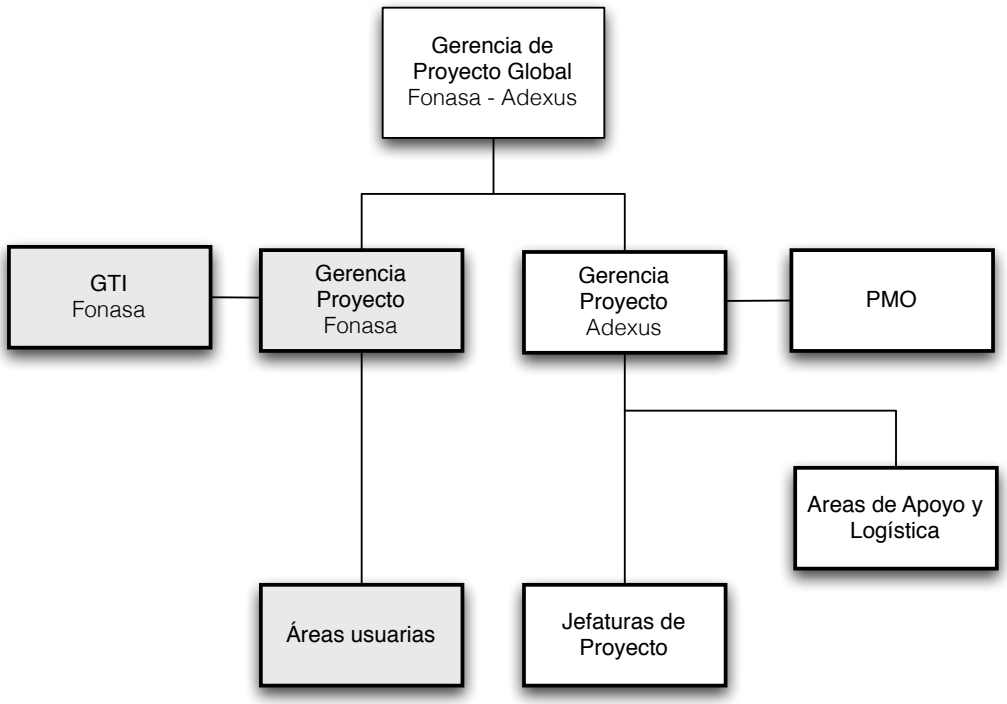


Figura 4.1: organización del proyecto

Una descripción de esta estructura es la siguiente:

- ▷ Gerencia de Proyecto (Fonasa-Adexus)
  Grupo ejecutivo principal del proyecto. Está conformado por Fonasa y Adexus. Corresponde al más alto nivel de decisiones y acuerdos.

- ▷ Fonasa
  - Gerencia de Proyecto Fonasa.
    Nivel ejecutivo principal para el proyecto, de parte de Fonasa.
  - GTI
    Area dedicada a la Gestión de Tecnologías de Información. Sus funciones son muy diversas. En el ámbito de este proyecto efectúan tareas de staff hacia la Gerencia de Proyecto Fonasa, labores en los diferentes proyectos específicos, así como apoyo a las áreas usuarias en cuanto a su interacción con las Jefaturas de Proyecto Adexus.
  - Areas usuarias.
    Son los grupos de usuarios estratégicos de las diferentes áreas de Fonasa. Pertenecen a recursos humanos, operaciones, áreas comerciales, finanzas, etc. Se agrupan de acuerdo a los proyectos específicos en los que les corresponde participar. Se relacionan fuertemente con las Jefaturas de Proyecto de su contraparte Adexus. Sus tareas principales corresponden a la definición de requerimientos, proporcionar información y antecedentes relacionados con la materia a resolver y efectuar las aprobaciones de los proyectos específicos.

- ▷ Adexus
  - Gerencia de Proyecto Adexus.
    Nivel ejecutivo principal para el proyecto, por parte de Adexus.
  - PMO
    Oficina de Control de Proyectos.
  - Jefaturas de Proyecto
    Su responsabilidad es llevar a cabo correctamente en plazos y costos las tareas de desarrollo de su proyecto específico.
  - Areas de Apoyo y Logística.
    Diferentes áreas necesarias para el normal desenvolvimiento del Proyecto Global, así como los diferentes proyectos específicos. Esto es, Arquitectura, Comunicaciones, Infraestructura, Explotación, Capacitación, Mesa de Ayuda, entre otros.

Mi rol en esta organización correspondió, por un lado, a la Jefatura de Proyecto Gestión de Usuarios y por otro, la consultoría y coordinación de las PNF [1] globales (Gestión de Usuarios y resto de las plataformas).

[1] Pruebas No Funcionales

4.1.2. Oficina de Administración de Proyectos (PMO)

La Administración del Proyecto Global así como de todos los proyectos estuvo bajo la responsabilidad de la PMO. Este es un grupo de staff dedicado a la recopilación, control, seguimiento y gestión de los proyectos en desarrollo, así como el reporte periódico de su estado hacia los niveles más altos de la organización.

Cada Jefe de Proyecto fue responsable de reportar el estado de avance y la gestión de su proyecto específico hacia la PMO. En particular, para el caso de Gestión de Usuarios se generaban dos reportes semanales, en que se indicaba el estado de avance de la carta Gantt y un reporte de novedades y riesgos. Además, una vez al mes se generaba información consolidada con toda la documentación (minutas, acuerdos e informes) para su registro y formalización como estado de avance mensual, así como con el estado de cumplimiento de hitos acordado en la respectiva Gantt.

4.1.3. Metodología en Gestión de Usuarios

CMMI como modelo de buenas prácticas

El Proyecto Global, así como cada uno de los subproyectos (entre los que se encuentra Gestión de Usuarios) fueron abordados utilizando las recomendaciones y buenas prácticas del modelo CMMI [2]. Aunque CMMI no es una metodología, establece un conjunto de elementos y objetivos que deben ser considerados en los procesos de desarrollo, mantención y operación de software. CMMI también persigue clasificar las organizaciones según su nivel de "madurez", esto es, el grado de evolución y calidad en que se encuentran dichos procesos.

Desde el punto de vista de buenas prácticas, CMMI pone especial énfasis en:

- ▷ gestión de requisitos, planificación y control del proyecto
- ▷ enfoque en el proceso, la coordinación entre grupos y el peer-review
- ▷ gestión cuantitativa y de calidad del software
- ▷ prevención de fallos y políticas de gestión de cambio del proceso y del software

Proceso Unificado

Aunque se utilizaron elementos comunes a diferentes enfoques, el marco metodológico más cercano a lo aplicado en el proyecto es el conocido como *Proceso Unificado* [3]. Esta metodología se caracteriza por la utilización de UML, la orientación a la arquitectura y el desarrollo incremental.

[2] Capability Maturity Model Integration

[3] Una de las implementaciones más populares del proceso unificado es RUP de IBM.

En el Proceso Unificado cada etapa del desarrollo se divide en cuatro fases:

- ▷ inicio
- ▷ elaboración
- ▷ construcción
- ▷ transición

Aunque no es exacto y también existe superposición de actividades, una buena aproximación es asimilar estar cuatro fases a las actividades de levantamiento de requerimientos, diseño, construcción y puesta en producción.

Directrices

En cuanto a las directrices para el desarrollo de Gestión de Usuarios, estas fueron las siguientes:

- ▷ Security Oriented
  Orientación hacia un estilo de arquitectura de solución, así como atributos de calidad que garanticen el cumplimiento de los procesos de seguridad.
- ▷ UML
  Utlización de UML como medio de intercambio y comunicación de información.
- ▷ Process Oriented
  Análisis y refinamiento de los procesos de negocio para comprender los requerimientos del sistema.
- ▷ Guiado por los Casos de Uso
  Establecimiento de casos de uso como formalización de requerimientos, definición de las necesidades de desarrollo y formulación del plan de pruebas.

Procedimientos

En el marco del proyecto, como procedimientos habituales se efectuaron las siguientes actividades:

- ▷ Reuniones de trabajo (usuario y técnicas).
- ▷ Reuniones de control de avance
- ▷ Revisiones de estado del proyecto y gestión de riesgos
- ▷ Presentaciones del aplicativo
- ▷ Sesiones de capacitación a usuarios (GTI, grupos de usuarios de diversas áreas, Help-Desk)
- ▷ Documentación y aprobación formal para cada actividad (minutas, acuerdos, cumplimiento de hitos)

4.1.4. El proyecto Gestión de Usuarios

Veamos ahora las etapas y actividades efectuadas durante el proyecto.

Inicio del Proyecto Global

- ▷ Se establecen directrices y objetivos globales.
- ▷ Se diseña el Plan Global (Gantt).
- ▷ Se define y difunde el Plan y la Metodología a utilizar.
- ▷ Se da inicio a cada uno de los proyectos asociados.

Primera Etapa del proyecto Gestión de Usuarios: Inicio

- ▷ Reforzamiento de la Metodología y acuerdos específicos de procedimientos
- ▷ Definición de los Equipos de Trabajo
- ▷ Levantamiento detallado y establecimiento de acuerdos de alcance del proyecto

El proyecto original de Gestión de Usuarios se centra en dos aspectos. Seguridad (usuarios) en la operación de las plataformas y luego los flujos de RRHH. Durante esta etapa se logró el acuerdo de dividir el proyecto en dos fases. En la primera fase cubrir la funcionalidad fundamental de la gestión de los usuarios y su interacción coordinada con los demás proyectos de la fase, y luego, en una segunda fase ampliar los desarrollos para considerar los flujos de RRHH.

Este acuerdo permitió sincronizar los desarrollos con el resto de los proyectos, cumplir con la salida a producción en conjunto (piloto y puesta en marcha) y llevar a cabo un plan de pruebas mucho más exhaustivo.

Adicionalmente, se confeccionó el detalle del cronograma (Gantt) del proyecto Gestión de Usuarios y se obtuvo el acuerdo formal de las partes.

Desarrollo

- ▷ Análisis y diseño
- ▷ Elaboración de prototipos
- ▷ Presentaciones a usuarios
- ▷ Ajustes
- ▷ Presentación final
- ▷ Pruebas funcionales unitarias internas

En esta etapa se realizó el análisis de detalle y el diseño de la solución. Además, y como consecuencia, se definieron y ejecutaron los requerimientos de configuración e implementación. Estas tareas requieren el concurso de diversas áreas. Los especialistas IBM para validar los diseños y varias tareas asociadas con Tivoli, el área de Arquitectura para ajustar los diseños, acuerdos con el resto de las equipos de desarrollo, la gente de Infraestructura para las instalaciones de software y los deployment de versiones, etc.

Durante esta etapa se efectuaron frecuentes reuniones de trabajo, internas y con usuarios, en orden de avanzar en los desarrollos, formalizar los acuerdos de detalle, y validar anticipadamente el avance de los trabajos. Las tareas son fundamentalmente iterativas y crecientes. Se avanza en las configuraciones, se verifica la calidad de lo desarrollado hasta ese punto, se revisa en conjunto con el usuario, surgen ajustes y cambios y el ciclo se reitera una y otra vez hasta obtener los resultados deseados.

Pruebas con usuario

- ▷ Pruebas funcionales unitarias
- ▷ Pruebas integradas
- ▷ Pruebas No Funcionales

En esta etapa, como primera medida se establecieron los acuerdos de estrategias de pruebas para obtener la certificación de los sistemas. Posteriormente se realizaron tareas internas de certificación y, una vez asegurada la calidad, concurrir ante el usuario para obtener la certificación oficial.

Es importante destacar que las pruebas no se limitaron a revisar los resultados obtenidos en una pantalla, sino muchas veces implicaron revisiones en logs, archivos planos y diversos otros mecanismos. Esto es, la calidad de las pruebas y los resultados obtenidos se deben revisar tanto en los puntos de interacción (pantallas, listados) como en los efectos internos que producen (persistencia, logs, archivos, mediciones de uso de recursos, etc).

Migración de datos

En esta etapa se definen las necesidades de migración de información, se establecen acuerdos y criterios para la recepción y validación de los datos y se realizan las tareas de diseño e implementación del software de migración.

Las fuentes de información son múltiples: Bases de datos históricas, sistemas actualmente en operación, nóminas no digitalizadas, etc. Las tareas se realizaron en conjunto con el área usuaria y en estrecha colaboración. Esto es, la recopilación y ordenamiento de la información, la resolución de inconsistencias, y el establecimiento de los niveles de acierto/error “aceptables” que permitieron, en última instancia, considerar el proceso como exitoso.

Piloto y Puesta en Marcha

Las tareas fueron:

- ▷ Definición de estrategia y plan.
  Entre las posibles estrategias, inicio de operación inmediato en todo el país (*Big-Bang*) o una opción más conservadora, inicio de operación escalonado a través de un Piloto, se optó por esta última alternativa. En este escenario se elaboró un plan detallado de trabajo y se definieron las tareas, recursos, prerrequisitos, plazos y necesidades de coordinación entre las áreas.
- ▷ Definición de mecanismos de "vuelta atrás".
  Un mecanismo de “vuelta atrás” es un procedimiento (tareas y responsables de ejecutarlas) que establece cuáles son los niveles de tolerancia y criterios para tomar una decisión de abortar una puesta en producción y cómo debe realizarse. Esto es, quién o quienes están autorizados a tomar dicha decisión, el tiempo estimado total en realizar dicha operación, los requisitos previos y recursos humanos y materiales a utilizar durante la vuelta atrás, los pasos requeridos, los mecanismos a aplicar en cada paso (manuales y automáticos), la descripción del proceso de restauración de datos y software, los mecanismos y puntos de control, el chequeo final de vuelta atrás exitoso, y el reinicio de la operación en las condiciones anteriores a la puesta en producción.

En esta etapa se definió la estrategia de inicio de operación de los sistemas. Como se indicó, el acuerdo consistió en llevar a cabo un Piloto con un conjunto de oficinas representativas y posteriormente la incorporación paulatina del resto de las oficinas.

También se definieron actividades previas al Piloto (y a la subida de cada nueva oficina) y actividades a efectuar durante el Piloto propiamente tal. Dentro de las tareas relevantes se estableció el acuerdo de efectuar labores de capacitación a las áreas usuarias, a través del entrenamiento de monitores y con el apoyo de manuales y material audiovisual.

Es relevante señalar que el Piloto así como la Puesta en Marcha son actividades en que todos los sistemas inician en forma única y conjunta su operación. Por ello la planificación y ejecución de estas tareas requiere no sólo la participación activa de los usuarios, sino también una estrecha coordinación con las áreas de apoyo (Arquitectura, Infraestructura, Mesa de Ayuda, etc.) y especialmente con el resto de los grupos de desarrollo.

4.2 Estrategia de Solución

Cumpliendo con el plan de trabajo, en este punto corresponde llevar a cabo las tareas necesarias para lograr la implementación exitosa del sistema.

Para ello se utiliza la información obtenida en la etapa de Levantamiento (Ingeniería de Requisitos) y se procede con las siguientes actividades:

- ▷ Análisis de alternativas de solución
- ▷ Diseño Arquitectura de la solución
- ▷ Diseño detallado
- ▷ Implementación
- ▷ Apoyo a la Gestión del Cambio

4.2.1. Uso de Tecnologías

Hoy en día resulta frecuente el uso de *suites* de software para dar solución a los requerimientos sistémicos de diferentes empresas y sectores de la Industria. En este contexto, la utilización de una suite para el caso de Fonasa no resulta extraña. En este mismo sentido, el uso de software de tipo Commercial Off The Shelf (COTS) para componer partes de sistemas también resulta coherente.

Por esta razón, además del desarrollo directo, una de las alternativas de solución para el sistema Gestión de Usuarios es la utilización de un COTS.

4.2.2. Alternativas de Solución

El desarrollo e implantación del sistema de Gestión de Usuarios de Fonasa no está exento de riesgos. La alternativa escogida debe cumplir con todos los requisitos tanto funcionales como no funcionales que están planteados, y a la vez, garantizar que podrá seguir creciendo para absorber nuevas funcionalidades. Además debe ser capaz de albergar las integraciones con los futuros desarrollos que se harán dentro del marco global del Proyecto Fonasa.

Es decir, se trata de un sistema que debe:

- ▷ satisfacer sus especificaciones funcionales
- ▷ ser capaz de evolucionar y sobrevivir en un ambiente integrado y complejo de múltiples aplicaciones y plataformas interactuando
- ▷ proporcionar altísimos niveles de disponibilidad y tolerancia a fallas

El diseño debe contemplar diferentes escenarios de contingencia, y sus atributos de resiliencia deben cubrir tanto aspectos de fallas propias como de las plataformas y sistemas

usuarias.

Además, cualquiera sea la solución adoptada, ésta marca estructuralmente toda la arquitectura de la solución completa y de las futuras aplicaciones que se agreguen a la plataforma.

Por ello, el correcto análisis de ventajas y desventajas y la evaluación ponderada de los trade-off que naturalmente surgen en las decisiones de diseño globales, resulta ser una necesidad insoslayable en este proyecto.

En los párrafos que siguen se delinean las principales alternativas consideradas y los criterios de análisis y evaluación utilizados.

4.3 Desarrollo versus COTS

4.3.1. Introducción

Para la solución del sistema Gestión de Usuarios de Fonasa se consideraron diferentes alternativas. Como primera aproximación, se analizaron las opciones:

- ▷ Desarrollo de un sistema en forma tradicional (construcción a la medida)
- ▷ Utilización de COTS (Comercial Off The Shelf), específicamente un LDAP

Y ante la opción de COTS, un segundo nivel de decisión obliga a elegir cuál de las COTS disponibles resulta más apropiada.

4.3.2. Criterios de Evaluación

- ▷ Costos y plazos de desarrollo/implantación
- ▷ Impacto en los sistemas actuales de Fonasa (período de convivencia de sistemas legacy con nuevos sistemas)
- ▷ Impacto en los nuevos desarrollos (del Proyecto Global)
- ▷ Existencia de estándares en la Industria
- ▷ Disponibilidad de herramientas
- ▷ Facilidad de adherencia a principios y restricciones definidos por el sistema y los estándares existentes

4.3.3. Mecanismo de Análisis de Alternativas

Desarrollo de un Sistema ad-hoc

- ▷ Escenarios de resultados obtenidos vía un desarrollo tradicional
- ▷ Grado de cumplimiento con los criterios de evaluación
- ▷ Ventajas y Desventajas

Utilización de COTS

- ▷ Estudio de algunos LDAP disponibles. Específicamente, analizar una alternativa comercial, el LDAP Tivoli de la suite IBM y la opción open source, OpenLDAP.
- ▷ Análisis de características y comparación de ambas herramientas
- ▷ Grado de cumplimiento con los criterios de evaluación
- ▷ Ventajas y desventajas

4.3.4. Desarrollo versus COTS

Análisis Alternativas

Desarrollo (Sistema ad-hoc)	
Ventajas	• Diseño ajustado y guiado por los requerimientos
Desventajas	• Construcción "desde cero" • Extenso plan de pruebas con posiblemente intensivos niveles de depuración • Plazo de desarrollo puede ser muy extenso. Riesgos de cumplimiento.

Tabla 4.1: Alternativa vía desarrollo

Para el caso de COTS, el principal requisito es la necesidad de adaptación de los diferentes Sistemas al protocolo LDAP.

COTS (LDAP)	
Ventajas	• Utilización de componentes con funcionalidad "probada" • Utilización de estándares a nivel Industria • Disponibilidad de herramientas y componentes con adherencia al estándar
Desventajas	• Requiere adaptación de todos los sistemas al protocolo LDAP y mantener un alto nivel de coordinación y comunicación entre los equipos de desarrollo. • Necesidad de difusión de nuevos estándares de construcción de aplicaciones

Tabla 4.2: Alternativa vía COTS

Alternativa elegida

Considerando los antecedentes expuestos, la decisión adoptada es usar un LDAP.

4.3.5. COTS - OpenLDAP e IBM Tivoli

Para la solución del sistema Gestión de Usuarios de Fonasa utilizando COTS, se analizaron las opciones OpenLDAP y la suite IBM Tivoli.

En los párrafos que siguen se exponen las características de ambas suites, y el análisis comparativo que llevó a la decisión final de cuál fue la tecnología adoptada para llevar a cabo el proyecto.

4.4 OpenLDAP

OpenLDAP es una iniciativa de la OpenLDAP Foundation, del año 1998, cuyo objetivo es proveer a la comunidad con una suite open source, multiplataforma, robusta, de alto nivel, con aplicaciones y herramientas LDAP, full compliance y actualizada a las últimas especificaciones y definiciones del protocolo. A la fecha, OpenLDAP se encuentra en la versión OpenLDAP 2 y satisface completamente la especificación LDAPv3.

OpenLDAP corre en diferentes plataformas (Windows, Linux, Unix) y puede utilizar diferentes backends, entre los que se incluye BDB (Oracle Berkeley database), SQL database, archivos planos (en formato LDIF), o incluso otro directorio LDAP.

4.4.1. Estructura

La suite OpenLDAP [4] tiene cuatro componentes:

- ▷ Server, que provee servicios LDAP
- ▷ Clientes, que permiten manipular la data LDAP, a través del server
- ▷ Utilitarios, que permiten interactuar directamente con la data y metadata [5]
- ▷ Librerías, que proveen APIs (Application Program Interface) orientadas a lenguajes de programación

Server

El server de OpenLDAP se denomina SLAPD (Stand-Alone LDAP Daemon) y provee acceso a la información contenida por uno o más directorios.

Los servicios ofrecidos son autenticación, búsqueda y manipulación de los datos del(los) directorio(s).

El server puede almacenar localmente la información de directorio o bien accederla (proxy access) en fuentes externas.

