

TOTEM

Documento de Arquitectura del Software

<Versión 1.0>

Historial de Revisiones

Fecha	Versión	Descripción	Autor(es)
29-05-2015	1.0	Versión Inicial	Carlos Lozano, Santiago Bernal, Samuel Villegas

Contenido

1	Introducción.....	4
1.1	Propósito.....	4
1.2	Alcance.....	4
1.3	Personal involucrado	4
1.4	Definiciones, acrónimos y abreviaturas.....	4
1.5	Referencias.....	5
1.6	Resumen	5
2	Representación Arquitectónica	5
3	Metas de la Arquitectura y Restricciones	6
3.1	Metas	6
3.2	Restricciones	6
4	Vista Lógica	7
4.1	Vista de Paquetes.....	7
4.2	Diagrama de Clases	7
5	Vista de Casos De Uso	8
5.1	Casos de Uso	8
5.2	Diagramas de Casos de Uso	13
5.3	Diagramas de Secuencia	13
6	Vista de Despliegue.....	13
7	Vista de Implementación	14
8	Vista de Datos	14
9	Vista de Seguridad	14
10	Calidad.....	15
10.1	RNF	15
	Desempeño	15
	Seguridad	15
	Robustez.....	15
10.2	Estándares.....	16
11	Tamaño y Performance.....	16
12	Diagramas de Secuencia Inicio de Sesión	17
13	Diagrama de Secuencia de Recuperación de Clave	18
14	Diagrama de Secuencia de Cierre de Sesión	19
15	Diagrama ER.....	20
16	Diagrama de Clases Módulo 1.....	21

1 Introducción

El documento de arquitectura del software para el Sistema de gestión de documentos de análisis TOTEM, muestra el diseño arquitectónico del proyecto y el conjunto de partes en las que el sistema se dividió para asegurar un sistema más robusto, entendible y escalable, teniendo que cada división va destinado a un grupo de requerimientos funcionales que se deben de cumplir para la completitud del proyecto. El documento sirve de marco de referencia para guiar la construcción del software y estableciendo la estructura, funcionamiento e interacción entre las partes.

1.1 Propósito

El propósito de este documento es mostrar una visión global de la arquitectura usada para el desarrollo del software, así como las diferentes capas de: Datos, Lógica, Dominio, Excepciones, Interfaz y Pruebas Unitarias, así como también las herramientas seleccionadas y estándares.

1.2 Alcance

El alcance va vinculado al documento de especificación de requerimientos que describe los requerimientos del sistema y con ello se proporciona una vista arquitectónica que cumpla con todos los requerimientos del documento anterior.

1.3 Personal involucrado

Nombres	Santiago Bernal, Carlos Lozano, Samuel Villegas
Rol	Grupo 1.
Responsabilidades	Módulo de ingreso y recuperación de clave

1.4 Definiciones, acrónimos y abreviaturas

Nombre	Descripción
ERS	Especificación de Requerimientos de Software
DAS	Documento de Arquitectura de Software
Usuario	El usuario es empleado de Totem y tiene cierto rol en la empresa de desarrollo
Administrador	El administrador es un empleado de Totem y tiene un rol de administrador del sistema
Vista	Es una representación de un área de interés o perspectiva del sistema en alto nivel.

RNF	Requerimiento No Funcional
C#	<i>Lenguaje de programación</i>
ASP.NET	<i>Framework diseñado para el desarrollo web para crear páginas web dinámicas</i>

1.5 Referencias

Referencia	Título	Ruta	Autor
IEEE	Standard IEEE 830 - 1998	http://ayudantiasubb.files.wordpress.com/2008/06/ieee.pdf	IEEE
ERS	Especificación de Requerimientos del Software	https://github.com/jafuentest/totem/wiki	Grupo de Desarrollo TOTEM
Programación en capas	Arquitectura de N Capas	http://jtentor.com.ar/post/Arquitectura-de-N-Capas-y-N-Niveles.aspx	Julio Tentor
W3C	Estándares Webs	http://www.w3c.es/estandares/	W3C

1.6 Resumen

Este documento explica la arquitectura del software utilizada para el desarrollo del sistema de documentación de software TOTEM. Describiendo las vistas bajo las cuales el sistema se ha implementado, utilizando diagramas de casos de uso, diagramas de secuencia y cualquier artefacto que facilite dicha descripción. Explicando también como las vistas soportan los requerimientos funcionales y no funcionales que tienen alto impacto en las decisiones de la arquitectura del software.