Clientes OpenLDAP

Los clientes OpenLDAP accesan el server SLAPD a través de un protocolo de red LDAP, y le solicitan efectuar operaciones sobre el(los) directorios. El procedimiento habitual es que el

[4] Butcher [3, part 1]
[5] datos de configuraciones, parámetros, etc.

cliente se conecta al server (binding), luego se autentica (sesión anónima o con credenciales), solicita efectuar operaciones (búsquedas, modificaciones, etc), y finalmente se desconecta.

Utilitarios

Los utilitarios OpenLDAP no utilizan el server, sino que permiten manipular directamente la data del directorio.

Librerías

Las librerías OpenLDAP proveen funcionalidad que es compartida y utilizada por las diferentes aplicaciones LDAP (server, clientes, utilitarios).

Además, proveen de APIs que permiten desarrollar aplicaciones o partes de aplicaciones que necesitan utilizar los servicios LDAP. Las APIs nativas de OpenLDAP están escritas en el lenguaje C, y además existen dos APIs para Java.

4.4.2. Seguridad

La información contenida por un directorio en OpenLDAP es claramente información sensible (pasword, información de los usuarios, etc). Para proteger la información OpenLDAP considera tres aspectos de seguridad. Estos son:

- ▷ Seguridad de la conexión de red
- ▷ Autenticación
- ▷ Autorización

Seguridad de la conexión de red

Corresponde a la protección de la información intercambiada entre los clientes y el servidor. Es decir, es la seguridad en la network (red) y la utilización de SSL y TLS.

El protocolo LDAP, por defecto, intercambia mensajería en texto no encriptado. Esto tiene ventajas desde el punto de vista de la configuración y la rapidez, pero naturalmente presenta riesgos de interceptación en una red (LAN/WAN), riesgos que son aún mayores si se trata de Internet.

OpenLDAP provee dos mecanismos para encriptar el tráfico de red.

- ▷ Utilización del puerto 636, el que por defecto encripta toda la información. Este método está consignado en la versión 2 del protocolo LDAP, pero en la versión 3 deja de ser la opción recomendada.

- ▷ Utilización de SSL/TLS. Esto forma parte de LDAPv3 y corresponde al uso de los mecanismos de seguridad Secure Sockets Layer (SSL) y Transport Layer Security (TLS). Se trata de dos protocolos criptográficos para intercambiar información sobre una red. Ambos protocolos son muy similares (de hecho, TLS es una evolución de SSL) y su funcionalidad se basa en dos aspectos: Establecer la autenticidad de la conexión sobre la red y la encriptación de la mensajería intercambiada.

Autenticación

La autenticación es efectuada a través de los mecanismos de binding. Estos pueden variar desde un binding anónimo, autenticación básica, autenticación usando SASL y/o SSL/TLS.

Autorización

OpenLDAP implementa mecanismos de autorización a través de ACLs. La sintaxis es del tipo *access to [resources] by [who] [parameters]*. Esto permite especificar los *recursos* que están restringidos (porciones del DIT, atributos determinados, etc) y "quiénes" los pueden acceder y en qué forma. Los who pueden estar definidos, por ejemplo, en base a atributos (dn, cn), usuarios anónimos o autenticados, grupos, etc.

4.4.3. Configuraciones Avanzadas

OpenLDAP provee algunas interesantes capacidades avanzadas. Entre ellas:

- ▷ multiple database backends. OpenLDAP puede administrar varios directorios simultáneamente, o particiones de directorios.
- ▷ Tuning . Es posible especificar parámetros de tiempos, tamaños, threads, etc, apuntando a obtener mejores rendimientos del LDAP.
- ▷ Directory overlays. Son una suerte de "subrutinas" que permiten agregar funcionalidad y/o cambiar el comportamiento del servidor. Por ejemplo, para modificar sutilmente la operatoria del log, para restringir globalmente el uso de ciertas operaciones, definir políticas de password, etc.
- ▷ Uniqueness constraints. LDAP por defecto exige unicidad en el atributo dn (distiguished name). Con esta directiva, OpenLDAP permite agregar restricciones de unicidad adicionales para otros atributos. Por ejemplo, puede resultar útil exigir unicidad en el campo uid (user identity) de las entries del directorio.

4.4.4. Replicación

OpenLDAP ofrece dos mecanismos de replicación de directorios:

- ▷ SLURPD
- ▷ LDAP Sync Replication

SLURPD

OpenLDAP soporta dos Daemon:

- ▷ SLAPD, del cual ya hemos mencionado sus funcionalidades y usos
- ▷ SLURPD, Stand-Alone LDAP Update Replication Daemon.
  SLURPD es un Daemon que proporciona un mecanismo único para mantener copias sincronizadas de directorios.

La forma de operar de SLURPD es muy sencilla:

- ▷ Opera en modo *master-slave*
- ▷ Define un *master SLAPD server*
- ▷ Define un conjunto de *slaves SLAPD servers*
- ▷ Mantiene un tracking de todas las operaciones efectuadas por el master
- ▷ Envía dichas operaciones a todos los slaves, para mantener la sincronización

LDAP Sync Replication

SLURPD aún es soportado, pero está declarado obsoleto. En su lugar, a contar de OpenLDAP 2.2, se promueve el uso de la funcionalidad denominada LDAP Sync Replication. En esta modalidad no se utiliza un segundo daemon, sino que el slave se conecta periódicamente al master y le solicita el envío de las modificaciones para efectuar la sincronización.

4.5 IBM Tivoli

IBM posee una suite de productos orientada a temas de seguridad. Permite administrar usuarios y cuentas, así como proteger y monitorear los accesos a diferentes recursos en un cierto entorno. Su principal característica es separar las diferentes funcionalidades en módulos configurables independientemente, pero con la capacidad de interoperar en forma natural y coherente. Esto constituye una gran fortaleza a la hora de customizar la suite para una realidad determinada, y también nos asegura la integración entre todos los elementos y sistemas que componen la plataforma.

La suite se denomina IBM Security Identity and Access Assurance (también conocida como IBM Tivoli Identity and Access Assurance) y está compuesta por una serie de módulos, donde destacan:

- ▷ Tivoli Directoy Server (TDS)
- ▷ Tivoli Identity Manager (TIM)
- ▷ Tivoli Access Manager (TAM)

4.5.1. IBM Tivoli Directory Server

TDS corresponde a la componente LDAP de la suite. Sus características son:

- ▷ Implementa la especificación LDAPv3
- ▷ disponibilidad en una amplia variedad de plataformas (Windows, Linux, Unix)
- ▷ fuerte integración con el resto de las componentes de la suite Tivoli

Componentes

- ▷ Base de datos IBM DB2, usada como backing store
- ▷ El server: idsslapd
- ▷ Herramientas para administrar y configurar el directorio
  - Web Administration Tool
  - Configuration Tool: GUI para configurar el directorio y la base de datos
  - Command line
- ▷ TDS SDK, que es un conjunto de herramientas para desarrollar aplicaciones LDAP (librerías, headers y ejemplos en lenguaje C)

Seguridad

Implementa los mecanismos del estándar LDAPv3. Es decir, todos los mecanismos de autenticación, y utilización de SASL y SSL/TLS.

En cuanto a la autorización provee poderosos mecanismos de ACLs (cuáles recursos, quiénes y qué tipo de acciones).

Configuraciones

TDS define una serie de configuraciones que apuntan a múltiples aspectos: Rendimiento, política de passwords y otros.

Replicación

Apuntando a la alta disponibilidad y escalabilidad, TDS provee varios mecanismos de replicación (dependiendo de la topología de los servidores LDAP) :

- ▷ Master - replica
- ▷ Cascading
- ▷ Peer-to-peer
- ▷ Gateway
- ▷ Distributed directory usando un directory proxy server

4.5.2. IBM Tivoli Identity Manager

TIM es un módulo específicamente diseñado para la administración de usuarios y cuentas.

Funciones

- ▷ permite modelar, entre otros, los patrones de autorización RBAC, DAC y MAC
- ▷ administra los siguientes elementos:
  - usuarios, cuentas y atributos
  - passwords
  - grupos
  - managed systems y applications (para conciliación de cuentas en otros sistemas y plataformas)

Para efectuar estas tareas [6] TIM define los siguientes mecanismos:

- ▷ organizational tree and roles
  Para definir la estructura de la organización.
- ▷ Identity Manager Groups and ACLs
  Para administración de permisos y atribuciones.
- ▷ Policy
  Administra cuatro tipos de policies (políticas de seguridad). Estas son provisioning policy, password policy, identity policy y service selection policy.
- ▷ Workflow
  Para administrar autorizaciones de cambios y permisos (por ejemplo, se puede configurar para solicitar autorización de cambios de roles de usuarios)
- ▷ Audit logs
- ▷ Reports
- ▷ Lifecycle management

[6] IBM [9, capítulo 18.2]

Componentes

- ▷ Web User Interface Layer
  - TIM Administración
  - Autoatención
- ▷ Application Layer
  El núcleo de funcionalidad de TIM.
  - Workflow
  - Administración de políticas (provisionamiento, password y otros)
  - Administración de entidades (identidades, usuarios)
  - Administración de cuentas
  - Módulo de configuración
  - Accounting
- ▷ Service Layer
  Son un conjunto de APIs y conectores para conciliación de cuentas entre sistemas, que permiten:
  - exponer servicios
  - conectarse con otras aplicaciones
  - efectuar el provisionamiento de cuentas en otros sistemas y plataformas de la infraestructura [7]
- ▷ Backend
  - LDAP based
  - Metadata

4.5.3. IBM Tivoli Access Manager (TAM)

TAM es un módulo de la suite que concentra las funciones de autenticación y autorización. Además proporciona agentes o servicios que son directamente utilizables por las diferentes aplicaciones de la organización.

Funciones

- ▷ permite definir las políticas de control de acceso a los recursos
- ▷ efectúa un control centralizado de autenticación de los usuarios
- ▷ efectúa un control centralizado de autorización de acceso de los usuarios hacia las aplicaciones y recursos

Para implementar las políticas de Autorización [8], TAM define los siguientes mecanismos:

[7] este punto es sumamente útil para una robusta implementación de single-sign-on
[8] IBM [9, capítulo 10.1]

- ▷ Access control list (ACL)
  Un ACL (lista de control de acceso) especifica acciones predefinidas que un conjunto de usuarios y grupos puede efectuar sobre un objeto. Por ejemplo, un grupo de usuarios que tiene acceso garantizado sobre un cierto objeto (una aplicación son los administradores de un sitio o de una determinada base de datos)
- ▷ Protected object policy (POP)
  Una POP (política para protección de objeto) especifica condiciones de acceso ligadas o propias de un objeto o recurso, y que aplican a todos los usuarios y grupos por igual. Por ejemplo, una POP sobre un objeto que indique que no puede ser accesado los días domingo (una aplicación de este concepto son los respaldos full de bases de datos, efectuados los fines de semana)
- ▷ Authorization rule
- ▷ Object space
- ▷ Resource manager

Una política de seguridad se construye como una combinación de

- ▷ ACLs
- ▷ POPs
- ▷ Authorization rules

En la práctica, la implementación del control de autorización se traduce en la configuración o parametrización de estos conceptos en TAM, de acuerdo a las necesidades de la organización.

Además, para efectuar el control centralizado de autenticación y autorización:

- ▷ Utiliza las definiciones de usuarios registradas en LDAP (u otros repositorios de autenticación)
- ▷ Utiliza las políticas de autorización según estén definidas
- ▷ Efectúa un registro de todas las actividades realizadas (Accounting)

Componentes

- ▷ User registry (LDAP)
- ▷ Authorization database (Metadata)
- ▷ Administración de la configuración vía comandos: pdadmin
- ▷ Web Portal Manager, para efectuar administración vía web
- ▷ Servicios de autenticación/autorización (publicables)

4.6 Análisis comparativo de COTS

Como ya se indicó, la primera decisión adoptada fue optar por la utilización de un COTS. Luego, en este último escenario se consideraron las opciones OpenLDAP e IBM Tivoli, cuyas características fueron expuestas en los párrafos precedentes.

La comparación se estableció sobre la base de que ambas suites presentan objetivos comunes y están orientadas a los mismos aspectos funcionales. Ambas suites proporcionan un núcleo de funcionalidad LDAP, mecanismos básicos estándar de manipulación de directorios LDAP, y software y funcionalidad adicional para efectuar operaciones más específicas.

En definitiva, la alternativa adoptada fue la utilización de IBM Tivoli. Entre las razones fundamentales se encuentra el grado de cobertura de esta suite respecto de los requerimientos de Fonasa. En los párrafos siguientes ahondaremos las razones de esta decisión.

4.6.1. Foco del análisis comparativo

En toda evaluación se deben considerar costos y beneficios de las alternativas. En materia de costos, si bien existen importantes diferencias entre ambas suites, las decisiones últimas fueron tomadas en los niveles más altos de la estructura organizacional del Proyecto Global. Por esta razón, el foco de este análisis comparativo fue eminentemente técnico.

No obstante, para completitud de este documento, mencionaremos los aspectos de costos que resultan relevantes a esta evaluación.

4.6.2. Estructura de Costos/Beneficios

En la tabla 4.3 se indican las componentes de costo/beneficio de este proyecto.

Costos	Beneficios
Licenciamiento	Minimización de riesgos
Mantención	Ingresos por outsorcing
Desarrollo de proyecto	Economías de escala
Soporte	
Costos por incumplimiento SLA	

Tabla 4.3: Costos y beneficios del proyecto

Licenciamiento y mantención

IBM Tivoli es una suite comercial, cuya estructura de costos se basa en el licenciamiento del producto y la mantención anual, la que da derecho a parches, correcciones y nuevas versiones del software (esto, por supuesto, no incluye el costo de las migraciones). Por su parte OpenLDAP es una suite open source y no presenta costos de licenciamiento ni mantención.

Desarrollo

En cuanto a los costos de desarrollo del Proyecto, están presentes en cualquiera de las opciones. No obstante es habitual que los costos de desarrollo y consultoría en herramientas comerciales sean más altos que en sus símiles open source.

Soporte

También, cualquiera sea la opción, existen costos de soporte asociados. Esto es, la posibilidad de recurrir a personal experto ante contingencias que no puedan ser resueltas internamente.

En el caso de IBM, esta corporación ofrece un servicio de soporte escalonado, con estructuras horarias definidas, y cuya cobertura está directamente relacionada con las necesidades del usuario y los niveles de costo que esté dispuesto a solventar.

Costos por incumplimiento de SLA y Minimización de riesgos

Un aspecto estratégico del tema costos es la minimización de riesgos, y tiene directa relación con las garantías del desarrollo y los niveles de servicio asociados al soporte. Un tema muy relevante, ya que Fonasa impone SLAs altamente exigentes, con multas asociadas a fallas y desviaciones en el cumplimiento del servicio (desarrollo y outsorcing). Considerando este hecho, resulta sumamente atractivo contar con un proveedor que exhiba una estructura y tamaño tal, que le permita colaborar y comprometerse también activamente en la provisión de dichos niveles de servicio.

Ingresos por outsorcing

Este punto está directamente relacionado con los anteriores. Existen ingresos por concepto de outsorcing que se pueden ver mermados por los incumplimientos de SLA, y que requieren la minimización de riesgos de incumplimiento.

Economías de escala

Adicionalmente, en el caso de IBM también se estaba negociando la utilización de otros productos de software en varios de los proyectos del Proyecto Global. Esto redunda en la obtención de economías de escala asociadas.

Dependencia del proveedor

Por cierto, un aspecto que no se puede soslayar es la dependencia del proveedor. En la medida que una empresa u organización comienza a hacer uso creciente de determinadas tecnologías, también se produce una dependencia del proveedor de dichas tecnologías, lo que puede redundar en un potencial riesgo ulterior. Aunque esto no es completamente evitable, una medida de mitigación importante es la transferencia de conocimientos o transferencia tecnológica. Por esto, como parte de la negociación se incluyó la exigencia a IBM de disponer de entrenamiento, tutoriales y otros recursos de capacitación para el personal interno de Adexus.

4.6.3. Criterios técnicos de comparación

Los criterios técnicos de comparación de ambas soluciones son los siguientes:

- ▷ Cobertura de la solución hacia los requerimientos del sistema
- ▷ Conocimiento experto en la tecnología
- ▷ Soporte

4.6.4. Aspectos funcionales

Veamos los cuadros comparativos de ambas alternativas. Primero los aspectos funcionales (tabla 4.4):

	Requisito Funcional	OpenLDAP	IBM Tivoli
1	Integración sistemas primera fase	vía servicios	vía servicios
2	CRUD Usuarios/Cuentas	construir	configurar TIM
	Desbloqueo contraseñas	construir	configurar TIM
	Administración owners	construir	configurar TIM
	Solicitud/Autorizacion Roles	construir	configurar TIM
	Autoatención	construir	configurar TIM
3	Servicios autenticación y autorización	construir	configurar TAM
4	Capacidades avanzadas (NO de esta etapa)	no aplica	no aplica
5	Migración de datos	construir	construir

Tabla 4.4: OpenLDAP vs IBM Tivoli. Funcionales

Integración

Tanto OpenLDAP como IBM Tivoli proveen apis que permiten efectuar la integración directa con los sistemas que lo requieren.

Funcionalidad usuarios

TIM provee dos sitios web (TIM Administración y Autotención) que a través de configuración y customización del look & feel pueden ser utilizados de inmediato en estas labores. En el caso de OpenLDAP existen desarrollos open source provistos por terceros, que permiten manipular la data del directorio. Sin embargo, en este último caso la necesidad de capacitación y la posibilidad de errores aumenta considerablemente. Por ello es requisito desarrollar mecanismos web que efectúen validaciones y proporcionen sólo aquella funcionalidad que es necesaria para los diferentes usuarios (administradores y usuarios normales).

Servicios de autenticación/autorización

TAM provee servicios de autenticación/autorización configurables y publicables para su uso. En el caso de OpenLDAP podrían utilizarse directamente los servicios que proporciona, sin embargo esto aumenta los niveles de riesgo de hacking de los datos del directorio (a través de técnicas de LDAP injection). Una buena práctica en este caso es anteponer una componente que filtre sólo aquellas queries que son válidas. Por esta razón, lo más adecuado es disponer de servicios de autenticación/autorización que medien entre LDAP y los sistemas que requieren esta información.

Otro aspecto relevante es que al disponer de servicios de autenticación/autorización se concentra en un único punto la lógica asociada. Esto significa que las plataformas simplemente deben invocar los servicios (pasando los parámetros adecuados) y son estos módulos los que

resolverán si una conexión es o no válida y si se tiene o no acceso a determinada funcionalidad. Además, en estos módulos se puede manejar el concepto de sesiones, tiempo de validez de las credenciales, excepciones a los accesos (por ejemplo, durante respaldos), etc., lo que simplifica la administración y contribuye a la encapsulación de funcionalidad.

Migración de datos

Tanto IBM Tivoli como OpenLDAP proporcionan servicios de carga masiva de datos hacia el directorio (a través de LDIF). No obstante, en cualquier caso es necesario construir software de validación y formateo que tome la data generada por Fonasa, genere informes de validación (errores y aciertos) y produzca como salida los registros válidos en formato LDIF, listos para su carga.

4.6.5. Aspectos No funcionales

Y los aspectos No funcionales en la tabla 4.5:

	Requisito No Funcional	OpenLDAP	IBM Tivoli
1	Rendimiento	no testeado	no testeado
2	Seguridad	cumple	cumple
3	Alta disponibilidad	replicación	replicación
4	Integridad	cumple	cumple
5	Interoperabilidad	publicación servicios	publicación servicios
5	Evolucionabilidad	vía diseño	via diseño

Tabla 4.5: OpenLDAP vs IBM Tivoli. No Funcionales

En los aspectos no funcionales tanto IBM Tivoli como OpenLDAP cumplen con el estándar LDAPv3. Los elementos de seguridad se encuentran cubiertos a través de la utilización de los mecanismos de comunicación proporcionados por el protocolo. Respecto de la alta disponibilidad, ambas suites permiten trabajar en ambientes de clustering y proporcionan herramientas para replicación de la información. En cuanto a la integridad de los datos, IBM utiliza como backing store la base de datos DB2, mientras que OpenLDAP utiliza preferentemente BDB (Oracle Berkeley database).

La interoperabilidad con los diferentes sistemas se logra a través de la publicación de servicios, los que son proporcionados por ambas plataformas.

En cuanto a la evolucionabilidad, más que una particularidad de IBM Tivoli o de OpenLDAP, es una característica del modelo de información del LDAP y está garantizada a través del diseño del directorio.

4.6.6. Otros aspectos

El resto de los elementos en análisis (tabla 4.6):

Otros	OpenLDAP	IBM Tivoli
Conocimiento experto	empresas de servicio	disponible en consultores IBM y partners
Soporte	empresas de servicio	proporcionado por la corporación IBM y partners

Tabla 4.6: OpenLDAP vs IBM Tivoli. Otros aspectos

Conocimiento experto

El conocimiento de la herramienta OpenLDAP está muy difundido, pero más que encontrar empresas especializadas o representantes del producto, se deben utilizar empresas de servicio que hayan tenido experiencia en desarrollos similares. En el caso de IBM Tivoli, el conocimiento está disponible en IBM y en las empresas partners de IBM.

Soporte

Lo propio ocurre con el soporte. Este es un aspecto relevante a considerar, dado que es necesario tanto desarrollar el sistema, como dar el servicio de outsorcing. En el caso de IBM Tivoli, existen alternativas de soporte con diferentes niveles de horario y cobertura.

4.6.7. Conclusiones

OpenLDAP vs IBM Tivoli. Por qué se adoptó la tecnología IBM.

En cuanto a los aspectos No funcionales ambas tecnologías son equivalentes. Donde existe la mayor diferencia es en los aspectos funcionales, donde IBM proporciona una serie de módulos configurables e interconectados, que requieren parametrización, versus la opción OpenLDAP en que estos requisitos se deben construir.

En lo que respecta al conocimiento experto y soporte, IBM provee dichos servicios directamente o a través de sus partners. En cambio, para el caso de OpenLDAP es necesario recurrir a empresas de servicio que se dediquen al tema.

Tal como se indicó, el foco de este análisis fue eminentemente técnico y cumple con proporcionar estos antecedentes para una toma de decisión en niveles más altos de la estructura

del Proyecto Global.

La toma de decisión, por supuesto, considera y pondera los diferentes elementos de que dispone (técnicos, costos/beneficios, estratégicos, etc). En definitiva, la alternativa elegida para llevar a cabo el proyecto Gestión de Usuarios fue IBM Tivoli.

4.7 Arquitectura

Como se ha dicho, la decisión adoptada para el desarrollo de este sistema es la utilización de la suite de productos de IBM Tivoli.

Veremos ahora la arquitectura global del sistema Gestión de Usuarios.

4.7.1. Introducción

¿Cómo se describen las decisiones de diseño adoptadas?

Los ADLs (Architecture Description Language) que utilizaremos para mostrar la arquitectura de este sistema son:

- ▷ UML
- ▷ Textual
- ▷ Diagrama gráfico informal (Visio, Omnigraffle)

UML resulta una alternativa adecuada, dado que es universalmente conocido y presenta Views específicas orientadas a mostrar aspectos muy concretos del sistema.

Respecto de los diagramas gráficos informales, permiten mostrar "visualmente", en forma rápida, aspectos globales del sistema, y resultan muy intuitivos para algunos stakeholders.

Los ViewPoint son:

- ▷ Funcional
- ▷ Información
- ▷ Desarrollo
- ▷ Deployment
- ▷ Operación

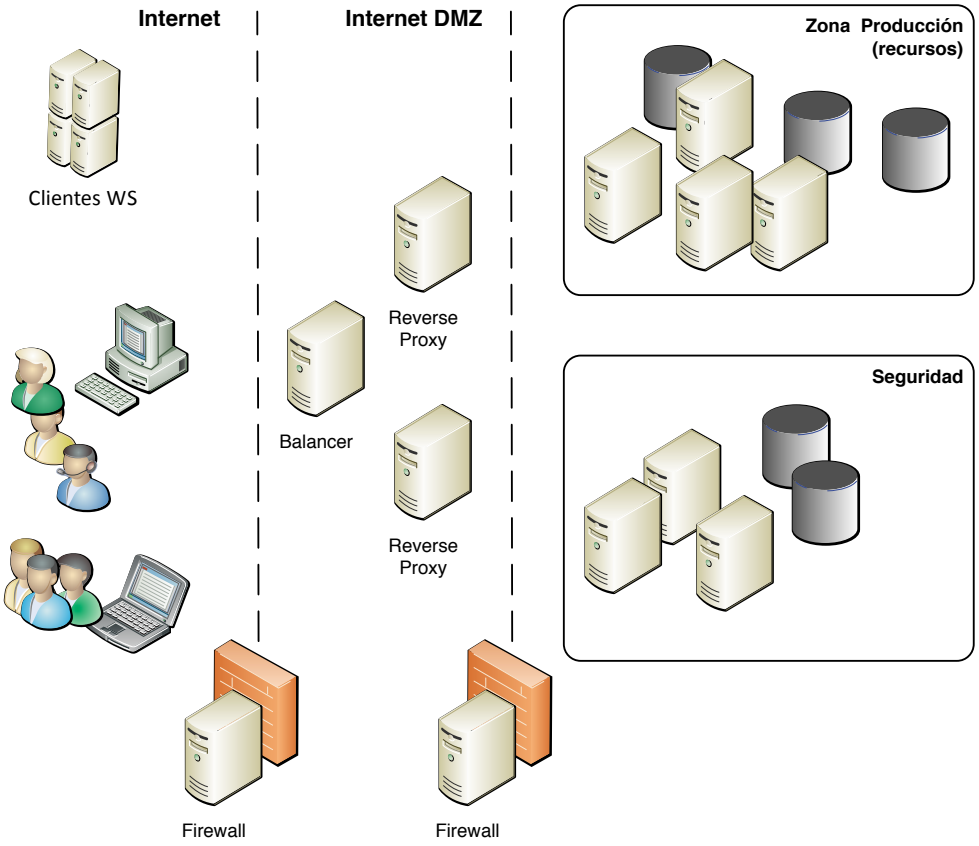


Figura 4.2: Diagrama de Arquitectura global

4.7.2. Diagrama Global

En la figura 4.2 se observa un diagrama global físico de la arquitectura y la ubicación de las áreas de Seguridad

Asimismo, en la figura 4.3 podemos ver una descripción lógica de las aplicaciones y su interacción. El sistema Gestión de Usuarios es parte integral de la funcionalidad de Seguridad.

Los servicios de autenticación y autorización pueden ser accedidos directamente desde el reverse proxy o a través del bus de servicios. La decisión depende de la implementación específica de cada una de las aplicaciones residentes en la zona de producción.

En el caso de utilizar el reverse proxy como punto de chequeo, estamos haciendo uso del patrón de autorización Check Point (Access Verification), que se refiere a tener un punto único de verificación de autenticación y autorización.

Si se utiliza el bus de servicios, una forma sencilla de verificación es utilizar un agente de intercepción que efectúe el chequeo de autenticación y autorización para cada solicitud de servicio que transite por el bus. En otras palabras, estamos haciendo uso del patrón Reference Monitor (Policy Enforcement Point).

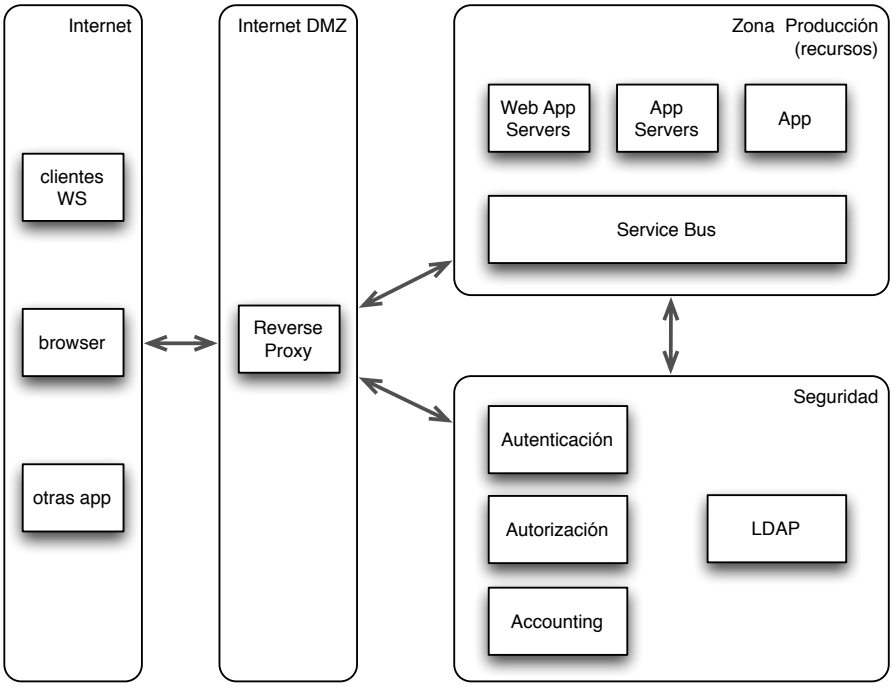


Figura 4.3: Diagrama lógico global

Para efectos de lograr el single-sign-on, los servicios de autenticación/autorización deben administrar un sistema de sesiones de usuarios, que permita manejar las credenciales y datos de seguridad necesarios para evitar la reconexión a cada nueva aplicación. Además se deben manejar políticas de tiempos máximos de duración de las sesiones y permisos. En otras palabras, utilizar el patrón Security Session.

La arquitectura global podría variar, pero desde el punto de vista del sistema Gestión de Usuarios, lo importante es que debe disponibilizar los servicios de autenticación y autorización.

4.7.3. Funcionalidad

En la figura 4.4 se observan las componentes del sistema Gestión de Usuarios.

Estas componentes se agrupan en los módulos de IBM TIvoli, de la siguiente forma:

- Tivoli Identity Manager (TIM)
  - CRUD usuarios y cuentas
  - Desbloqueo de contraseñas
  - Administración de owners
  - Solicitud de asignación de roles
  - Autorización de roles
  - Autoatención

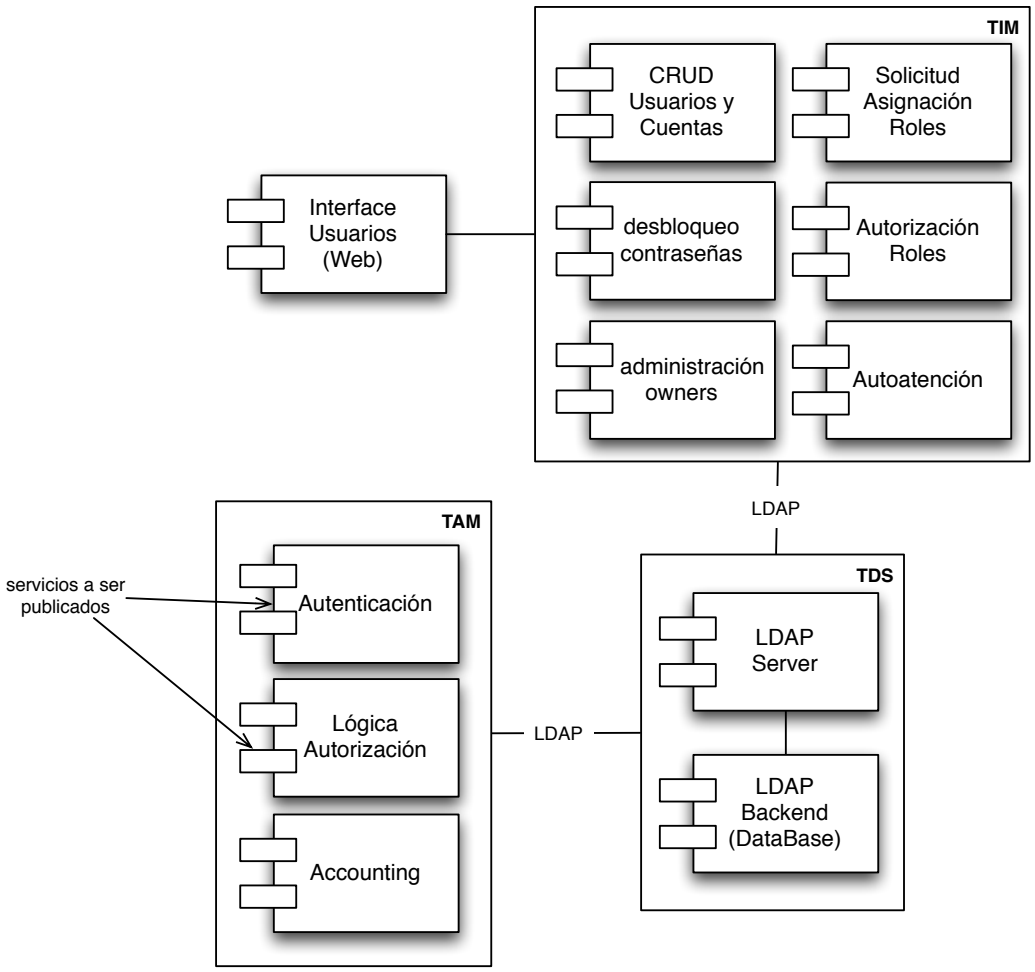


Figura 4.4: Diagrama de componentes

  - Accounting de actividades propias
- ▷ Tivoli Access Manager (TAM)
  - Autenticación
  - Autorización
  - Accounting (Autenticación y Autorización)
- ▷ Tivoli Directory Server (TDS)
  - LDAP Server
  - LDAP Backend (Database)

Los servicios que son publicados para su uso externo son los correspondientes a TAM-Autenticación y TAM-Autorización.

Toda la comunicación desde TIM y TAM hacia el LDAP server TDS es LDAP-based.

4.7.4. Información

Como opción de diseño del DIT se utilizó un estilo plano. Veamos ahora el contenido y razones de esta decisión.

Contenido del DIT

En la figura 4.5 se observa la disposición de los nodos. La información en los nodos es la siguiente:

root naming context
: El nodo raíz del DIT, con nombres de dominio dc=fonasa, dc=cl.

users
: Contiene los datos de usuarios internos, usuarios EAPS y de clientes externos para servicios WS proporcionados por Fonasa.

groups
: Contiene nóminas de grupos de usuarios para los cuales se establecerán permisos de acceso a los recursos (ACLs). Los grupos permiten clasificar a los usuarios en diferentes segmentos (administradores, resolutores, solicitadores), owners y roles. Por supuesto los grupos no son excluyentes, de modo que un usuario puede pertenecer a más de una categoría y a más de un rol.

apps
: Corresponde a las aplicaciones y grupos funcionales dentro de las aplicaciones (recursos).

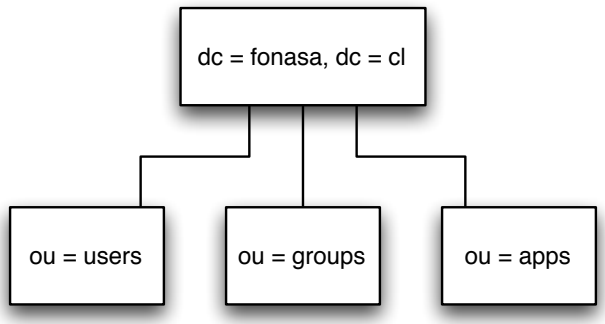


Figura 4.5: DIT de Fonasa

Análisis de la estructura del DIT

Un DIT puede ser plano (poca jerarquía) o profundo (con muchos niveles jerárquicos). Según ha sido mencionado, en general es una buena práctica mantener el DIT lo más plano

posible (lo que simplifica las búsquedas), pero teniendo en mente satisfacer los requerimientos de la aplicación. En este caso, los elementos considerados para la elección de un DIT plano son los siguientes:

- ▷ Representación de la información
  Con los elementos descritos es posible representar todos los elementos requeridos en el sistema (usuarios, recursos, accesos)

- ▷ Organización.
  No existen necesidades de divisiones geográficas ni organizacionales. En este sentido, Fonasa es tratado como una unidad, y las sucursales y reparticiones regionales corresponden a atributos del usuario, y no conllevan restricciones ni tratamientos especiales de ninguna especie. Lo propio ocurre con los usuarios EAPS. Una ulterior consideración es que resulta frecuente definir niveles jerárquicos adicionales en las organizaciones tipo holding, pero este tampoco es el caso.

- ▷ Particionamiento
  Toda la información se encuentra consolidada en el directorio, con administración centralizada, y por tanto no existen necesidades de particionamiento de la data que podrían justificar niveles jerárquicos adicionales.

- ▷ Replicación
  En un DIT profundo es posible definir replicación por ramas o segmentos completos del árbol. En un escenario plano, la replicación del directorio se efectúa como una unidad, que es precisamente el tratamiento más adecuado en este caso.

- ▷ Control de Acceso
  El control de acceso es de tipo funcional, es decir, obedece a los roles que ejercen los funcionarios. En consecuencia para su representación es suficiente con generar agrupaciones simples de usuarios. Por ende, no se requieren niveles o divisiones jerárquicas adicionales para representar este aspecto.

- ▷ Soporte a la aplicación
  Las funciones del sistema Gestión de Usuarios, así como los servicios, se pueden resolver mediante la modificación y consulta de datos en los diferentes componentes del directorio, y no requieren otros directorios ni otros niveles jerárquicos para su resolución.

4.7.5. Desarrollo

Se requiere configurar TDS (LDAP) utilizando los esquemas del directorio.

En el caso de TIM es necesario efectuar la customización de las pantallas autoatención y administrador, y configurar los flujos para la solicitud de asignación de roles.

Para el caso de TAM se requiere configurar las reglas de permisos de acceso a los recursos y publicar los servicios de autenticación y autorización.

4.7.6. Deployment

En la figura 4.6 se visualiza la disposición de los elementos físicos y su conectividad. Se aprecia la duplicidad de todos los servidores y el ambiente en clustering. La conectividad es doble y cruzada para alta disponibilidad.

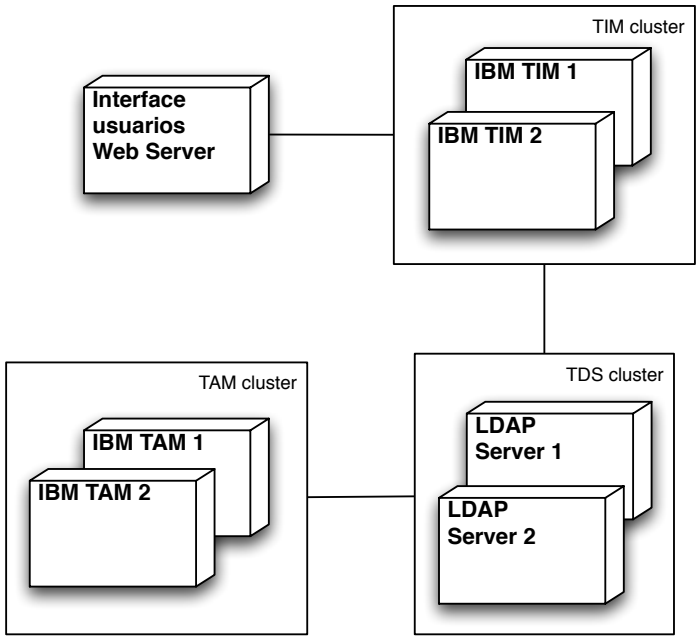


Figura 4.6: Diagrama físico

En la figura 4.7 se muestra la disposición y distribución de las componentes de software en los dispositivos o servidores físicos (hardware).

Los cluster TIM y TAM operan en modalidad activo-activo. Las transacciones son derivadas hacia los servidores TIM y TAM a través de balanceadores de cargas externos. En caso de falla de alguna componente/servidor de un cluster, el balanceador continúa enviando las transacciones al servidor que aún está funcionando. Una vez restaurado la operación del servidor "caido" el balanceador lo identifica y comienza a enviarle transacciones nuevamente. En definitiva, un evento de falla sólo disminuye el rendimiento, pero no afecta la operación.

El cluster TDS opera en modalidad activo-pasivo. Uno de los servidores TDS recibe todas las transacciones y las replica al otro servidor. En caso de falla del componente activo, el nodo pasivo pasa a modo activo, toma el control y continúa procesando las transacciones. Una vez restaurada la falla el nodo recuperado toma el rol de pasivo, se sincroniza con el nodo activo y continúa la operación normal.

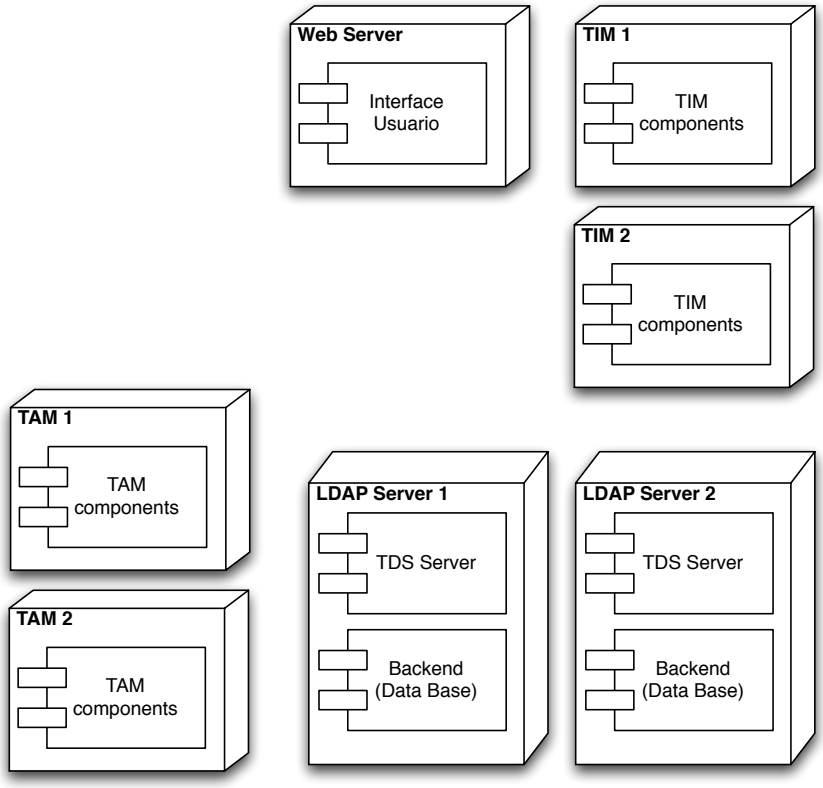


Figura 4.7: Diagrama de deployment

4.7.7. Operación

Las aplicaciones operan en cluster, de modo que la instalación se debe realizar en duplicado.

La operación de los servicios debe supervisar que todos los ambientes operen correctamente. El monitoreo de aplicaciones y servicios se debe efectuar utilizando las herramientas que se definan para la supervisión y control global de la plataforma Fonasa.

En este sentido, Gestión de Usuarios es una componente más dentro de la plataforma, y por tanto se adhiere al régimen general de control de los sistemas Fonasa.