2 Representación Arquitectónica

El sistema fue diseñado usando una arquitectura de “n” capas, en este caso fueron una capa de Datos, una capa de Lógica de Negocios, una capa de Interfaz, una capa de Dominio, una capa de Excepciones y una capa de Pruebas Unitarias, aunque esta última funciona como un proyecto aparte que prueba el código del sistema principal, no como una capa funcional del sistema. La capa de Datos es aquella que está encargada de buscar información en la base de datos, cada módulo definió su propia clase para implementar métodos que construyan el query para buscar la información necesaria de la base de datos, se estableció una clase que era la encargada de manejar la conexión y de ejecutar

los métodos, también se establecieron las clases Parámetro y Respuesta para el manejo de los parámetros de los query y respuesta para almacenar la información extraída de la base de datos. La capa de Lógica de Negocio, es la encargada de manejar la lógica de la aplicación, es la que hacia las llamadas a las clases en la capa de Datos y es la que usaba esa información para realizar las operaciones necesarias de la aplicación, métodos como los de Encriptar, Desencriptar y Enviar Correo se encuentran en esta capa. Al igual que la capa Datos, la capa Lógica estaba separada por módulos, de tal manera que cada módulo tiene al menos una clase en la capa lógica. La capa de la interfaz es la encarga de mostrar la información a través de una interfaz de usuario, en este caso en una página web, y es la encargada de comunicarse con la capa de Lógica. La capa de interfaz debía tener la menor cantidad de código posible, ya que cualquier operación que quiera hacer se debería hacer desde la capa lógica. Las capas de Dominio y de Excepciones, son conocidas por todo el proyecto, de manera que tanto la Interfaz, como la capa Lógica y la capa de Datos conocen de ambas, ya que en las 3 capas se implementa el manejo de objetos del Dominio y el manejo de excepciones. En la capa de Dominio se definen todas las clases, con sus atributos y métodos que cubren los requerimientos del proyecto. En la capa de Excepciones, también separada por modulo, se consiguen las Excepciones de Lógica del proyecto con los posibles errores que podrían ocurrir en tiempo de ejecución y así capturarlos y controlar lo que sucede luego con el sistema.

3 Metas de la Arquitectura y Restricciones

La arquitectura se escogió de esta manera para tener un mejor control y orden de código, además de poder distinguir y distribuir las responsabilidades del sistema, también al hacer el proyecto de esta manera garantizamos escalabilidad, usando un código que es fácil de entender y fácil de agregar nuevas funcionalidades y también se consigue un código que es fácil de mantener. Con la delimitación de las responsabilidades conseguimos un proyecto con alta cohesión y bajo acoplamiento.

3.1 Metas

La meta principal por la cual se estructuró el proyecto de esta manera es la de conseguir un diseño modular que dividiera las responsabilidades del proyecto en diferentes capas de tal manera que la capa de Interfaz solo conozca la capa Lógica, la capa Lógica conoce la capa de Datos solamente, y la capa de Datos se conecta con la base de datos.

3.2 Restricciones

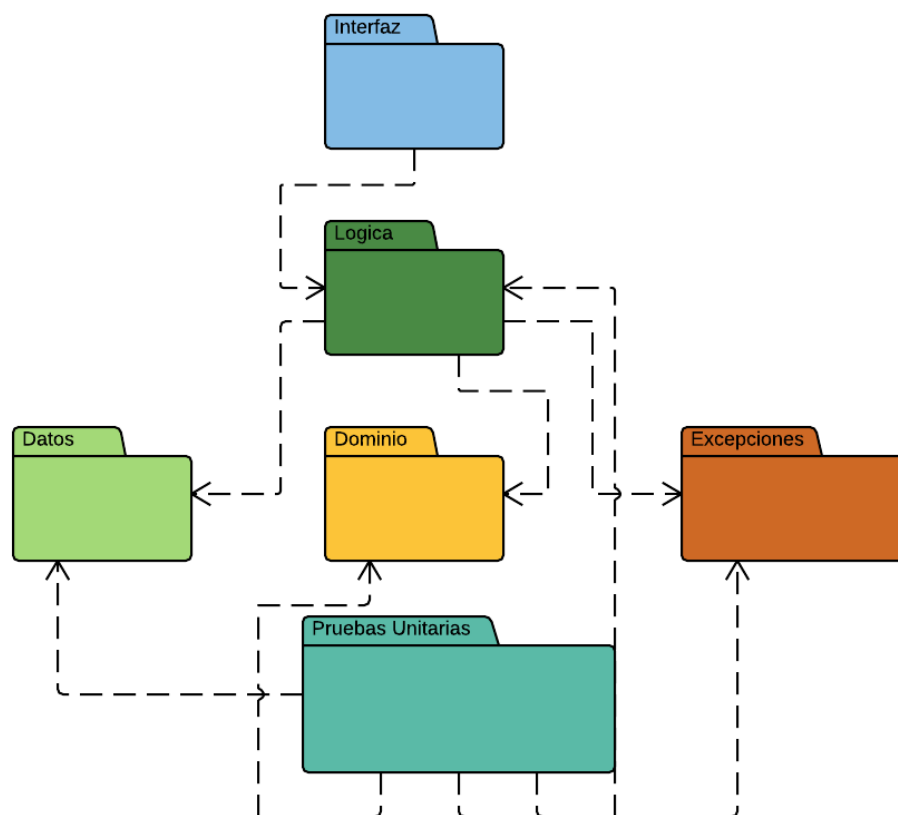
Las restricciones que este tipo de diseño implica es la inhabilidad de poder comunicar dos capas que no corresponden, por ejemplo: la capa lógica no debería conocer la capa de interfaz y la capa de interfaz no debería conocer la capa de datos ni viceversa, lo cual implica que en algunos casos se tenga que extender el tiempo necesario para realizar algunas funcionalidad, lo cual lo contrarresta el hecho de que la