4.8 Diseño y Modularización

4.8.1. Necesidades de programación y/o customización

Según indica Howes y otros en su libro de LDAP [9], el diseño del directorio debe contemplar los siguientes aspectos:

- ▷ Requerimientos hacia el Directorio
- ▷ Data
- ▷ Schema
- ▷ Namespace
- ▷ Topología
- ▷ Replicación
- ▷ Seguridad

En el caso de la suite Tivoli el diseño se materializa a través de las herramientas de configuración.

Para estos efectos se configuraron:

- ▷ Tipos de usuario Tivoli (usuarios desde el punto de vista funcional del sistema Gestión de Usuarios)
- ▷ Customización de schemas del LDAP, de acuerdo al diseño del modelo de usuarios de Fonasa
- ▷ Utilización del Namespace definido, en el momento de efectuar la carga inicial de usuarios
- ▷ Customización de permisos de Tivoli
- ▷ Customización de permisos del primer grupo de aplicaciones del proyecto global de Fonasa

4.8.2. LDAP TDS

Se definieron los esquemas, la estructura de usuarios y el árbol de Namespace (DIT) a utilizar.

Los esquemas utilizados corresponden al que provee TDS (que es estándar LDAPv3 más algunas extensiones de IBM).

Respecto al namespace las entries se agregan utilizando el diseño definido para el DIT.

[9] Howes *et al.* [8, capítulo 5]

El DIT es plano y centralizado. Además se utilizan los mecanismos proporcionados por TDS para la replicación en cluster.

Respecto de la seguridad, tanto TDS como el resto de las componentes se encuentran dentro del área protegida de la plataforma Fonasa.

4.8.3. TIM

Customización para TIM (usarios/cuentas)

TIM proporciona dos interfaces web que operan de acuerdo al nivel de permisos de los usuarios.

- ▷ Administración de TIM
- ▷ Autoatención

Interfaces Usuario

El usuario hará uso de las dos aplicaciones web de TIM.

La configuración fue simplemente:

- ▷ Personalización de pantallas con logo Fonasa (usando plantillas CSS)
- ▷ Funcionalidades permitidas son actualizadas por TIM automáticamente, de acuerdo al tipo de usuario conectado. Es decir, la configuración de permisos genera automáticamente la customización de los menús de TIM.

Grupos y Roles

Para hacer uso del módulo TIM se configuraron roles/perfiles de usuario. Estos roles corresponden a grupos dentro de TIM:

- ▷ Usuarios administradores de aplicaciones de seguridad
- ▷ Usuarios de negocio
  - – Usuario normal (perfil base mínimo para usuario Fonasa)
  - – Usuario Solicitador
  - – Usuario Resolutor

Todos los usuarios de negocio, independientemente de su tipo, presentan *roles* de acuerdo a las funciones que les toca desempeñar. Por ejemplo:

- ▷ Usuarios administradores

- ▷ Usuarios de negocio
  - Cajero
  - Supervisor
  - Ejecutivo
  - etc.

La solución fue definir roles no excluyentes, es decir, un usuario es creado con un rol o perfil base (como funcionario Fonasa) y puede tener uno o más roles adicionales, según sea necesario.

TIM permite entonces agregar o eliminar roles a los usuarios. Por su parte la relación entre roles y recursos es establecida en el módulo TAM.

Dueños de recurso

Los roles/recursos pueden en algunos casos tener *dueños de recurso*, que son los que autorizan que un usuario pueda o no acceder a dicho rol/recurso (y a los permisos que conlleva). Esto se modela como grupos de usuarios asociados a un rol/recurso, en este caso en calidad de propietarios. Además, a través de la herramienta de workflow de TIM se configuró la solicitud de autorización planteada en los requerimientos del sistema.

Política de password

Se configuraron especificaciones respecto de la administración de password. Detalle de customización para LDAP (largo password, caracteres permitidos y obligatorios, período de validez de la password, cantidad de intentos de conexión, etc).

4.8.4. Servicios TAM

Customización para TAM (control de accesos)

Como habíamos señalado durante el levantamiento, en esta etapa se desarrolló el concepto de *Matriz de Permisos*. Corresponde al patrón de autorización *Role Rights Definition*, que se encuentra asociado al RBAC y fundamenta la definición de roles en base a los casos de uso.

Como tipos de usuario y roles se utilizaron los definidos en TIM (usuarios administradores, los diferentes usuarios de negocio y los roles).

Recursos

Como *recursos* se identificaron y agruparon las funcionalidades asociadas a los sistemas de la primera etapa. Cada funcionalidad tiene definida una URL y se modela como un objeto/recurso que debe ser protegido. Estos objetos protegidos son ingresados a la configuración de TAM.

Matriz de Permisos

La Matriz de Permisos consiste en asociar entonces los roles con los recursos (funciones), indicando si tienen o no acceso (de acuerdo a las tareas que les corresponde realizar). El formato se puede ver en la tabla 4.7.

Funciones		**Roles**			
Sistema	Función	Cajero	Supervisor	Ejecutivo	Otros
Sistema 1	función 1	x	x	x	
	función 2		x	x	x
	función 3	x	x		x
	función 4		x		x
Sistema 2	función 5	x	x		
	función 6		x		x

Tabla 4.7: Tabla Matriz de Permisos

Sistema 1 y Sistema 2 representan sistemas del Proyecto Global, y las funciones corresponden a las tareas específicas que existen dentro de dichas aplicaciones, por ejemplo, consultas, actualizaciones de información, emisión de documentos y certificados, otras operaciones. En esta tabla se muestran los sistemas y las funciones asociadas, y los roles que existen dentro de la organización. Las marcas en la tabla indican si está permitido el acceso de un rol a una determinada función de un sistema dado. En el ejemplo, Sistema 1 tiene las funciones 1, 2, 3 y 4, y Sistema 2 tiene las funciones 5 y 6. El rol Cajero tiene acceso a las funciones 1 y 3 del Sistema 1, y la función 5 del Sistema 2, y el rol Supervisor tiene acceso a todas las funciones de ambos sistemas.

A medida que se vayan incorporando nuevos sistemas a la plataforma se debe también ir completando la matriz de permisos, agregando nuevos sistemas/funciones y los accesos correspondientes.

Una matriz de permisos se puede modelar en TAM definiendo para cada objeto protegido (funciones) un ACL asociado. Y en cada ACL se incluyen los grupos (roles) de usuarios que deben tener acceso al objeto respectivo.

En definitiva esta información se ingresó como matriz de permisos a TAM, suscribiendo de este modo el esquema global de seguridad y autorización definido.

Policies

La política de autorización puede tener muchas variantes, pero en su forma más básica opera de la siguiente forma:

- ▷ recibe un requerimiento de autorización de acceso a un determinado recurso, por un cierto usuario
- ▷ el recurso viene indicado por su dirección URL
- ▷ se busca en el LDAP un recurso que "mapee" con la URL recibida, y se obtiene la entry de recurso asociada
- ▷ se busca un ACL asociado al recurso, de tipo "autorización de acceso", y se obtienen los roles asociados
- ▷ luego se buscan los usuarios dentro de los roles
- ▷ decisión de autorización: Si el usuario pertenece a alguno de los roles, se autoriza la acción. Si no es así, se rechaza.

Publicación de servicios TAM

Como hemos visto, TAM provee los servicios de AAA (Autenticación, Autorización y Accounting) como una componente activa de la plataforma global. En especial, se dejaron disponibles los servicios TAM de Autenticación y Autorización, para su publicación y uso por las aplicaciones y sistemas de la plataforma.

De esta forma todas las aplicaciones podrán hacer uso de los servicios de seguridad, y de acuerdo a los usuarios y configuraciones establecidos en el repositorio LDAP, TIM y TAM.

4.8.5. Requisitos No Funcionales

Esta arquitectura está especialmente enfocada al cumplimiento de los requisitos no funcionales.

En lo esencial, tenemos:

- ▷ Componentes dedicadas para el cumplimiento de funcionalidades específicas
- ▷ Duplicación de componentes en ambiente de clustering
- ▷ Uso de balanceadores de carga
- ▷ Utilización de un service bus para el manejo de mensajería (Message broker)

Rendimiento y Escalabilidad

En esta arquitectura cada módulo aborda una parte de la funcionalidad. Se encuentran interrelacionados pero son independientes. Los módulos están duplicados y el rendimiento se

amplifica al utilizar ambiente de clustering con balanceadores de carga para cada módulo.

Ante necesidades de crecimiento cada módulo se puede hacer crecer por separado. Esto además permite focalizar los upgrade y no se generan efectos laterales indeseados en el resto de la plataforma.

Tolerancia a Fallas y Alta Disponibilidad

Todos los módulos se encuentran duplicados, en un ambiente de clustering, apuntando así a los requisitos de Tolerancia a Fallas y Alta Disponibilidad.

Además, resulta interesante señalar que una eventual falla en TIM sólo inhabilita las operaciones sobre la base de usuarios, pero la plataforma global puede seguir operando.

Desde ese punto de vista, en orden de criticidad, las plataformas más sensibles son TDS (LDAP) y TAM, y luego TIM.

Mantenibilidad

Los módulos son independientes e interrelacionados. Si es necesario efectuar una mantención o algún cambio, éste se focaliza en el módulo respectivo.

Los nuevos sistemas se irán incorporando a la plataforma en forma natural. Respecto de sus necesidades de seguridad, los protocolos de autenticación sólo deben usarse, pues no hay necesidad de programarlos. En cuanto a los requerimientos de autorización, los nuevos sistemas deben entregar su propia matriz de permisos, para efectuar la mantención respectiva en el módulo TAM.

Site 1 y 2

Respecto de los site 1 y 2, y aunque LDAP TDS dispone de mecanismos de duplicación inter-site, se adoptó como decisión de diseño no utilizar estas funcionalidades. La forma de resolver este tema fue simplemente adherirse al mecanismo global de duplicación de sites que ha sido definido para todas las aplicaciones de la plataforma.

La razón es simplemente separar responsabilidades y disponer de un único mecanismo global de sincronización, switching y restauración entre sites. En la práctica se está haciendo uso de los principios GRASP [10], aplicados en este caso a la arquitectura de la solución.

Sin pretender ser exhaustivos, los mecanismos de site 1 y 2 están conformados por una compleja infraestructura de equipamiento, red y software, así como procedimientos y servicios que controlan y supervisan en todo momento el estado de las plataformas. Además, cada site por si mismo presenta alta disponibilidad.

[10] General Responsibility Assignment Software Patterns

Los sites operan en modalidad activo-pasivo, y toda la información se encuentran duplicada y actualizada permanentemente. El site principal juega el rol de nodo activo, mientras que su homólogo es el pasivo. Cuando se producen eventos o circunstancias que requieran la conmutación del switch, en lo esencial, lo que ocurre es que se intercambian los roles, es decir, el nodo activo pasa a pasivo (o bien, dependiendo de la contingencia, puede haber dejado de funcionar) y el nodo pasivo se convierte en activo y toma las funciones de su par. El objetivo es que no exista impacto directo en las aplicaciones, y que la conmutación de sites sea absolutamente transparente para los usuarios. El site secundario es completamente autónomo y puede operar en estas condiciones todo el tiempo que sea necesario.

Una vez determinada la condición de falla o excepción producida en el site primario, se efectúan las tareas necesarias para su recuperación. Luego, para restablecer la operación del site primario, se ejecutan una serie de protocolos definidos para estos efectos. Basicamente, se restablece la comunicación, se sincronizan datos y aplicaciones, y el site primario vuelve a tomar el rol de nodo activo y el secundario el de nodo pasivo, con lo cual se vuelve al estado original.

4.9 Análisis de la Solución

Separación de Intereses

En un sistema *security oriented* se tiene especial cuidado en garantizar el cumplimiento de los procesos de seguridad en forma eficiente y estructurada. Los procesos de autenticación y autorización se encuentran separados y claramente delimitados. Los procesos de auditoría se efectúan en forma automática y siguiendo las directrices y necesidades de la organización.

En el diseño de esta solución se observa este principio y se cumple con:

- ▷ módulos funcionales dedicados
- ▷ configuración y parametrización independiente
- ▷ escalabilidad

Además se observa:

- ▷ la seguridad es una plataforma dedicada, e independiente de las aplicaciones
- ▷ la seguridad está modularizada en diferentes componentes, configurables y mantenibles por separado: Autenticación, autorización y auditoría
- ▷ cada componente es escalable independientemente. Si observamos contención en alguna componente podemos tomar medidas focalizadas y dirigidas, sin afectar las aplicaciones ni al resto de las componentes de la plataforma de seguridad

Veamos las ventajas de la separación de intereses:

En muchos desarrollos antiguos y nuevos se separa la autenticación, pero se sigue man-

teniendo la autorización como parte de la lógica de las aplicaciones. Esto redunda en multiplicidad de esquemas, dificultades de mantención, dificultades de conciliación, distracción de recursos, etc.

En este desarrollo se va más allá: Tanto la autenticación como la autorización se constituyen en servicios separados, independientes y lo más importante: Son utilizables por las aplicaciones.

El disponer de un módulo específico y dedicado para los procesos de autorización nos permite:

- ciclos de desarrollo más cortos. Nuevos desarrollos utilizan la infraestructura de autorización. No requieren construirla.
- reducción en los costos de desarrollo
- soporte a las nuevas aplicaciones
- infraestructura de seguridad única y controlada

Los nuevos sistemas de las siguientes fases no necesitan construir sistemas de autenticación/autorización. Por supuesto es perentorio que se utilice la estructura de seguridad proporcionada y que se disponga de la matriz de roles/permisos para agregarla a la configuración. El acceso a los sistemas está controlado a través del reverse proxy o el bus de servicios. De esta forma cada vez que un usuario requiera acceso a un sistema, primero se chequea si está o no conectado. Si no lo está, se solicita el servicio de autenticación. Una vez identificado, se solicita el servicio de autorización, y si pasa las verificaciones se entrega el control al sistema solicitado. Cada nueva transacción ejecutada sobre el sistema vuelve a ser verificada (en conexión/autorización) y el ciclo continúa.

Adicionalmente, los sistemas pueden invocar directamente los servicios de autenticación/ autorización y efectuar sus propias validaciones de seguridad, por supuesto respetando los formatos de intercambio de información definidos. Por ejemplo, si el sistema desea mostrar sólo la funcionalidad habilitada (en un menú de opciones), puede utilizar los servicios de autorización, y de esta forma "saber" a priori cuáles son las funcionalidades autorizadas para el usuario.

Todo esto se traduce en que los nuevos desarrollos estén soportados por la plataforma, que la funcionalidad de seguridad simplemente se utilice y -en definitiva- que no sea necesario invertir esfuerzos adicionales en esta materia.

Cumplimiento de requisitos

- Los aspectos funcionales han sido cubiertos mediante customización de las componentes de IBM y vía construcción para el caso de la migración de datos
- En cuanto a los aspectos No funcionales, estos se cumplen por el uso de patrones, la arquitectura en clustering y las propiedades de LDAP

4.10 Migración de Datos y Carga Inicial

Habitualmente en el desarrollo de sistemas la migración de datos es relegada a un segundo término y no pocas veces esto genera retrasos y problemas de última hora.

En el caso del proyecto Gestión de Usuarios el tema fue abordado desde un comienzo y se trabajó paralelamente con el desarrollo de la aplicación. [11]

Para llevar a cabo exitosamente esta parte del proyecto se consideraron las siguientes macro tareas:

- ▷ Definición de estrategia de Migración y Carga
- ▷ Elaboración de un Plan de Trabajo
- ▷ Validación y retroalimentación
- ▷ Carga Inicial

4.10.1. Problemas

- ▷ Multitud de orígenes de datos (planillas, archivos planos, documentos de texto, etc)
- ▷ Codificación no única (códigos distintos, uso de texto en vez de códigos)
- ▷ Codificación no actualizada en todas las áreas usuarias
- ▷ Codificación no actualizada en todos los sistemas
- ▷ Desconocimiento de la nueva codificación
- ▷ Información desactualizada o histórica mezclada con la información válida y vigente (por ejemplo, códigos de usuario para personas que ya no pertenecían a la Institución)
- ▷ Planillas con distinto orden en las columnas.
- ▷ Rótulos en planillas de datos con identificadores distintos (por ejemplo, “sucursal” y “código suc.”)

Esto conlleva un arduo trabajo de clasificación y ordenamiento de la información. Software para efectuar validaciones automáticas, definición de procedimientos de recopilación y retroalimentación, etc.

4.10.2. Estrategia Aplicativo Migración

Para disponer del software necesario existían dos alternativas:

- ▷ Utilización de algún paquete de software tipo ETL (Extract-Transform-Load)

[11] Dependiendo del volumen de datos y su complejidad, en algunos casos es recomendable desarrollar un proyecto paralelo de migración de datos, con recursos dedicados en forma exclusiva

▷ Desarrollo de aplicativo específico

En este caso se optó por el desarrollo de un aplicativo. La motivación fue la siguiente:

▷ costos más altos en uso de ETL
▷ mayor flexibilidad en un desarrollo ad-hoc, ante la variedad de entradas de información
▷ utilización del aplicativo por una única vez

La utilización de un ETL requiere de desarrolladores y/o consultores que dominen la tecnología específica a utilizar, así como la adquisición de las licencias respectivas. Por supuesto existen opciones open source, pero sigue presente el problema del conocimiento de la herramienta. Además, la alternativa de invertir horas en aprendizaje de alguna herramienta open source, por razones de tiempo se descarta de plano.

Por su parte, las horas de programación y/o consultoría en algún lenguaje de programación (C++, Java, Python, etc) son de menor costo, e incluso existe disponibilidad interna de recursos para efectuar estas tareas.

En cuanto a los aspectos de flexibilidad en el desarrollo, las herramientas ETL habitualmente proveen interfaces gráficas que resultan muy atractivas e intuitivas, pero no siempre resultan fáciles de modificar y/o reutilizar. En el caso de los lenguajes de programación, especialmente aquellos orientados a objeto, los cambios y la reutilización de código son inherentes a la naturaleza del lenguaje (por supuesto, en manos de un buen programador y haciendo un correcto uso de paradigmas y patrones). En el caso de Gestión de Usuarios, se sabía de antemano que existía una gran variedad de orígenes de información, pero no se conocía a priori la calidad de los datos que se iban a recibir. Los cambios y ajustes (mayores y menores) eran inevitables. En este sentido, la expectativa es que modificar graficamente varios módulos ETL, es más lento que agregar "if" en algunas clases Java (u otro) específicas.

El último aspecto considerado es si el aplicativo tiene o no un uso recurrente, En los casos de uso recurrente, ETL en general representa una ventaja. La razón es que la comunicación y/o cambios se pueden efectuar interactuando directamente con los usuarios. Incluso resulta razonable transferir el control de estos cambios directamente a las áreas usuarias. Pero este no es el caso en la migración de datos de Gestión de Usuarios. Es un aplicativo que se va a utilizar por una única vez, al inicio de la operación del sistema.

4.10.3. Plan Migración/Carga

En la tabla 4.8 se observa la descripción del plan con las actividades y responsables. Este plan contempla un trabajo conjunto para lograr establecer acuerdos en reglas de migración, y procedimientos de pruebas y retroalimentación, de manera de alcanzar los niveles de satisfacción requeridos.

Actividad	Responsable
Definición de reglas de Migración y Carga	Equipo Desarrollo y GTI
Definición de reglas de validación	Equipo Desarrollo y GTI
Difusión de acuerdos y codificación	GTI
Desarrollo de software de validación	Equipo Desarrollo
Recopilación de Información	GTI y áreas Usuarias Fonasa
Validación y retroalimentación	Equipo Desarrollo
Pruebas de Cargas de datos en ambiente de Test	Equipo Desarrollo
Obtención de los niveles de calidad acordados	Equipo Desarrollo, GTI y áreas Usuarias Fonasa
Carga Inicial definitiva	Equipo Desarrollo y GTI

Tabla 4.8: Plan migracion de datos

4.10.4. Diseño Aplicación Migración/Carga de Datos

Recopilación de Información

- ▷ Desarrollo de planillas Excel con formularios de ingreso de datos
- ▷ inclusión de hojas con códigos de tablas para selección
- ▷ inclusión de macros para selección de códigos, de forma de minimizar los errores de ingreso de datos

Aplicativo de Validación

- ▷ Software desarrollado en Java
- ▷ Entrada.
  Formato uniforme en archivos planos, extensión “csv”, utilizando pipes como separadores (pues la “,” y el “;” eran caracteres válidos en algunas glosas).
  El contenido de los archivos corresponde a los datos de usuarios y la lista de tablas para efectuar validación de códigos
- ▷ Persistencia.
  Considerando los volúmenes de datos de tablas no fue necesario utilizar elementos de persistencia (se podría haber usado minibases tipo SQLite u otro, pero no fue requerido)

- ▷ Procesamiento Batch.
  Carga de tablas en memoria. Validación de nóminas de usuarios. Chequeos de consistencia de la data (por ejemplo, dígito verificador), chequeos contra códigos de tablas. Generación de errores y acumulación de estadísticas.
- ▷ Salida.
  Archivos planos con errores (indicando para cada registro erróneo la lista de errores asociados) y estadísticas de validación

En diagrama 4.8 se observa la composición de los package del aplicativo batch de validación.

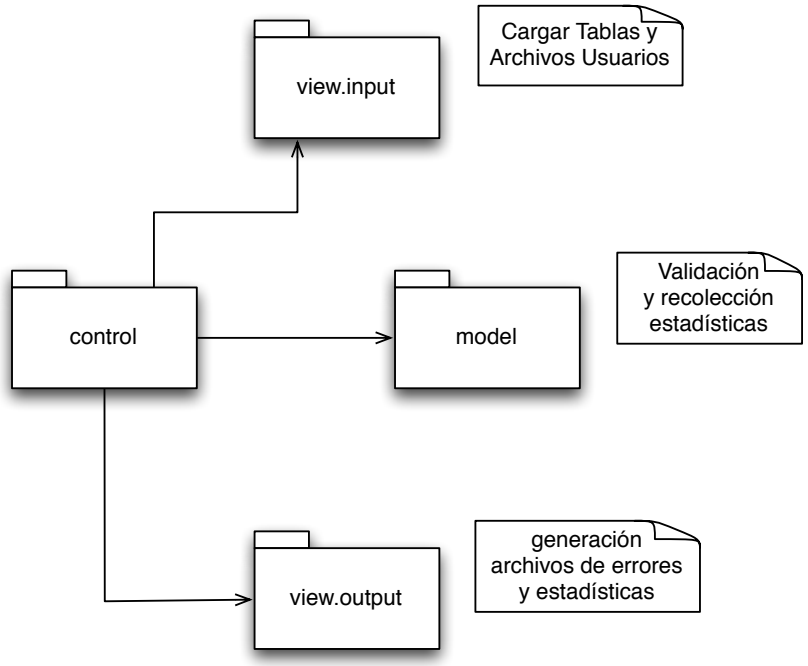


Figura 4.8: Diagrama de paquetes de aplicativo de validación

Las nóminas de archivos se validaron en grupos, de acuerdo a distribución geográfica y otros criterios. Cuando se logró el porcentaje de error deseado para un grupo determinado, se procedió a la etapa de carga.

En el Anexo 3 se muestra un mayor detalle de este aplicativo.

Carga

- ▷ Obtención del umbral de error deseado
- ▷ Definición de fechas de carga
- ▷ Proceso de carga, utilizando las herramientas de carga masiva de LDAP y TIM (estas herramientas toman los archivos excel con extensión “csv” y generan internamente un archivo LDIF, que es el que se carga en definitiva)

4.11 Implementación Gestión Usuarios

Para el desarrollo e implantación del sistema Gestión de Usuarios se dispuso de tres ambientes separados, con procedimientos estrictos de traspaso de datos y versiones de software.

- ▷ Ambiente de Desarrollo
- ▷ Ambiente de Test
- ▷ Ambiente de Producción

En cuanto a la estrategia de inicio de operaciones, se barajaron dos opciones:

- ▷ *Big-Bang*
  Inicio de operación inmediato, en todo el país.
- ▷ Piloto
  Inicio de operaciones escalonado, partiendo con un piloto.

La opción big-bang tiene como ventaja que es más simple desde el punto de vista de la administración de la plataforma. Un nuevo sistema, único, todas las transacciones ingresando en dicho sistema, etc.

En la opción Piloto es necesario conciliar la coexistencia de sistemas antiguos funcionando en algunas oficinas y el sistema nuevo, operando en las oficinas y reparticiones del Piloto. Además, es necesario inyectar las transacciones del día en forma cruzada para mantener la consistencia.

No obstante, en la eventualidad de errores, problemas, o dificultades de uso de la nueva plataforma, la opción Piloto permite mantener la situación bajo control y proporciona un nivel de reacción bastante más rápido.

Por esta razón, la opción más conservadora y de menor riesgo, y que fue adoptada en definitiva, es iniciar la puesta en marcha con un Piloto.

4.11.1. Instalación y Paso a Producción

Efectuar un Piloto fue una estrategia global de inicio de operación. Esto incluye, por supuesto, la Gestión de Usuarios y los sistemas de la Primera Fase del Proyecto Global.

Como etapas del Piloto se definió:

- ▶ Piloto (elegir oficinas representativas)
- ▶ Incorporación paulatina del resto de oficinas y reparticiones (Santiago y Regiones)

4.11.2. Piloto

Estrategia para llevar a cabo el Piloto. Etapas.

- ▷ Recopilación y Validación Usuarios Piloto
- ▷ Carga Inicial de Usuarios
- ▷ Pruebas previas al Inicio del Piloto (funcionales y de ambiente)
- ▷ Inicio del Piloto (pruebas durante la ejecución del Piloto)
- ▷ Desarrollo del Piloto con operaciones normales
- ▷ Carga paulatina del resto de los Usuarios (no sobrecargar Help-Desk)
- ▷ Evaluación del Piloto para inicio operación resto de Oficinas

Requisitos para inicio del Piloto

- ▷ Carga Usuarios completa
- ▷ Pruebas previas efectuadas
- ▷ Conectividad verificada
- ▷ Capacitaciones efectuadas (usuarios, monitores, mesa de ayuda, etc.)
- ▷ Procedimiento de "vuelta atrás" definido (global y de cada sistema)

Evaluación del Piloto

- ▷ Evaluación de incidencias y su nivel de riesgo para la operación (volumen de incidencias, riesgos asociados y correcciones realizadas)
- ▷ Control de operaciones efectuadas vs registradas
- ▷ Revisión de estadísticas
- ▷ Balance general y decisión

4.11.3. Incorporación paulatina resto Oficinas

Estrategia para llevar a cabo la Incorporación Resto Oficinas. Etapas.

- ▷ Carga inicial de Usuarios por grupos de Oficina (ejecutado durante el Piloto)
- ▷ Inicio de operación de cada grupo de Oficinas

Requisitos para inicio operaciones resto Oficinas

- ▷ Piloto "exitoso"

- ▷ Carga Usuarios completa
- ▷ Pruebas previas efectuadas
- ▷ Conectividad verificada
- ▷ Capacitaciones efectuadas (usuarios en oficinas y regiones, mesa de ayuda, etc.)
- ▷ Procedimiento de "vuelta atrás" definido (global y de cada sistema)

Estos requisitos persiguen el objetivo que cada nueva oficina incorporada, tenga asegurada su correcta operación y, en caso de eventuales fallas, tenga un procedimiento de "bajada" o "vuelta atrás" claramente definido.

Para el caso de las capacitaciones, para el caso de Gestión de Usuarios, se estableció con Fonasa efectuar las siguientes capacitaciones:

- ▷ Capacitación directa a usuarios monitores (que a su vez efectuaron capacitaciones en sus áreas/regiones específicas)
- ▷ Disponiblidad de sitio de pruebas/capacitación para pruebas y autocapacitación
- ▷ Entrega de manuales de usuario (Manual de Uso General, Manual de Uso GTI, Manual Asignación de Roles y Permisos)
- ▷ Capacitación específica a personal de Help-Desk

En cuanto al procedimiento de "vuelta atrás", es un documento que establece las condiciones y mecanismos para "bajar" la sucursal de operación y retornar al estado anterior a la subida a operación en el nuevo sistema. Cabe señalar que el inicio de operación implica la utilización de Gestión de Usuarios y de todos los sistemas de la primera etapa. En consecuencia, la "vuelta atrás" considera también los procedimientos para la bajada de todos los sistemas involucrados.

Este documento, que es de carácter global, indica cuales son las condiciones y niveles de tolerancia, los requisitos para tomar una decisión de abortar la puesta en producción y quién o quienes están autorizados a tomar dicha decisión. Además, para cada sistema involucrado, los tiempos y recursos asociados y el detalle del procedimiento de "vuelta atrás". La entrega de los antecedentes para cada sistema fue responsabilidad de cada Jefe de Proyecto, y lo propio ocurrió en el caso de Gestión de Usuarios.

4.11.4. Procesos y Procedimientos básicos

- ▷ Monitoreo
- ▷ Estadísticas de Operación
- ▷ Registro de incidencias (evaluación y correcciones)
- ▷ Respaldos

4.11.5. Apoyo a la Gestión del Cambio

Para materializar exitosamente su proyecto Global de Modernización de Plataformas, Fonasa elaboró un Plan de Gestión del Cambio, que buscaba cumplir con una variedad de objetivos, orientados principalmente al conocimiento de los beneficios y la internalización de los usos y procedimientos asociados. Esto se tradujo en las siguientes directrices:

- ▷ Motivación
- ▷ Difusión
- ▷ Capacitación
- ▷ Coordinación de las Áreas
- ▷ Participación activa de los Usuarios

Nuestro aporte en esta materia fue entregar oportunamente los diversos elementos que fueron requeridos durante la concreción de las tareas asociadas con dicho Plan, especialmente en las actividades previas al inicio de la operación.

Tareas desarrolladas para la nueva Plataforma

- ▷ Uniformidad y estándares de Codificación (Tablas)
- ▷ Documentación y entrega para difusión de su uso

Específicamente para Gestión de Usuarios

- ▷ Documentación de estándares y normas utilizados (formato de códigos de usuario, derechos y responsabilidades de los usuarios en relación al nuevo sistema, etc)
- ▷ Redacción de manuales de Usuario / Operación / Help-Desk
- ▷ Efectuar labores de Capacitación (Usuarios / Help-Desk)
- ▷ Difusión (generación de correos masivos informando creación cuentas)

Capítulo 5

Validación

La verificación del éxito del sistema tiene varios aspectos. Para los usuarios finales, en lo esencial, el sistema debe satisfacer funcionalmente sus requerimientos. Para arquitectos, desarrolladores y explotadores, el sistema debe satisfacer sus requisitos de calidad.

En cualquier caso, la adecuada planificación, y un diseño detallado de los escenarios y casos de prueba resultó fundamental para obtener la certificación y aprobación del sistema, y en definitiva lograr su paso a Producción.

5.1 Actividades

Los pasos seguidos para definir el Plan de Pruebas:

- ▷ Definir ámbito de las pruebas (sistemas, plataformas, HW, SW)
- ▷ Estrategia de Pruebas
  - Acordar cobertura y alcance de las pruebas
  - Acordar criterios de aceptación
- ▷ Plan de Pruebas
  - Definir y documentar casos de prueba (globales y detalle)
- ▷ Efectuar pruebas, tomar evidencias, documentar resultados, acordar cierre/repetición de la prueba
- ▷ Cierre de pruebas y conclusiones

5.2 Definición del Proceso de Prueba

Ámbito de las Pruebas

Las pruebas del sistema Gestión de Usuarios tienen como ámbito:

- ▷ Funcionales
  Corresponde al chequeo de la correcta funcionalidad de la aplicación, tanto en forma aislada como integrada con el resto de las aplicaciones.
- ▷ No Funcionales (PNF)
  Corresponde a la verificación de los atributos de calidad de la aplicación.
- ▷ Piloto
  Es un subconjunto de las pruebas funcionales, a ser verificado antes y durante el Piloto.

Criterios de Validez

El objetivo es responder la pregunta ¿En qué porcentaje/umbral la prueba estará superada?

Los SLAs ya están establecidos en las bases de la licitación efectuada por Fonasa. Sin embargo, durante la etapa de prueba corresponde definir el detalle de cómo se llevará a cabo la medición para cumplimiento de SLA y certificación de pruebas.

Los ítemes a definir para cada SLA y en cada prueba son:

- ▷ Qué medir
  Cuál es el concepto o conceptos que mejor reflejan la medición del SLA.
- ▷ Cómo medir
  Cómo se debe llevar a cabo la medición. En qué condiciones. Cómo se tomarán las evidencias. Tipos de evidencia que serán considerados válidos (printscreen, logs)
- ▷ Cuándo medir
  Cuándo se debe medir. En qué intervalos de tiempo. Instantaneamente, con un printscreen de las pantallas de monitoreo, o en diferido, con una estadística de logs, los propios logs, etc.

Etapas del Proceso de Pruebas

- ▷ Definición del Plan de Pruebas (grupos de prueba, estrategia, diseño detalle).
- ▷ Calendarización de las pruebas.
- ▷ Ejecución de Pruebas
- ▷ Conclusiones y Acuerdos

Procedimiento de Ejecución de Pruebas

Los pasos a ejecutar durante cada Prueba son:

- ▷ Verificar cumplimiento de prerrequisitos
- ▷ Efectuar prueba
- ▷ Tomar evidencias
- ▷ Documentar resultados
- ▷ Acuerdos con Usuario (Repetir, cerrar, etc)
- ▷ Cierre de Pruebas (Acta/Correo/Documento)

5.3 Plan de Pruebas

El Plan de Pruebas se divide en:

- ▷ Plan de Pruebas Funcionales
- ▷ Plan de Pruebas No Funcionales
- ▷ Plan de Pruebas Piloto

Pruebas Funcionales

Las pruebas funcionales consideran los siguientes aspectos:

- ▷ Prueba de las aplicaciones Cliente (Autoatención y Administración)
- ▷ Pruebas de los servicios de Autenticación y Autorización
- ▷ Pruebas del Módulo de Carga

Pruebas No Funcionales

Las pruebas No Funcionales se agrupan en las siguientes categorías:

- ▷ Rendimiento (Pruebas de esfuerzo)
- ▷ Tolerancia a Fallas
- ▷ Seguridad
- ▷ Instalación y Convivencia
- ▷ Respaldo y Recuperación
- ▷ Site 1 y 2

Pruebas Piloto

Las pruebas durante el Piloto correspondieron a un subconjunto de las pruebas funcionales, y su objetivo fue verificar cumplimiento en un ámbito controlado, con una cobertura de usuarios/oficinas mínimo pero completo.

Las pruebas en ambiente Piloto permiten evaluar el comportamiento del sistema en un ambiente absolutamente real y eventualmente detectar errores o desviaciones que por alguna causa no surgieron en las pruebas funcionales anteriores. En este ambiente controlado la capacidad de reacción es mayor, los eventuales impactos en usuarios finales son menores y existe un tiempo mayor para reparar y corregir errores. Por último, estas pruebas permiten pronosticar proyectivamente el comportamiento del sistema, para cuando esté sometido a una carga en operación regular, y adoptar anticipadamente las medidas que puedan resultar necesarias.

Los errores durante un Piloto pueden ser múltiples, pero habitualmente son de tipo ambiental (no de tipo funcional) y normalmente corresponden a un incorrecto paso a Operación, o a diferencias entre los ambientes de Test y Producción que no fueron detectados oportunamente.

5.3.1. Estrategia Pruebas Funcionales

En lo que sigue se delinea la estrategia utilizada para las pruebas funcionales:

Prueba de las aplicaciones Cliente (Autoatención y Administración)

Este grupo de pruebas se dividió en dos partes.

- Unitarias
  Se efectuaron pruebas de cada funcionalidad en forma aislada del resto de la plataforma. Una vez superadas se pasó a la siguiente etapa.
- Integradas
  Efectuar pruebas integradas con el resto de los sistemas de la primera Fase. Las pruebas consisten en verificar los diferentes escenarios: Usuario durante su primera conexión, en una conexión normal, cambios de permisos en los aplicativos que deben reflejarse en aumento o disminución de funcionalidades para el usuario, etc.

Pruebas de los servicios de Autenticación y Autorización

La estrategia de prueba para estos servicios, que son eminentemente integrados, es dividirlos en dos grupos de prueba:

- de aplicaciones internas
  Se debe chequear el correcto acceso de los usuarios, de acuerdo a los permisos establecidos para su respectivo rol.
- de aplicaciones externas (servicios web)
  Se chequea el correcto acceso desde plataformas externas a los web-services proporcionados por los sistemas de la primera fase, de acuerdo a los permisos de acceso establecidos.

Pruebas del Módulo de Carga

Dado que este es una aplicación de tipo batch, que opera en forma stand-alone, la estrategia utilizada fue:

- efectuar pruebas unitarias con un volumen de datos bajo, y luego verificar que los usuarios creados tuviesen acceso, y de acuerdo a los permisos establecidos en el archivo de carga.
- efectuar pruebas unitarias con un volumen de datos bajo, que incluyera diferentes tipos de error, para verificar el comportamiento de la carga en condiciones incorrectas o con diferentes tipos de inconsistencias
- efectuar pruebas con diferente variedad de tipos de usuario
- efectuar pruebas con sucursales completas

5.3.2. Estrategia Pruebas No Funcionales (PNF)

A continuación se describe la estrategia definida para la realización de las pruebas No funcionales:

Rendimiento (Pruebas de esfuerzo)

El foco de esta prueba fue determinar el desempeño de los servicios de autenticación y autorización, que son precisamente aquellos que serán más exigidos durante el funcionamiento de las nuevas aplicaciones.

En otras palabras, el rendimiento del sistema Gestión de Usuarios no está marcado por el número de usuarios, sino por la cantidad de autenticaciones/autorizaciones que son requeridas por el resto de los sistemas y aplicaciones de la Plataforma.

El sistema Gestión de Usuarios actúa siempre en conjunto con el resto de las plataformas. En consecuencia la correcta evaluación de su desempeño debe efectuarse en las condiciones más reales posibles. Por esta razón la estrategia fue efectuar las pruebas de rendimiento en conjunto con las pruebas de las transacciones más habituales.

Las pruebas se efectuaron inyectando masivamente transacciones del sistema SCI, las que hacen uso del sistema Gestión de Usuario, tanto en autenticación como autorización.

La inyección masiva y escalonada de transacciones busca emular el comportamiento de la red de sucursales de Fonasa, durante su funcionamiento habitual. Se aplicaron escenarios de demanda "'normal" y demandas en horarios peak.

Para materializar la simulación se decidió utilizar un sofware específico de "inyección" controlada de transacciones. El *IBM Rational Performance Tester*.

Tolerancia a Fallas

Las pruebas de Tolerancia a Fallas buscan determinar el comportamiento de la aplicación ante "caídas" de diversa índole. Se definieron escenarios en los cuales se consideran fallos de componentes de software y fallos de hardware.

La aplicación debe ser capaz de responder de dos formas posibles:

- ▷ continuar respondiendo (falla transparente), existiendo una posible disminución del rendimiento
- ▷ recuperar su estado "normal" de operación en un período de tiempo "pequeño" (de acuerdo a los SLA comprometidos)

Seguridad

Las pruebas de seguridad corresponden, por un lado, a temas de redes, intromisión, denegación de servicio, etc. que abarcan la red y plataforma como un todo. Estas pruebas no son específicas de un sistema, y por tanto fueron efectuadas separadamente, por un grupo de trabajo independiente.

La prueba específica de seguridad, que corresponde al sistema Gestión de Usuarios, es precisamente la funcionalidad propia del sistema, es decir, se remite a comprobar su correcto funcionamiento, de acuerdo a la configuración de los roles propios de TIM (Tivoli Identity Manager), los permisos versus roles asociados a las aplicaciones (TAM), y los servicios de Autenticación y Autorización.

Adicionalmente, a efectos de cumplimiento del Accounting, la estrategia definida fue verificar la correcta generación de reportes y logs del sistema, buscando en los mismos transacciones que hubiesen sido efectuadas en algún determinado horario, y verificando que efectivamente dichas transacciones quedasen registradas.

Instalación y Convivencia

La aplicación Gestión de Usuarios debe operar coherentemente en las plataformas de Fonasa y coexistir con las actuales aplicaciones, sin que se produzcan fallos funcionales en ninguna de las aplicaciones.

La estrategia de prueba fue efectuar simulaciones de comportamiento del sistema Gestión de Usuarios en conjunto con las aplicaciones y transacciones más frecuentes de Fonasa, realizadas en los ambientes de escritorio (desktop) oficiales.

El resultado esperado es que no existan fallas funcionales de ninguna especie, ni en el sistema Gestión de Usuarios ni en las aplicaciones consideradas.

Respaldo y Recuperación

Efectuar pruebas de respaldo y restauración de toda la configuración, incluyendo los ambientes de Gestión de Usuarios.

Site 1 y 2

Efectuar pruebas de alternancia y operación de los sites. Se consideran todas las aplicaciones, incluyendo el sistema Gestión de Usuarios.

5.3.3. Estrategia Pruebas Piloto

La estrategia definida para las pruebas del Piloto fue utilizar un subconjunto de las pruebas funcionales que cumpla con el mínimo suficiente para asegurar la correcta operación. Entre las pruebas se debe incluir la operación de los funcionarios normales así como las funcionalidades asociadas a los jefes de Sucursal.

Además se debe considerar la participación de la Mesa de Ayuda y efectuar mediciones de tiempos, a objeto de determinar la capacidad de absorción de la demanda inicial de consultas, la que se espera sea alta durante los primeros días de funcionamiento de la nueva plataforma (Esto es, Gestión de Usuarios y los sistemas de la Primera Etapa operando conjuntamente).

5.3.4. Diseño de Pruebas (x casos)

La especificación detallada de cada prueba se estableció en base a documentos formales acordados con Fonasa.

Los documentos en su marco general establecen:

- ▷ Condiciones previas (requisitos para poder efectuar la prueba)
- ▷ Detalle de la prueba
- ▷ Casos de prueba
- ▷ Set de datos para cada caso de prueba
- ▷ Toma de evidencias
- ▷ Conclusiones
- ▷ Acuerdos

El detalle de cada documento es bastante extenso, de modo que sólo incluiremos en los Anexos, a modo de ejemplo, algunos cuyos aspectos resultan más relevantes.

En el Anexo 4 se muestra un extracto del Plan de Pruebas.

En el punto 4.2 del Anexo se muestran las pruebas del Plan de Convivencia.

En el punto 4.3 del Anexo se muestra el Plan de Pruebas asociado al Paso a Piloto.

5.4 Ejecución de Pruebas

Se ejecutaron pruebas exitosas en los ambientes de Test.

Luego se procedió a efectuar pruebas en los ambientes de Producción.

La ejecución de las pruebas No funcionales resultó ser la más compleja, precisamente porque interfería con las pruebas funcionales. Por ejemplo, las pruebas de Tolerancia a Fallas y Alta Disponibilidad requería simular "caídas" de la plataforma, lo que entorpecía las pruebas de las aplicaciones.

Fue necesario segmentar los horarios de pruebas y ejecutar las PNF en horarios fuera del horario normal de los grupos de desarrollo.