arquitectura ofrece una capacidad para la reutilización de código, lo cual al final puede disminuir el tiempo necesario para realizar las funciones.

4 Vista Lógica

4.1 Vista de Paquetes

A continuación se encuentra un diagrama de los paquetes de alto nivel del proyecto TOTEM y su relación en el proyecto, como vemos los paquetes pueden darnos una vaga idea de las capas utilizadas en arquitectura de la solución.



4.2 Diagrama de Clases

El diagrama de clases asociado al Módulo 1 del proyecto TOTEM se encuentra en los anexos.

[Diagrama de Clases Módulo 1](#)

5 Vista de Casos De Uso

5.1 Casos de Uso

Caso de Uso: TOT_CU_1_1_1 Iniciar Sesión

Caso de Uso	TOT_CU_1_1_1 Iniciar Sesión
Precondición	El usuario o administrador debe estar registrado en el sistema TOTEM
Condición Final de Éxito	El usuario o administrador inició la sesión exitosamente.
Condición Final de Fallo	El usuario o administrador no pudo iniciar sesión. El usuario o administrador no está registrado en el sistema TOTEM.
Actor Primario	<ul style="list-style-type: none">• Usuario• Administrador
Disparador	El usuario o administrador ingresa al sistema TOTEM
Escenario Principal de Éxito	
<ol style="list-style-type: none">1. El usuario o administrador ingresa al sistema TOTEM.2. El sistema despliega la pantalla de Inicio de Sesión.3. El usuario o administrador ingresa los datos de Nombre de Usuario y Contraseña.4. El sistema verifica la existencia del usuario en el sistema y permite el ingreso del usuario al sistema.5. El sistema despliega la pantalla principal del sistema TOTEM. El Caso de Uso termina.	
Extensiones	
4-A. El usuario o administrador que se ingresó no está registrado en la base de datos.	

A1. El sistema notificará al usuario que debe ingresar un Nombre de Usuario que esté registrado en la base de datos.

A2. CU_7_1_1_1 Registro de Usuario

A3. El caso de Uso Termina.

4-B. La clave que el usuario o administrador introdujo no concuerda con la almacenada en la base de datos.

B1. Volver al paso 1.

Caso de Uso: TOT_CU_1_2_2 Recuperación de Clave

Caso de Uso	TOT_CU_1_2_2 Recuperación de Clave
Precondición	El usuario o administrador debe acceder a la pantalla de inicio
Condición Final de Éxito	El usuario o administrador realizó el cambio de clave exitosamente.
Condición Final de Fallo	El usuario o administrador no pudo realizar el cambio de clave.
Actor Primario	<ul style="list-style-type: none">• Usuario• Administrador
Disparador	El usuario o administrador selecciona la opción de ¿Olvidó su contraseña?
Escenario Principal de Éxito	
<ol style="list-style-type: none">1. El usuario o administrador selecciona la opción de ¿Olvidó su contraseña?.2. El sistema despliega la pantalla Ingresar Correo.	

3. El usuario o administrador ingresa su correo con el que se registró al sistema y presiona el botón de Continuar.
4. El sistema valida el correo ingresado y envía un mensaje a dicho correo.
5. El usuario o administrador selecciona el enlace enviado a su correo.
6. El sistema despliega la pantalla con la pregunta de seguridad.
7. El usuario o administrador ingresa la respuesta de la pregunta formulada y presiona el botón de Continuar.
8. El sistema valida la respuesta ingresada y despliega la pantalla de Cambio de Clave.
9. El usuario o administrador ingresa la Nueva Clave y su confirmación y presiona el botón de Continuar.
10. El sistema valida que ambas claves ingresadas en el punto anterior sean iguales.
11. El sistema registra en la Base de Datos la nueva clave asociada al usuario. El Caso de Uso termina.