En definitiva, superadas las dificultades, las pruebas se efectuaron satisfactoriamente. El balance es el siguiente:

- Funcionales
  Se efectuaron las actividades de prueba planificadas, tanto unitarias como integradas. Luego de un necesario período de ajustes e iteraciones se obtuvieron resultados exitosos en todos los casos.
- No Funcionales
  Se efectuaron las actividades de prueba planificadas, también con resultados exitosos.
- Piloto
  Las pruebas de Piloto sólo se iniciaron una vez ejecutadas las pruebas funcionales y no funcionales. Una vez iniciadas las actividades de Piloto se efectuaron las pruebas definidas, con resultados exitosos.

5.4.1. Resultado de las Pruebas

Se muestra ahora el resultado de las diferentes pruebas efectuadas para certificar el funcionamiento y calidad del sistema Gestión de Usuarios. Esto corresponde a las nóminas de los casos de prueba, cantidad mínima de casos probados y los resultados obtenidos. Estas nóminas también fueron utilizadas para efectuar pruebas de regresión.

Los datos se disponen en tablas cuyas columnas son autoexplicativas. La columna "min OK" indica la cantidad mínima de casos que deben estar correctos para considerar que la prueba es satisfactoria.

Todas las pruebas se efectuaban en primera instancia, en reuniones de trabajo internas. Luego se acudía a su certificación y aprobación en sesiones con usuarios. Para que una prueba se considerase correcta se debía cumplir en dicha sesión con la ejecución del número mínimo de pruebas y todos los resultados estar correctos (ok). Además, en dichas pruebas no debían aparecer fallas (resultados incorrectos o ko).

Funcionales Unitarias

En las tablas 5.1 y 5.2 puede verse el resultado de las pruebas funcionales unitarias que fueron efectuadas.

Caso de Prueba	Subcaso	Resultado Esperado	Resultado Obtenido	mín OK
CRUD Usuarios				
Creación de usuarios	Crear usuario existente	Sistema debe rechazar la acción.	Se verifica rechazo.	2
	Crear usuario nuevo	Usuario debe ser creado correctamente y debe quedar en status de primera conexión.	Se verifica que el nuevo usuario reciba mail con su usuario y contraseña. Se pasa al caso de prueba Primera conexión (ingreso directo a SCI o a Gestión de Usuarios).	10
Cese de funciones	Eliminar usuario inexistente	Sistema debe rechazar la acción.	Se verifica rechazo.	2
	Eliminar usuario existente	Usuario debe ser eliminado correctamente, tanto a nivel de usuario como cuentas.	Se verifica eliminación mediante rechazo de conexión tanto en SCI como Gestión de Usuarios.	3
Desbloqueo contraseña				
Desbloqueo	Desbloqueo	Sistema debe aceptar el desbloqueo	Usuario verifica cambio de contraseña	2
Gestión de Roles				
Solicitar roles	Solicitudes sin owner	Sistema debe autorizar inmediatamente la solicitud.	Se verifica autorización. Para certificar nueva funcionalidad en sistema SCI se pasa a caso de prueba Conexión normal.	2
	Solicitudes con owner	Sistema debe quedar a la espera de aprobación/rechazo.	Se verifica que solicitud aparece a la espera de resolución por parte de alguno de los owner. Además, al ingresar a SCI las funciones del usuario no deben tener cambio. Se pasa al caso de prueba Aprobación/Rechazo de solicitud.	6
	Eliminación de rol para un usuario	Sistema debe aceptar inmediatamente la solicitud.	Para verificar la eliminación del rol se pasa a caso de prueba Conexión normal.	2
Autorizar/rechazar roles	Autorizar rol	Sistema debe aceptar la acción.	Se verifica aceptación y se pasa a caso de prueba Conexión normal, para verificar los nuevos permisos del usuario.	4
	Rechazar rol	Sistema debe aceptar la acción (el rechazo de la solicitud)	Se verifica aceptación y se pasa a caso de prueba Conexión normal (el usuario no debe experimentar ningún cambio en sus permisos).	2

Tabla 5.1: Resultado pruebas funcionales Unitarias, (1) de (2)

Caso de Prueba	Subcaso	Resultado Esperado	Resultado Obtenido	mín OK
Autoatención				
Primera conexión	Ingreso a sistema SCI	Sistema debe rechazar conexión y enrutar hacia sistema Gestión de Usuarios.	Se verifica rechazo y se pasa al caso de pruebas Primer Ingreso a sistema Gestión de Usuarios.	10
	Primer ingreso a sistema Gestión de Usuarios	Sistema debe aceptar conexión con usuario/password proporcionada. Además debe solicitar cambio de contraseña y llenado de preguntas de autenticación.	Se efectúa correctamente procedimiento primera conexión y se pasa al caso de prueba Conexión normal.	10
	Carga Masiva: Primer ingreso a sistema Gestión de Usuarios	Sistema debe operar en la misma forma que usuarios creados via on line.	Se efectúa correctamente procedimiento primera conexión y se pasa al caso de prueba Conexión normal.	4
Cambio preguntas autenticación	Cambio respuestas	Sistema debe aceptar cambio y registrar nuevas respuestas ingresadas.	Se verifica cambio y se pasa a caso de prueba Olvido contraseña.	2
Cambio contraseña	Asignar contraseña que no cumple criterios	El sistema debe rechazar los cambios (se deben probar diferentes escenarios de rechazo: largo, caracteres, etc)	Se verifica rechazo.	6
	Asignar contraseña consistente	El sistema debe aceptar el cambio.	Se verifica el cambio y se pasa a caso de prueba Conexión normal.	2
Olvido contraseña	responder correctamente preguntas	Sistema debe solicitar respuesta a las preguntas y aceptar el cambio.	Se verifica cambio y se pasa a caso de prueba Conexión normal.	2
	responder incorrectamente preguntas	Sistema debe solicitar respuesta a las preguntas y rechazar el cambio.	Se verifica rechazo.	2
Migración de datos				
Caarga masiva	Carga de un archivo de muestra	Sistema debe procesar archivo y generar archivos de resultados.	Se verifican informes generados y la creación de usuarios/cuentas en el sistema. Se pasa a caso de prueba Primera conexión para verificar correcta operatoria.	100 %
	Carga archivo con diferentes tipos de usuario	Sistema debe procesar archivo y generar archivos de resultados.	Se verifica generación de informes y la cantidad de usuarios/cuentas creados en el sistema.	100 %
	Carga de una sucursal completa	Sistema debe procesar archivo y generar archivos de resultados.	Se verifica generación de informes, los rechazos y la cantidad de usuarios/cuentas creados en el sistema.	99 %

Tabla 5.2: Resultado pruebas funcionales Unitarias, (2) de (2)

Funcionales Integradas

En las tabla 5.3 puede verse el resultado de las pruebas funcionales integradas.

Caso de Prueba	Subcaso	Resultado Esperado	Resultado Obtenido	mín OK
Conexión normal				
Conexión normal	Ingreso simple a sistema SCI	Conexión con funcionalidad habilitada	Se efectuó ingreso a sistema SCI y se verifica la correspondiente funcionalidad de acuerdo al perfil del usuario	10
	Conexión usuario con modificación de roles	Sistema debe aceptar ingreso y SCI debe reflejar los cambios de rol del usuario.	Se verifican cambios en la funcionalidad.	6
	Conexión usuario con rechazo en su modificación de roles	Sistema debe aceptar ingreso y no deben presentarse cambios en los permisos del usuario.	Se verifica ingreso y la permanencia de los permisos.	2
	Conexión usuario con eliminación de roles	Sistema debe aceptar ingreso y SCI debe reflejar la eliminación de rol del usuario.	Se verifican cambios en la funcionalidad.	2
	Conexión usuario con modificación de contraseña	Sistema debe aceptar ingreso con la nueva contraseña.	Se verifica ingreso.	4
	Conexión usuario eliminado	Sistema debe rechazar la conexión.	Se verifica rechazo.	1
	Conexión usuario creado via carga masiva	Sistema debe aceptar ingreso y operar normalmente, en forma idéntica a la creación de usuarios on line.	Se verifica conexión.	4
Web services	Conexión servicio no autorizado	"request refused"	Se verifica rechazo a servicio web.	3
	Conexión servicio autorizado	Sistema debe aceptar conexión	Se verifica la respuesta al servicio web. Se chequean logs con registro correcto de las transacciones.	3

Tabla 5.3: Resultado pruebas funcionales Integradas

Pruebas No Funcionales

Veremos ahora el resultado de las pruebas no funcionales.

Estas pruebas están orientadas a la plataforma en su conjunto, es decir, se consideran todas las aplicaciones involucradas (entre las que se encuentra Gestión de Usuarios) junto con el hardware y software que las sustentan.

Estas pruebas tienen muchísimos aspectos y detalles, y se genera una gran cantidad de información de estadísticas, mediciones y verificaciones, por lo que como resultados nos centraremos en aquellos aspectos más relevantes y medulares de cada prueba.

En la tabla 5.4 se puede ver el resultado de las pruebas no funcionales de rendimiento.

Estas pruebas requieren la inyección masiva de transacciones, para lo cual se utilizó el IBM Performance Tester, alimentado de archivos planos con diferentes tipos de transacciones. Los tipos de transacciones utilizados son los que -en acuerdo con los usuarios- se consideran más representativos. Es importante poner especial cuidado en que las transacciones hagan uso de distintos segmentos de la base y con distintos usuarios, de modo de simular un escenario lo más cercano posible a la realidad.

También se requiere verificar los resultados de la acción de las transacciones, para asegurar que todo está operando correctamente. Dado su volumen no es posible certificar estos resultados en pantalla, de modo que es necesario revisar los logs correspondientes y los efectos en las bases de datos.

Otro aspecto relevante a considerar es la tasa de arribo de transacciones, los horarios peak y valle esperados, y efectuar simulaciones en condiciones normales y en condiciones de mayor exigencia, a objeto de determinar los rangos de tolerancia de la plataforma.

Todas las transacciones utilizan la componente de Gestión de Usuarios correspondiente a los servicios TAM, es decir, identificación y autorización, que son precisamente las que serán sometidas a mayor exigencia durante el uso habitual del sistema.

Las pruebas de rendimiento hacen surgir problemas del desarrollo de software y de contención en diferentes puntos de la arquitectura. Esto últimos habitualmente se resuelven aumentando tamaños de buffers, número de conexiones concurrentes, cantidad de hilos en las aplicaciones, índices en las bases de datos, etc.

Caso de Prueba	**Subcaso**	**Resultado Esperado**	**Resultado Obtenido**	**mín OK**
Rendimiento				
Web Services	Consultas de información.	Se utilizaron 2 tipos de servicios web distintos. Se espera tiempo de respuesta menor a 1 segundo.	Se enviaron ráfagas de transacciones a razón de 5, 10, 50, 60 hasta 100 por segundo, en un lapso de 5 minutos. Los tiempos de respuesta fueron inferiores a los 200 milisegundos.	2
Sistema SCI	Consultas y actualizaciones de información.	Se utilizaron 5 tipos de consultas distintos y 1 transacción del tipo de actualización. Se espera tiempo de respuesta menor a 1 segundo en las consultas y menor a 5 segundos en las actualizaciones.	Se enviaron ráfagas de transacciones a razón de 5, 10, 50, 60 hasta 100 por segundo, durante un lapso de 5 minutos. 3 consultas estuvieron con promedios menores a 1 segundo. Una consulta y la actualización presentaron tiempos mayores y fueron devueltas a los desarrolladores e Infraestructura para su mejora. Finalmente todas las transacciones obtuvieron rendimientos dentro de los tiempos acordados.	6

Tabla 5.4: Resultado pruebas no funcionales de Rendimiento

En la tabla 5.5 se puede ver el resultado de las pruebas no funcionales de tolerancia a fallas.

El objetivo es asegurar que ante fallas emergentes, la plataforma continúa su operación dentro de los parámetros acordados. Las "caídas" se simulan bajando componentes de hardware (servidores) o de software completos.

Las pruebas más exigentes consistieron en bajar en forma simultánea varios servicios. Los resultados que se muestran corresponden precisamente a esos casos.

Caso de Prueba	**Subcaso**	**Resultado Esperado**	**Resultado Obtenido**	**mín OK**
Tolerancia a Fallas				
Hardware	Comunicaciones	Tiempos de restauración menores a 1 minuto	Se baja uno de los switch de redes. El tráfico se desvía automáticamente hacia los otros switch. No se detectan interrupciones en el servicio. Se sube el switch y se repite la prueba bajando otro switch, obteniendo resultados similares.	2
Hardware/Software	Componentes activo-activo	Tiempos de restauración menores a 1 minuto	Se bajan componentes de hw/sw de la plataforma. El tráfico se desvía automáticamente hacia las componentes activas. Se producen delays de segundos en la respuesta de las transacciones (3-15s) durante el período de bajada, y luego de unos minutos (2-3 m) las respuestas se normalizan. No se detectan interrupciones en el servicio. Se suben las componentes detenidas y se repite la prueba bajando el segundo grupo de componentes activas. Los resultados son similares.	2
	Componentes activo-pasivo	Tiempos de restauración menores a 1 minuto (activo) y menores a 5 minutos (pasivo)	Se bajan componentes de hw/sw de la plataforma, de la parte pasivo. Se reciben mensajes de alerta y cesan las replicaciones. No se detectan delays ni interrupciones en el servicio. Se suben las componentes detenidas y se repite la prueba bajando componentes pasivas (una componente a la vez). Los servicios pasivos emiten mensajes de alerta y cambian rol a activos, en un lapso de 3-4 minutos. Algunas transacciones dan time-out durante el cambio. Luego de la activación la plataforma se normaliza y las transacciones vuelven a operar normalmente.	2

Tabla 5.5: Resultado pruebas no funcionales Tolerancia a Fallas

En la tabla 5.6 se observa el resultado de las pruebas no funcionales de instalación y convivencia.

Caso de Prueba	Subcaso	Resultado Esperado	Resultado Obtenido	mín OK
Instalación y Convivencia				
Operación normal de Oficina	operación previa usando sistemas actuales.	Se ejecutan las transacciones más frecuentes de una sucursal, incluyendo impresión de documentos. Se espera operación normal.	Operación sin contratiempos.	8
	transacciones SCI de consulta	Se incluye la generación e impresión de formularios. Se espera funcionamiento normal.	Se obtienen resultados correctos. Los formularios se imprimen correctamente en diferentes dispositivos.	10
	Gestión de Usuarios	Se incluyen transacciones de creación de usuarios, primera conexión y modificación de contraseñas. Se espera operación normal.	Operación correcta.	4
	operación posterior	Se repiten las pruebas efectuadas inicialmente y se agregan algunas adicionales. Se espera operación normal.	Operación sin contratiempos. Los resultados indican que no existe evidencia de incompatibilidad.	15

Tabla 5.6: Resultado pruebas no funcionales Instalación y Convivencia

En la tabla 5.7 se observa el resultado de las pruebas no funcionales de respaldo y recuperación.

Caso de Prueba	Subcaso	Resultado Esperado	Resultado Obtenido	mín OK
Respaldo y Recuperación				
Operación previa	Base de Datos	Se ejecutan transacciones sobre la Base de Datos elegida para prueba. Se espera operación normal.	Las transacciones se ejecutan correctamente.	10
	File System	Se agregan archivos, algunos se modifican y otros se eliminan. Otros archivos se dejan sin cambios.	Operaciones efectuadas.	8
Respaldo	Respaldo	Respaldo completo de Base de Datos y File System sin inconvenientes.	El respaldo se ejecuta completo. Se verifican tamaños y cantidades de registros. Los resultados son correctos.	2
Recuperación (restauración)	Base de Datos	Se espera que la Base refleje los cambios efectuados.	Se verifican las transacciones y el contenido del resto de la Base. Se obtienen los resultados esperados.	10
	File System	Se espera que el File System refleje los cambios efectuados, y que el resto de los archivos esté correcto.	Se verifican los cambios y el contenido del resto del File System. Se obtienen los resultados esperados.	8

Tabla 5.7: Resultado pruebas no funcionales Respaldo y Recuperación

En la tabla 5.8 se observa el resultado de las pruebas no funcionales de alternancia de sites.

El site 2 presenta la misma infraestructura que el site 1, pero inicialmente no recibe transacciones. La información de las Bases de Datos y los File System se encuentra permanentemente sincronizada a través de mecanismos de réplica automáticos.

Cuando se produce una "caída" del site 1, como es lógico las réplicas de datos se interrumpen. El mecanismo de alternancia consiste en el "desvío" del flujo de tráfico hacia el site 2, el que entra en operación a partir de ese momento. Este proceso puede ser completamente automático o comportarse en modo semi-automático, pues depende del tipo de falla.

La "caída" se simula a través de bajada de nodos o dejando el site 1 sin comunicación.

Finalmente, la restauración del site 1 es un proceso bastante más largo, pues requiere reparar la falla del site, y ejecutar la resincronización y verificación de los datos. Una vez logrado, se ejecuta el procedimiento de restauración, el que -en lo esencial- es un proceso de alternancia de sites, en que se retorna el control y el flujo de datos hacia al site original.

Caso de Prueba	Subcaso	Resultado Esperado	Resultado Obtenido	mín OK
Site 1 y 2				
Site 1	operación previa	Se ejecutan diversas transacciones. Se obtienen estadísticas de las Bases de Datos. Se espera operación normal.	Las transacciones operan correctamente.	10
	simulación interrupción/caída servicio	se ejecuta simulacro de interrupción.	Se verifica interrupción de servicio en site 1.	1
Site 2	cambio switch	se espera alternancia de sites y recuperación de servicio en un lapso menor a 5 minutos.	Se observa ralentización del servicio y time-out de algunas transacciones. Servicio es restaurado en site 2 en un lapso de 3-5 minutos.	1
	verificación transacciones	se espera que las transacciones ejecutadas antes de la "caida" estén reflejadas en site 2.	Se verifican la presencia de las transacciones efectuadas. Se efectúan chequeos de consistencia de cantidades de registros.	10
	transacciones en site 2	Se ejecutan transacciones en site 2. Se obtienen estadísticas de las Bases de Datos. Se espera operación normal.	Las transacciones operan correctamente.	10
Site 1	restauración	Se espera que procedimiento de restauración opere correctamente.	Se levanta site 1, se restaura la información de las Bases de Datos y se ejecuta procedimiento de cambio de switch.	1
	verificación transacciones	se espera que las transacciones ejecutadas originalmente en site 1, así como las ejecutadas en site 2 estén presentes.	Se verifican ambos grupos de transacciones. Se efectúan chequeos exitosos de cantidades de registros.	20
	transacciones en site 1	Se ejcutan transacciones. Se espera operación normal.	Las transacciones operan correctamente.	5

Tabla 5.8: Resultado pruebas no funcionales Site 1 y 2

Pruebas Piloto

En las tablas 5.9 y 5.10 se observa el resultado de las pruebas para Piloto. Estas pruebas se repetían, con los respectivos datos, para cada nueva oficina incorporada.

Caso de Prueba	Subcaso	Resultado Esperado	Resultado Obtenido	mín OK
Migración de Datos				
Carga masiva	Carga de una sucursal completa	Sistema debe procesar archivo y generar archivos de resultados. El nivel de error no debe superar el 1 % de la cantidad de registros.	Se detectan algunos errores de datos en la carga inicial (códigos, email mal formados, etc). La carga se autoriza cuando se logra el 99 % ok. Observación: Los niveles de jefatura siempre deben estar 100 %	99 %
Pruebas Básicas				
Creación usuarios	Crear usuario nuevo	Usuario debe quedar en status de primera conexión	se verificó mail y se pasó a prueba Primera conexión	2
Cese de funciones	Eliminar usuario	Usuario debe ser eliminado tanto a nivel de usuario como cuentas.	Se verifica rechazo de conexión tanto en SCI como sistema Gestión de Usuarios.	1
Solicitar Roles	Sin owner	Sistema debe autorizar inmediatamente la solicitud.	Se verifica nueva funcionalidad disponible para el usuario en sistema SCI.	2
	Con owner	Sistema debe quedar a la espera de aprobación/rechazo.	Se verifica solicitud pendiente en alguno de los owner. Se pasa al caso de prueba Autorización/Rechazo de solicitud.	2
Autoriza/rechaza	Autorizar Rol	Sistema debe aceptar la acción.	Se verifica aceptación y se pasa a caso de prueba Conexión normal con autorización de nuevo rol.	2

Tabla 5.9: Resultado pruebas Inicio Piloto

Caso de Prueba	Subcaso	Resultado Esperado	Resultado Obtenido	mín OK
Autoatención				
Primera conexión	Ingreso a sistema SCI	Sistema debe rechazar conexión y enrutar hacia Sistema Gestión de Usuarios	Se verifica rechazo y se pasa al caso de pruebas Primer ingreso a sistema Gestión de Usuarios	2
	Primer ingreso a sistema Gestión de Usuarios	Sistema debe aceptar conexión con usuario/password proporcionada. Además debe solicitar cambio de contraseña y llenado de preguntas de autenticación.	Durante el piloto los usuarios debían efectuar esta actividad/prueba y reportar cualquier falla. Algunos usuarios observaron que no habían recibido el email (como solución se asignó contraseña manualmente). Otros tenían problemas en la conexión al sitio (se transfirió el request a Infraestructura para su solución), etc. La aprobación se logró cuando la cantidad de errores reportados no resuelto fue menor a 1 %. En los casos correctos se se efectuó procedimiento primera conexión, pasando enseguida al caso de prueba Conexión normal.	99 %
Conexión normal				
Conexión normal	Ingreso simple a sistema SCI	Conexión con funcionalidad habilitada	Los usuarios efectuaron sus operaciones normalmente, debiendo reportar cualquier falla o anomalía. Se detectaron algunos casos en que faltaba funcionalidad. La razón fue la asignación inicial, en la carga masiva, de un perfil informado incorrectamente al usuario. Estos casos fueron corregidos manualmente.	99 %
	Conexión usuario con modificación de roles	Sistema debe aceptar ingreso y SCI debe reflejar los nuevos permisos.	Se verifican cambios en la funcionalidad.	4
	Conexión usuario eliminado	Sistema debe rechazar la conexión.	Se verifica rechazo.	1

Tabla 5.10: Resultado pruebas Piloto efectuadas por los usuarios

5.5 Aprobación y Acuerdos

Concluida satisfactoriamente la etapa de pruebas Funcionales y No funcionales, se obtuvo la aprobación al sistema Gestión de Usuarios.

Posteriormente se iniciaron las actividades de Piloto, coordinadamente con el sistema SCI (los sistemas correspondientes a la Primera Fase).

En última instancia, con el paso a Piloto, las sucursales de Fonasa de todo el país se fueron incorporando paulatinamente al ambiente de Producción final.

Capítulo 6

Conclusiones

En este capítulo haremos un balance final del proyecto, las actividades realizadas y los logros obtenidos.

Además, se inserta una digresión respecto de otras aplicaciones y usos de LDAP en las organizaciones.

6.1 Balance

6.1.1. Objetivos logrados

Desde el punto de vista del Proyecto, resultó clave la temprana delimitación de alcances. En lo esencial, el proyecto Gestión de Usuarios planteaba originalmente dos aspectos: Primero, la seguridad de los accesos a las plataformas y segundo, la relación del sistema con RRHH.

El ajuste de expectativas, acordado con los stakeholders, se basó en dividir el proyecto Gestión de Usuarios en dos etapas: Primero lograr los aspectos de seguridad y luego, como segunda etapa, desarrollar los aspectos y flujos de RRHH. Esto permitió focalizar los esfuerzos en los aspectos más estructurales de la solución, desarrollar una mecánica de pruebas mucho más exigente y sentar las bases para las siguientes etapas.

Establecidos estos acuerdos, en definitiva se lograron todos los objetivos planteados. Los usuarios tuvieron activa presencia, tanto en la etapa de definiciones, durante el desarrollo, y en la ejecución de pruebas. Las pruebas efectuadas fueron exhaustivas, en los aspectos funcionales como los no funcionales. La migración de datos fue oportuna y se logró dar inicio al Piloto en condiciones óptimas. Se efectuaron extensas jornadas de capacitación y difusión, y el paso a producción fue paulatino y con un mínimo de trastornos para las áreas usuarias.

El sistema Gestión de Usuarios se encuentra operando y ya se dio inicio a la segunda etapa de los desarrollos.

Desde el punto de visto de la Memoria, se lograron los objetivos específicos. Por una parte analizar comparativamente dos opciones de LDAP, generando un informe con resultados y criterios de evaluación. Por otra, aplicar exitosamente la tecnología LDAP, y llevar a buen puerto el desarrollo del proyecto, pasando por todas las etapas metodológicas planteadas originalmente. Esto es, establecer un sólido modelo conceptual, y efectuar los levantamientos, análisis, diseño, construcción, pruebas e implantación, en forma ordenada y consistente con la planificación.

6.1.2. Objetivos no logrados y/o diferidos

En lo esencial, y en base a los acuerdos de alcance establecidos con Fonasa, se lograron todos los objetivos propuestos. Sin embargo, una mirada en retrospectiva indica que puede efectuarse algunas mejoras.

En particular se puede hacer simplificaciones al modelo de permisos. Esto tanto desde el punto de vista de procedimientos como en la estructura de grupos dentro del DIT.

En el modelo de esta primera etapa es posible asignar permisos tanto directamente a los usuarios como a los roles. En mi opinión, una mejor práctica es asignar permisos sólo a roles y

que los usuarios simplemente adquieran los roles que necesitan, pero nunca en forma directa.

La razón es la siguiente:

En el modelo actual para efectuar autorización existen dos posibilidades: Que el usuario pertenezca a un rol que tiene acceso al recurso solicitado, o bien que el usuario tenga directamente acceso a dicho recurso. Para fines de control y auditoría es mejor y más simple que sólo perviva la primera opción. Eso evita la presencia de asignaciones de permisos directos por cortos períodos, que luego al ser revocados son difíciles de pesquisar.

Además, un único modelo de asignación de permisos es proceduralmente más simple, ya que es más estructurado, obliga a un mayor ordenamiento de roles en la organización, y también presenta ventajas desde el punto de visto del diseño del DIT y del algoritmo de autorización.

6.1.3. Claves del Éxito del Proyecto

Un balance no estaría completo si no se revisaran las razones de su éxito o de sus contratiempos. En este caso, el proyecto se llevó a cabo en los plazos y alcances definidos, de modo que resulta útil detallar las razones de este resultado. En mi opinión, ello se debe a:

- ▷ Metodología formal
- ▷ Establecimiento de acuerdos con Fonasa
- ▷ Definición de fases y alcance claro en cada fase
- ▷ Utilización de tecnologías (LDAP y COTS de IBM). Evaluación alternativas y elección racional y documentada
- ▷ Establecer la Gestión de Usuarios como una componente activa de la seguridad de la organización
- ▷ La Arquitectura y los Requerimientos: Actividades simultáneas y complementarias
- ▷ Migración de datos en paralelo con el Proyecto
- ▷ Plan de pruebas formal y detallado
- ▷ Estrategia de Piloto y puesta en Producción definidos como hitos del Proyecto

6.2 Crecimiento de la Aplicación

6.2.1. Siguientes fases del proyecto Gestión de Usuarios

Integración con sistema de Recursos Humanos

El sistema Gestión de Usuarios fue diseñado de tal forma que la integración con RRHH sea natural y expedita.

RRHH está orientado a la administración y control de los recursos, y a una correcta ejecución de los flujos y procedimientos asociados. Desde esta perspectiva, la información ya se encuentra residente en la base LDAP, y los necesarios controles y estadísticas pueden obtenerse directamente a través de los medios proporcionados.

Además, los flujos y procedimientos de RRHH se pueden acoplar a los flujos ya construidos de creación/cese de usuarios, y asignación de roles y permisos, para formar un flujo de procesos de mayor alcance.

De esta forma, el sistema de RRHH puede hacer uso de los recursos y servicios proporcionados por Gestión de usuarios. Esto es, su desarrollo puede visualizarse como un crecimiento de los elementos y flujos ya creados en la primera etapa.

Integración con nuevas Aplicaciones de la Plataforma

A medida que el proyecto global de Fonasa avance, se irán integrando nuevos sistemas y aplicaciones. En este contexto es importante señalar que el sistema Gestión de Usuarios fue diseñado de tal forma que la integración con estas nuevas aplicaciones sea simple y natural.

¿Cómo se logra garantizar esta facilidad de integración?

Los procedimiento están claramente definidos y documentados. Esto en base a acuerdos formales establecidos con los grupos de desarrolladores de aplicaciones. En lo esencial, el procedimiento es el siguiente: Los nuevos sistemas deben presentar su matriz de roles/recursos para agregar los elementos adicionales (posibles nuevos roles y los nuevos recursos) en el repositorio LDAP y en los permisos TAM. Además los nuevos sistemas deben hacer uso de los servicios de autenticación/autorización que se encuentran publicados.

Usuarios externos

Un escenario interesante es el de la administración de usuarios tipo internet y/o público en general. ¡Todo ciudadano de Chile como potencial usuario!

En este caso el modelamiento y las reglas de autenticación/autorización son distintos. Además puede requerirse integración o conectividad con fuentes externas para validar información (por ejemplo, gabinete de identificación).

En mi opinión se trata de un proyecto esencialmente diferente. Los factores relevantes son la masividad, la agilidad en el registro, utilización de medios de verificación electrónicos tipo correo, y políticas de administración y permisos específicos. Desde un punto de vista LDAP, los contenidos de las entries, el DIT e incluso la arquitectura de la solución requeriría un enfoque de solución propio y particular.

6.2.2. Posibles usos Futuros de LDAP en Fonasa

LDAP fue utilizado como eje fundamental para el desarrollo de la Gestión de Usuarios de Fonasa. Sin embargo, existen otros ámbitos donde también podría resultar de utilidad. LDAP puede ser usado como repositorio directo de información (para información con bajo nivel de cambios), o como directorio para indicar la ubicación de otras fuentes de información,

Por ejemplo:

- ▷ Centralización de información.
  Datos de utilidad general, información de uso público
- ▷ Repositorio para localización de los recursos de la plataforma.
  Impresoras, pcs, dispositivos biométricos
- ▷ Parámetros globales de la organización.
  Datos propios de la organización, tablas de uso frecuente

6.3 Metodología para un Sistema Gestión Usuarios

Uno de los subproductos más interesantes de esta Memoria es el esbozo de una metodología para el desarrollo e implantación de un sistema Gestión de Usuarios en una organización.

Desde el punto de vista metodológico, un proyecto Gestión de Usuarios no difiere de cualquier otro proyecto de software.

Sin embargo cruza todos las áreas sensibles de la empresa:

- ▷ ámbito del Negocio
- ▷ Operación
- ▷ Gestión

desde ese punto de vista, la Gestión de Usuarios es un sistema de Misión Crítica para la organización.

Cada sistema de Gestión de Usuarios es distinto. Pero todos presentan aspectos e intereses que son comunes:

- ▷ estrechamente vinculado con la seguridad
- ▷ existencia de patrones
- ▷ disponibilidad de software específico para esta área
- ▷ relación con los sistemas de RRHH

Por ello, la definición de una metodología de desarrollo e implantación especialmente focalizada en esta materia son altamente deseables. Esto permite guiar el desarrollo hacia áreas que son esenciales, evitar la omisión de aspectos relevantes, y posibilita adelantar problemas y afrontar su solución anticipadamente.

6.3.1. Análisis Inicial para un sistema Gestión de Usuarios

Aunque existe literatura al respecto, el desarrollo de un sistema de Usuarios no puede limitarse a un largo checklist de ámbitos y opciones. Esto por supuesto también es necesario, pero se requiere una mirada holística del problema y las expectativas de alto nivel de los stakeholders de la organización.

En mi opinión, es requisito establecer las bases conceptuales para el modelamiento de la solución. Entre otros, los elementos a considerar son los elementos de seguridad y la utilización de patrones. En este sentido, el libro de Schumacher y otros es una excelente referencia [1]

Otra fuente de información es el blog de patrones de seguridad (como lo declaran en el propio blog, "dedicado a los fans de la seguridad"). Se trata de un blog sumamente activo y permanentemente actualizado con información y noticias acerca del tema [2].

Adicionalmente conocer las tecnologías disponibles, experiencias anteriores y las diferentes posibilidades de aplicación. LDAP ha probado ser un modelo exitoso, pero es necesario estar al tanto de sus ventajas y limitaciones.

6.3.2. Alternativas: LDAP es una excelente opción

¿Por qué usar LDAP?

LDAP es más que un protocolo. Se ha constituido en un estilo para construir determinadas aplicaciones. En la práctica, establece un framework para modelar y definir ciertos problemas, en un lenguaje estándar y con reglas claras y precisas.

El uso de estándares promueve la compatibilidad entre herramientas y el intercambio fluido y consistente de información.

[1] Schumacher *et al.* [16]
[2] collectors [5]

LDAP se encuentran disponible en una amplia variedad de herramientas, tanto open source como comerciales.

Los mecanismos de interconexión e integración son estándares. Múltiples aplicaciones ya están implementando la conectividad hacia LDAP. Además existe conectividad en un espectro creciente de lenguajes computacionales (Java, Python, etc).

Además existe una comunidad activa que está estudiando y promoviendo mejoras y ampliaciones al protocolo, así como descubriendo nuevas formas de utilización.

Y por último enfatizar nuevamente en los atributos de calidad que proporciona LDAP: Rendimiento, escalabilidad, seguridad, integridad, alta disponiblidad y tolerancia a fallas.

6.3.3. La Metodología

¿En qué consiste esta metodología?

En lo esencial, se trata de un conjunto de pasos a seguir, en que se da mayor énfasis a áreas específicas del tema Gestión de Usuarios.

- ▷ Levantamiento de requerimientos
- ▷ Establecer acuerdos con los Usuarios
  - – Calidad del Servicio (SLA esperados)
  - – Definir Niveles de error aceptables en los diferentes ámbitos
- ▷ Analizar alternativas de implementación directa o uso de COTs
  - – Privilegiar el uso de COTS
  - – Verificar aplicabilidad de LDAP
  - – Estudiar LDAP disponibles y el grado de cobertura sobre los requerimientos
- ▷ Utilización de Patrones de Seguridad
  - – definir ámbitos del Sistema (intranet, extranet, internet)
  - – definición del modelo de datos para seguridad en la Organización (DIT)
  - – utilización de plantillas (definición del modelos de manejo de passwords)
- ▷ Separación de intereses, tanto al efectuar el Análisis, como en la Implementación
- ▷ Análisis
  - – Estimación de tamaño
    - ∗ Definición de volúmenes de datos persistentes
    - ∗ Cantidad y tipo transacciones. Horarios peak y valle
  - – Análisis guiado por la triple A (Autenticación, Autorización, Accounting)
  - – Funcionalidad de Identificación/Autenticación
  - – Funcionalidad de Autorización
  - – Funcionalidad de Accounting (idealmente centralizar el accounting)
  - – Atributos de calidad

        * Rendimiento
        * Alta disponibilidad
- ▷ Definir un proyecto paralelo de Migración de Datos
    - Estrategia de Migración
    - Acordar niveles de error aceptables
    - Definir origenes de los datos y mecanismos de validación y filtro
    - Desarrollo de software o uso de software ETL
    - Plan de pruebas específico para la Migración
- ▷ Definir un plan de pruebas funcionales y No funcionales
- ▷ Definir espacio para tareas de Gestión del Cambio
- ▷ Utilización de Piloto como estrategia de Implantación y Puesta en Marcha

6.4 Digresión: Otros usos y aplicaciones de LDAP

Si bien LDAP es una excelente opción para la implementación de sistemas de administración y/o gestión de Accesos y Usuarios, no es el único ámbito en que LDAP resulta apropiado.

6.4.1. Aplicaciones conocidas

- ▷ Gestión de Usuarios
- ▷ Internet (administración de urls)
- ▷ Machine Authentication
- ▷ User Authentication
- ▷ User/System Groups
- ▷ Address book
- ▷ Organization Representation
- ▷ Asset Tracking
- ▷ Telephony Information Store
- ▷ User resource management (redes, impresoras, dispositivos y servicios de uso compartido, etc)
- ▷ E-mail address lookups
- ▷ Application Configuration store
- ▷ PBX Configuration store

6.4.2. Otras posibilidades

Los usos de LDAP pueden ser múltiples. El límite es simplemente la imaginación. Existe una comunidad activa investigando y desarrollando nuevas herramientas y aplicaciones en torno a LDAP. Las definiciones del protocolo se siguen perfeccionando y continuamente la normativa está recibiendo nuevos aportes e ideas de como mejorar y aumentar las especificaciones.

Además, recalcar que LDAP puede ser usado como repositorio directo o como directorio hacia otras fuentes de información.

Por ejemplo, un tema recurrente es la proliferación y duplicidad de tablas de códigos en las plataformas sistémicas de las organizaciones. Una misma tabla aparece duplicada una y otra vez en diferentes sistemas, con diferentes formatos y representaciones, pero -he aquí el punto- ¡conteniendo siempre la misma información! A lo mejor la tecnología LDAP tiene algo que decir en esta materia.

Glosario

aka
Also Known As, alias, denominación alternativa.

ANSI
American National Standards Institute, organización de Estados Unidos, de influencia mundial, dedicada al desarrollo de estándares. Se encuentra asociada a la ISO.

CMMI
Capability Maturity Model Integration. Conjunto de normas y buenas prácticas orientadas a identificar el nivel de "madurez" de las organizaciones dedicadas al desarrollo, mantención y operación de sofware, desde una perspectiva del análisis de los procesos que la empresa utiliza para llevar a cabo estas actividades.

COTS
Comercial Off The Shelf. En el ámbito del software se refiere a la utilización de productos o componentes, habitualmente comerciales, que satisfacen áreas de funcionalidad específicas, perfectamente definidas y completas.

CRUD
Create, Read, Update and Delete.

DHCP
Dynamic Host Configuration Protocol.

DNS
Domain Name System. Sistema de nombres de recursos en una red privada o Internet.

EAPS
Establecimiento de Atención Primaria de Salud.

Fonasa
Fondo Nacional de Salud. Organismo público del Estado de Chile, cuya función es recaudar, administrar y distribuir los recursos financieros del sector Salud, a objeto de dar cobertura y prestaciones a sus asegurados.

GTI
Area de Fonasa dedicada a la Gestión de Tecnologías de Información.

GUI
Graphical User Interface.

Help-Desk
Mesa de Ayuda.

IETF
Internet Engineering Task Force, organismo responsable de las definiciones del estándar LDAP.

ISO
International Standards Organization.

ITU-T (UIT-T)
International Telecommunication Unit. Organismo internacional dedicado a definir normativa, regulación y buenas prácticas, a nivel mundial, en el área de las Telecomunicaciones.

JNDI
Java Naming and Directory Interface.

LDAP
Lightweight Directory Access Protocol.

LDAP injection
Es un tipo de ataque de software que busca vulnerar la seguridad de un servidor LDAP, a objeto de obtener información no autorizada y/o efectuar modificaciones en la información del directorio.

LDIF
LDAP Data Interchange Format. Formato plano para intercambio y carga de archivos de datos desde/hacia un directorio LDAP.

look and feel
En el contexto de diseño de software se refiere a la apariencia de las interfaces gráficas, sus formatos, colores y la facilidad de uso.

OOP
Object Oriented Programming, Programación orientada a Objetos.

Outsorcing
Externalización de servicios de una empresa. En el contexto informático habitualmente se refiere a la utilización de diversos servicios externos de data center, housing, explotación, procesamiento, etc..

PMO
Project Management Office, Oficina de Administración de Proyectos.

PNF
Pruebas No Funcionales, corresponde a las pruebas asociadas a los requisitos de calidad.

RRHH
Recursos Humanos.

RUP
IBM Rational Unified Process.

SCI
Sistema Corporativo de Información. Corresponde al nombre que Fonasa utiliza para designar a su nueva plataforma. Los sistemas considerados en el SCI, en la Primera Fase del proyecto son Acreditación y Cotizaciones. Por supuesto, dentro de los sistemas de apoyo también está considerado el sistema Gestión de Usuarios..

SLA
Service Level Agreement. Acuerdo de nivel de servicios, entre una parte que actúa como cliente y la otra como proveedor.

UNICODE
Estándar universal de codificación de caracteres, mantenido por el Unicode Technical Committee (UTC), en el que participan diversas empresas, principalmente del ámbito informático.

X.500
Corresponde a un conjunto de especificaciones de redes de computadoras y servicios de directorio, definidas por la ITU-T.

XML
eXtensible Markup Language.

Bibliografía

[1] Apache. 2013. *The Apache Directory Project.* [en linea] <http://directory.apache.org/>[consulta: 6 agosto 2013].

[2] Arkills, Brian. 2003. *LDAP directories explained : an introduction and analysis.* Boston, MA ; London: Addison-Wesley.

[3] Butcher, Matt. 2007. *Mastering OpenLDAP: Configuring, Securing and Integrating Directory Services.* Packt Publishing.

[4] Carter, Gerald. 2003. *LDAP system administration.* Sebastopol, Calif. ; Farnham: O'Reilly.

[5] collectors, security patterns. 2013. *Security Patterns. A Blog dedicated to security enthusiasts.* [en linea] <http://www.securitypatterns.org/>[consulta: 6 agosto 2013].

[6] Donley, Clayton. 2003. *LDAP programming management and integration.* Greenwich, Conn. ; [Great Britain]: Manning.

[7] Guttman, Barbara, & Roback, Edward. 1995. *An introduction to computer security: The NIST handbook.*

[8] Howes, Tim, Smith, Mark, & Good, Gordon S. 2003. *Understanding and deploying LDAP directory services.* 2nd ed. edn. Boston ; London: Addison-Wesley.

[9] IBM. 2007. *Enterprise security architecture using IBM Tivoli security solutions.* 5th edn. IBM redbooks. Poughkeepsie, NY: IBM, International Technical Support Organization.

[10] IBM-LDAP. 2013. *Understanding LDAP - Design and Implementation.* [en linea] <http://www.redbooks.ibm.com/redbooks/SG244986/wwhelp/wwhimpl/js/html/wwhelp.htm>[consulta: 6 agosto 2013].

[11] IBM-TDS. 2013. *Tivoli Directory Server.* [en linea] <http://www-

03.ibm.com/software/products/us/en/directory-server/>[consulta: 6 agosto 2013].

[12] Microsoft. 2013. *Introduccion a Active Directory.* [en linea] <http://support.microsoft.com/kb/196464/es>[consulta: 6 agosto 2013].

[13] OpenLDAP. 2011. *OpenLDAP Software 2.4 Administrator's Guide.* [en linea] <http://www.openldap.org/doc/>[consulta: 6 agosto 2013].

[14] OpenLDAP. 2013. *OpenLDAP Software Home Page.* [en linea] <http://www.openldap.org/>[consulta: 6 agosto 2013].

[15] Oracle. 2013. *OpenDS. Open Source. Open Standards. Open Directory Service.* [en linea] <http://opends.java.net/>[consulta: 6 agosto 2013].

[16] Schumacher, Markus, Fernandez-Buglioni, Eduardo, Hybertson, Duane, Buschmann, Frank, & Sommerlad, Peter. 2006. *Security patterns : integrating security and systems engineering.* Chichester, Great Britain.: John Wiley. Wiley series in software design patterns.

[17] Stallings, William, & Brown, Lawrie. 2012. *Computer security : principles and practice.* 2nd edn. Boston: Pearson. (with contributions by Mick Bauer, Michael Howard).

Anexos

Anexo 1

La organización Fonasa

Breve historia institucional de la salud en Chile

Hacia 1979, el Servicio Médico Nacional de Empleados (SERMENA) y el Servicio Nacional de Salud (SNS) eran los dos organismos del Estado de Chile, responsables de la atención de las necesidades de salud de la población (el primero para empleados públicos y privados, y el segundo para obreros y población de menores recursos). Todos los chilenos estaban cubiertos por el sistema público de salud, y sólo aquellos que deseaban un mejor nivel de servicios acudían a profesionales del sector privado.

1979. Decreto Ley Nº 2.763. Se reestructura completamente el sector público de salud. Se disuelve SERMENA y SNS, y se crean nuevos organismos y funciones. Nace el Sistema Nacional de Servicios de Salud (SNSS) de Chile, el que se compone del Ministerio de Salud, las Secretarías Regionales Ministeriales de Salud (SEREMI), la red de Servicios de Salud y Centros de Referencia de Salud (CRS) de todo el país, el Fondo Nacional de Salud (Fonasa), el Instituto de Salud Pública (ISP) y la Central Nacional de Abastecimiento (CENABAST).

Las funciones de Fonasa establecidas en dicho decreto le confirieron la responsabilidad de recaudar, administrar y distribuir los recursos financieros. destinados a salud, originados en fondos estatales y de sus cotizantes.

1981. Aparecen dos nuevas reformas al sistema. La primera transfiere la administración de los Centros de Atención Primaria de Salud al gobierno local, representado por los Municipios. La segunda establece un sistema de privatización y elección individual de los servicios de

salud: La conocida ley de ISAPRES (Instituciones de Salud Previsional). Con ello se abrió la posibilidad de que las personas cotizaran su 7 % de salud en las instituciones privadas (ISAPRES) o en el sector público (Fonasa).

1990. Se promulga la ley 18.933, que crea la Superintendencia de Instituciones de Salud Previsional (ISAPREs), cuya función es supervisar y controlar a estas entidades.

2004. Se promulga la ley 19.966, de Acceso Universal a Garantías Explícitas (AUGE) cuyo objetivo es asegurar la atención de determinadas patologías, a todos los ciudadanos del país, independiente de su pertenencia al sistema público o privado.

2005. Nace la Superintendencia de Salud reemplazando a la Superintendencia de Isapres. Sus funciones son fiscalizar, en las materias de su competencia, tanto las Instituciones de Salud Previsional (ISAPRE), como al propio Fondo Nacional de Salud (Fonasa) y el Sistema de Salud de las Fuerzas Armadas.

El rol de Fonasa

Como lo declara en su sitio web, el Fondo Nacional de Salud, Fonasa, es el organismo público encargado de otorgar cobertura de atención, tanto a las personas que cotizan el 7 % de sus ingresos mensuales en Fonasa, como a aquellas que, por carecer de recursos propios, financia el Estado a través de un aporte fiscal directo.

A renglón seguido agregan: Sus funciones principales son: recaudar, administrar y distribuir los recursos financieros del sector salud; financiar las prestaciones de salud otorgadas a sus beneficiarios; identificar a los asegurados e informarles adecuadamente sobre sus derechos; conocer y resolver reclamos; fiscalizar las cotizaciones de salud y los recursos destinados a prestaciones.

FONASA da cobertura de salud a más de 13 millones de asegurados, sin exclusión de edad, sexo, nivel de ingreso, o enfermedades preexistentes.

Posee cobertura nacional a través de su Casa Matriz, sus cuatro Direcciones Zonales: Norte, Centro Norte, Centro Sur y Sur y su extensa red de oficinas y sucursales en todo el país.

Un desafío permanente: La modernización del aparato estatal

El gobierno de Chile está comprometido en la modernización de todas sus instituciones y reparticiones. Esto considera su operación, procedimientos, sistemas, estructura y un largo etcétera. Fonasa, siendo una institución pública, no está exenta de este compromiso y sus necesidades son múltiples.

Entre otros, desde los aspectos básicos de toda organización, hasta los específicos del área de negocios en que se encuentra: La correcta acreditación de sus beneficiarios, el registro,

administración y control de los recursos originados en las cotizaciones, la administración y fiscalizaciones de las prestaciones y quienes las materializan (prestadores de servicios), el intercambio de información con otras empresas y organismos del sector, etc.

Por esta razón, la mejora permanente de sus procesos, la optimización y eficiencia en sus operaciones, la evaluación y control de sus indicadores de gestión, responder adecuadamente a los entes fiscalizadores, disponer de las últimas y mejores tecnologías, etc. son objetivos y compromisos permanentes de Fonasa como organización.

Anexo 2

Adexus

Adexus, una empresa chilena de integración

Adexus es una empresa chilena dedicada a la integración de sistemas y las tecnologías de información y comunicaciones.

El inicio de sus operaciones se remonta a 1990, con el nombre Tandem Chile, y como representación en Chile de Tandem Computers. En 1997 Compaq adquiere el control de la empresa Tandem, y la representante chilena se escinde completamente, pasando a denominarse como Adexus, desde 1998.

Adexus provee una gran variedad de servicios, entre los que destacan la consultoría, el desarrollo de soluciones y especialmente el outsorcing, en diferentes modalidades. Hoy está presente en los más diversos sectores de actividad del país: Finanzas, telecomunicaciones, gobierno, educación superior, industria y comercio, servicios.

Adexus ha buscado la representación de marcas y compañías internacionales de punta, las alianzas estratégicas (HP, Sun , Cisco Systems, Microsoft, Hitachi, SAP) y las certificaciones (ISO/IEC 27001:2005, CMMI Nivel 3, ITIL, Six Sigma, SAP Hosting Partner) como un sinónimo de su calidad y su capacidad en outsorcing y en concretar soluciones corporativas multiplataforma.

Anexo 3

Aplicativo Validación Carga

El aplicativo de validación de datos para carga del LDAP es un proceso masivo o batch clásico.

Su objetivo es tomar un archivo de datos de usuarios y efectuar validaciones previas a la carga al LDAP. Estas validaciones son:

- contenido
  chequeo de dígito verificador de rut
- códigos
  chequeo contra tablas (rol del usuario, códigos de tablas)

Finalmente entregar estadísticas con los resultados. Cantidades de registros leídos, registros correctos, erróneos, totales de error por tipo de error y totales finales.

Descripción de Package

El aplicativo está desarrollado en Java, y se compone de los package:

- **control**
  Lógica de control central de la aplicación.

- **view.input**

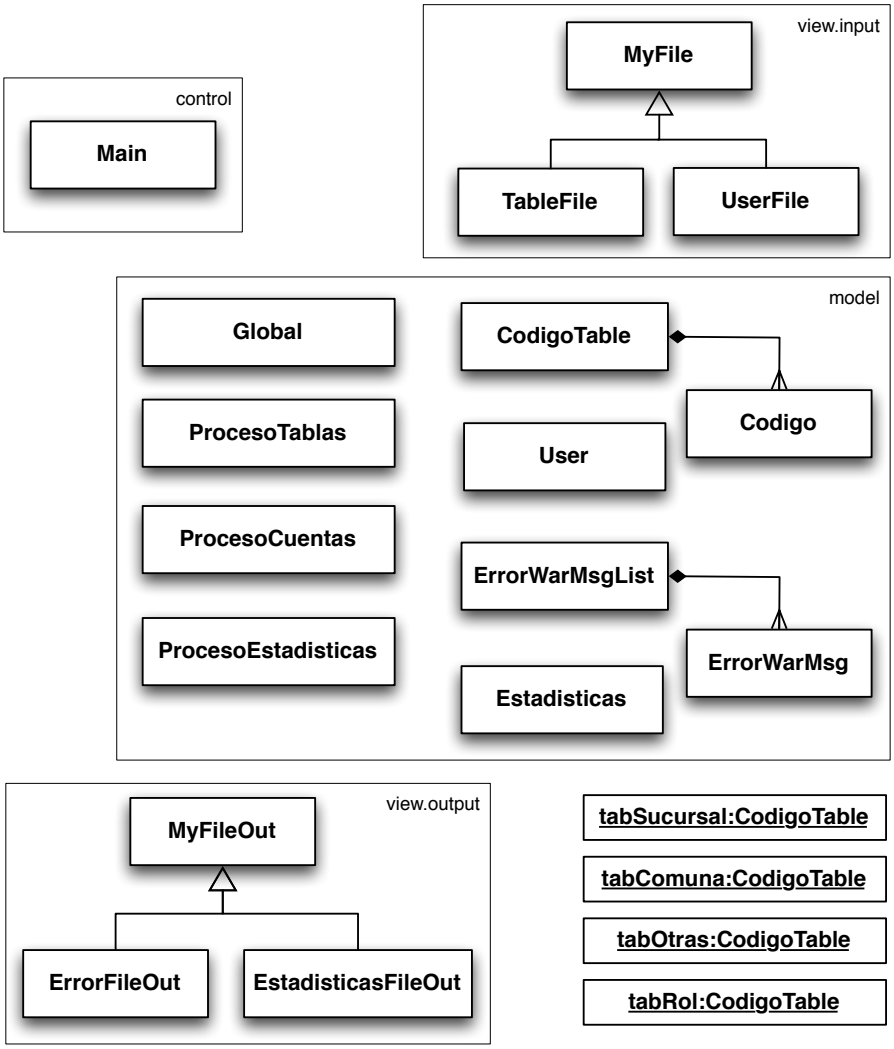
Clases para cargar tablas y archivo con datos de usuarios

- **model**
  Clases para validación de archivo con datos de usuarios, y recolección de estadísticas.

- **view.output**
  Clases para generación de los resultados: Archivos de errores y archivos de estadísticas de los resultados.

Los package y clases pueden verse en el siguiente diagrama:

Figura: Diagrama de clases de aplicativo de validación



En el diagrama se ilustran algunos objetos de tablas (de la clase CodigoTable) que son instanciados para las necesidades de validación.

Descripción de Clases

- package **control**

  - **Main**
    Clase que contiene al método *main* que controla todo el proceso. Utiliza una instancia de la clase Global, donde se almacena toda la información y paso de parámetros que son utilizados por las distintas clases.
    Llama a los métodos de las clases ProcesoTablas para efectuar la carga de tablas para validación, luego llama al método *validarCuentas* de la clase ProcesoCuentas y finalmente al método *printStat* de la clase ProcesoEstadisticas.

- package **view.input**

  - **MyFile**
    Es una clase abstracta que posee los métodos necesarios para leer archivos de extensión "csv" y efectúa el "parseo" de los headers para poder interpretar su contenido (esto es necesario porque en los archivos a veces cambia el orden de las columnas, y en ocasiones también el nombre: "sucursal" y "código suc.", por ejemplo).

  - **TableFile**
    Clase que hereda de MyFile. El objetivo final de la clase TableFile es entregar un objeto CodigoTable a partir de un archivo de entrada. Este objeto de la clase CodigoTable contendrá todos los códigos válidos para una tabla determinada.
    La clase TableFile lee registros de un archivo de entrada. Estos registros son pares de la forma (código tabla, glosa tabla). Para cada registro leído genera objetos de clase Codigo y cada uno de estos objetos es almacenado en una instancia de la clase CodigoTable.
    Finalmente entrega como resultado el objeto de la clase CodigoTable, que será luego usado para validación. En el diagrama se ilustran algunos objetos de tablas (de la clase CodigoTable) que son generados por la clase TableFile.

  - **UserFile**
    Clase que hereda de MyFile. Su objetivo es leer el archivo de cuentas de usuarios. Sus métodos serán llamados por la clase ProcesoCuentas, a la que irá entregando los registros de las cuentas de usuarios, en sucesivos objetos de la clase User.

- package **model**

  - **Global**
    Almacena parámetros y datos de uso común por todas las clases.

  - **ProcesoTablas**
    Clase que efectúa la carga de las tablas para validación, generando objetos del

tipo CodigoTable.

- **ProcesoCuentas**
  Clase que efectúa la validación de cuentas. Para cada registro genera un objeto ErrorWarMsgList. Este objeto está compuesto de objetos ErrorWarMsg que son los mensajes de tipo warning o error asociados al registro en validación.
  A medida que cada registro de cuentas es procesado, esta clase va grabando inmediatamente en el archivo de errores (de modo que no se produzca contención en memoria y se puedan procesar archivos grandes de usuarios). Además esta clase acumula las estadísticas de leídos, errores y ok en una instancia de la clase Estadisticas.

- **ProcesoEstadisticas**
  Clase que genera archivos con las estadísticas finales.

- **CodigoTable**
  Clase tipo lista, implementada a través de un HashMap, que contiene una lista de objetos de la clase Codigo, y un método *isIn (String codigo)* que informa si un código determinado está o no en la lista (para así efectuar la validación).

- **Codigo**
  Clase contenedora (POJO o Bean) que almacena los datos de un código-glosa de una tabla determinada.

- **User**
  Clase contenedora que almacena los datos de un registro de cliente determinado.

- **ErrorWarMsgList**
  Clase tipo lista, que contiene una lista de objetos de clase ErrorWarMsg, con los mensajes de error y warnings de validación, asociados a un registro de usuario determinado.

- **ErrorWarMsg**
  Clase contenedora que almacena la información de un error o warning específico, asociado a un registro de usuario determinado.

- **Estadisticas**
  Clase contenedora que almacena la información de las estadísticas y errores asociados al proceso.

▶ package **view.output**

- **MyFileOut**
  Es una clase abstracta que posee los métodos necesarios para grabar archivos planos.

- **ErrorFileOut**
  Clase que hereda de MyFileOut. Clase utilizada para grabar los mensajes de error/warning generados en el proceso de validación.

- **EstadisticasFileOut**
  Clase que hereda de MyFileOut. Clase utilizada para grabar las estadísticas del proceso de validación.

Anexo 4

Planes de Prueba (Extracto)

Los Planes de Prueba son extensos y con mucho detalle. Por esta razón sólo se incluyen, como ejemplo, algunos cuyos aspectos resultan más interesantes.

4.1 Pruebas Funcionales

Estas pruebas buscan verificar el comportamiento operacional del Sistema, y su cumplimiento con los requisitos funcionales.

Veamos un ejemplo de una prueba funcional para la Creación de Usuarios.

Concepto	Descripción
Categoría	Caso de prueba Funcional
Nombre	Creación de usuario
Objetivo	Certificar el correcto comportamiento del caso creación de usuarios y cuentas.
Alcances	Se probarán distintos escenarios, considerando creación de usuarios/cuentas con atribuciones básicas (con diferentes roles hacia el sistema SCI), así como usuarios con permisos de administración.
Estrategia	Efectuar la creación de usuarios/cuentas, verificar su correcto registro en el sistema, y verificar que la cuenta queda "operativa" en los ambientes de SCI y de autoatención. Con esto se logra certificar el ciclo de vida completo de la creación de usuario/cuenta.

Veamos la lista de condiciones que deben cumplirse para poder realizar la prueba:

Precondiciones
• Disponer de los manuales operacionales básicos de la aplicación • Plataforma preproducción habilitada • Nómina de usuarios a utilizar en las pruebas • Permisos a nómina usuarios aplicados en preproducción • Set de datos para casos de prueba • Datos requeridos para casos, cargados en ambiente preproducción • PCs habilitados con ambientes usuario • Conectividad Fonasa - Ambiente preproducción

El detalle del caso de prueba:

Caso prueba Creación de usuario **Paso de prueba**	**Resultado esperado**
elegir subcaso del set de datos de pruebas	
ir a url conexión TIM administrador	sistema presenta pantalla logon
ingresar user/password usuario TIM administrador del caso de prueba	sistema presenta menú opciones
elegir gestionar usuarios	sistema presenta screen CRUD usuarios
elegir crear usuarios	sistema abre screen creación usuarios y propone perfil base FonasaPerson
elegir continuar	sistema abre screen de información personal de usuario
ingresar datos del caso de prueba, elegir agregar	sistema abre screen con nómina de roles disponibles
seleccionar rol elegido para el caso de prueba y dar aceptar	sistema abre screen con linformación de empresa
ingresar aceptar	sistema abre screen con lista de sucursales (esto inicia búsqueda en LDAP)
elegir sucursal del usuario y dar agregar	sistema abre screen con información de contacto
ingresar mail elegido para el caso de prueba y dar continuar	sistema abre screen con linformación de generación de contraseña
elegir generar contraseña automáticamente y dar enviar seleccionar ver mi solicitud	sistema abre screen informando que solicitud de creación de usuario ha sido enviada al LDAP sistema abre screen con status de solicitudes. Luego de unos instantes debe mostar que la solicitud ha sido completada con éxito
seleccionar terminar sesión	sistema muestra pantalla de logon

Esta prueba continúa con :

- ▷ verificación del correo del usuario y del mail recibido
- ▷ ingreso al sistema SCI con el usuario creado y la contraseña indicada en el correo
- ▷ como es usuario nuevo la password se genera en estado de expirada y el sistema de seguridad (parte Autenticación) debe enrutar al usuario hacia el sistema de autoatención de Gestión de Usuarios
- ▷ en este punto efectuar el caso de prueba "Primera Conexión"
- ▷ ingresar nuevamente al sistema y efectuar el caso de prueba “Conexión Normal"
- ▷ generar evidencias y cerrar el caso de prueba

Y finalmente el cierre del caso de prueba:

Cierre
• Recopilación de evidencias • Entrega de evidencias • Registro de incidencias y su explicación • Observaciones y comentarios de la prueba • Establecimiento de acuerdos • Preparación documento de cierre del caso de prueba • Revisión de evidencias y ajustes al documento • Entrega final documento de cierre

4.2 PNF, pruebas de Convivencia

Las PNF de Instalación y Convivencia están orientadas a certificar que tanto la instalación del nuevo sistema, así como su uso no interfiere en la operación normal actual de Fonasa. Para estos efectos se define cuál es el ambiente operacional habitual, PCS e impresoras, software de S.O., browsers a utilizar, etc. Es decir, se debe recrear el ambiente normal de trabajo de Fonasa y verificar que todo opera correctamente.

Estas pruebas se dividen en tres grupos.

- Pruebas funcionalidad Fonasa *antes* de interactuar con las nuevas aplicaciones
- Pruebas de las nuevas aplicaciones (Gestión Usuarios y SCI)
- Pruebas funcionalidad Fonasa *después* de interactuar con las nuevas aplicaciones

Para determinar el conjunto de pruebas de los sistemas Fonasa, el usuario definió cuáles eran los sistemas y aplicaciones más frecuentes y/o más sensibles. Estas pruebas incluyeron la generación e impresión de informes y formularios, en los dispositivos de impresión más habituales.

Las pruebas fueron efectuadas en PCs e instalaciones de Fonasa, a objeto de verificar en terreno la correcta operación.

Esta tabla muestra los grupos y pruebas definidas. Esta tabla se complementa con columnas de detalle de prueba, responsable, tiempo, resultado esperado, status, observaciones, las que se omiten por claridad.

Pruebas de Convivencia	
Grupo Pruebas	Nómina Pruebas
Pruebas sistemas Fonasa, Antes	Sistemas de Fonasa Impresión de documentos
Pruebas nuevas aplicaciones SCI	Aplicaciones SCI
Pruebas nuevas aplicaciones Gestión de Usuarios	Administración, Creación de Usuario Autoatención - Cambio de password Verificación - Toma de evidencias
Pruebas sistemas Fonasa, Después	Sistemas de Fonasa Impresión de documentos

4.3 Pruebas Paso a Piloto

Esta tabla muestra las actividades requeridas para el paso a Piloto. Se muestra el grupo de actividad y el detalle.

Esta tabla se complementa con columnas de responsable, modalidad (procedimiento, manual o vía programa) tiempo, status, observaciones, las que se omiten por claridad.

Actividades y pruebas para paso a Piloto	
Actividades	Descripción
Anteriores al Piloto	Ajustar Matriz Permisos Carga Tabla Sucursales Carga Tabla EAPS en Sucursales Chequeo Detallado Calidad de Datos
Borrado completo de Usuarios de la BD	Borrado Verificación
Carga Usuarios	Prueba carga con Un Usuario Verificación Recepción de Correo Carga Usuarios Sucursal Carga Usuarios EAPS Carga Usuarios Nivel Central y GTI Generar dump Nómina Full Usuarios en la Base Verificación
Corrección Errores de Carga	Corregir Datos Agregar Usuarios
Check Point	Carga Usuarios está completa
Establecer Permisos Adicionales	Usuarios con rol ALL Usuarios TIM Solicitador Usuarios TIM Resolutor Usuarios Dueños de Proceso Verificación
Check Point	Carga Permisos está completa
Chequeos Disponibilidad	Ambiente Producción Autoatención Gestión Usuarios Ambiente Producción sistemas Plan Global Manual de Usuario Ejecutivo Sucursal Manual de Usuario EAPS
Pruebas y Verificaciones	Primera Conexión Recuperación Contraseña Permisos Usuario tipo Ejecutivo Sucursal Permisos Usuario tipo Jefe Sucursal Permisos Usuario tipo Ejecutivo EAPS
Prueba Mesa de Ayuda	Llamado x ayuda 1ra conexión Restablecimiento Contraseña
Check Point	Pruebas completas / Tarea finalizada