Extensiones

3-A. El usuario o administrador presiona el botón de Cancelar

A1. El sistema lo redirecciona a la página de Inicio de Sesión.

A2. CU_1_1_1 Iniciar Sesión. El Caso de Uso termina.

4-A. El correo ingresado no está registrado.

A1. Se despliega el siguiente mensaje de error: "El correo ingresado es inválido o no está registrado".

A2. Volver al paso 3.

7-A. El usuario o administrador presiona el botón de Cancelar

A1. El sistema lo redirecciona a la página de Inicio de Sesión.

A2. CU_1_1_1 Iniciar Sesión. El Caso de Uso termina.

8-A. La respuesta es inválida

A1. Se despliega el siguiente mensaje de error: "La respuesta es inválida, consulte a su administrador".

A2. Volver al paso 7.

9-A El usuario o administrador presiona el botón de Cancelar

A1. El sistema lo redirecciona a la página de Inicio de Sesión.

A2. CU_1_1_1 Iniciar Sesión. El Caso de Uso termina.

10-A. Las claves no son iguales

A1. Se despliega el siguiente mensaje de error: "Las claves no coinciden."

A2. Volver al paso 9.

11-A. El sistema no pudo registrar el cambio de clave en la Base de Datos.

A1. Se despliega el siguiente mensaje de error: "Error en la conexión. Intente más tarde."

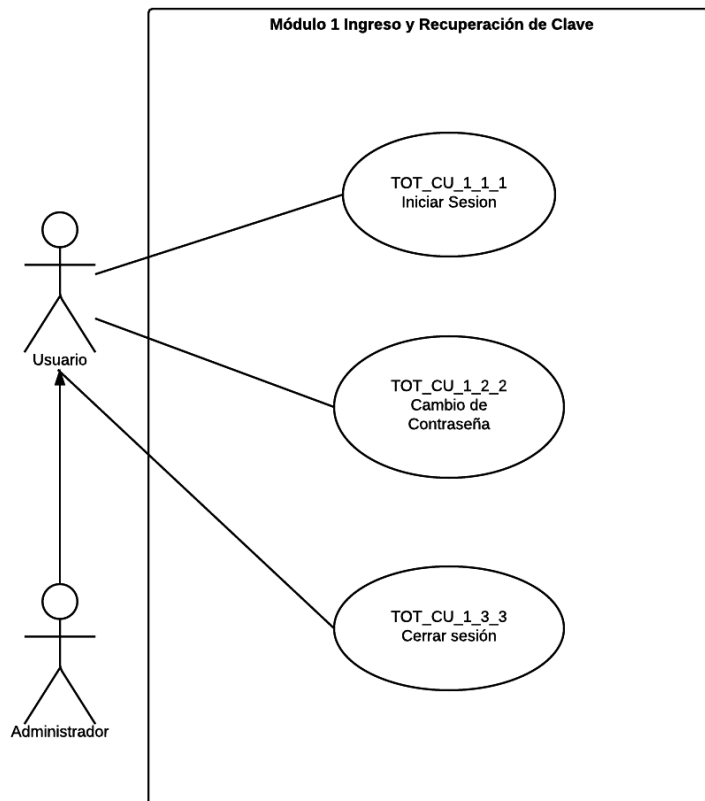
A2. CU_1_1_1 Iniciar Sesión

Caso de Uso: TOT_CU_1_3_3 Cerrar Sesión

Caso de Uso	TOT_CU_1_3_3 Cerrar Sesión
Precondición	El usuario o administrador debió haber iniciado la sesión.
Condición Final de Éxito	El usuario o administrador cerró la sesión exitosamente.

Condición Final de Fallo	El usuario o administrador no pudo cerrar la sesión del sistema.
Actor Primario	<ul style="list-style-type: none"> • Usuario • Administrador
Disparador	Seleccionar el botón Cerrar sesión
Escenario Principal de Éxito	
<ol style="list-style-type: none"> 1. El usuario o administrador selecciona el botón de cerrar sesión. 2. El sistema despliega la pantalla de Inicio de Sesión del sistema. El Caso de Uso termina 	
Extensiones	
<p>2-A. El sistema no pudo cerrar la sesión</p> <p>A1. Mostrar mensaje de error: "No se pudo cerrar la sesión. Intente nuevamente."</p> <p>A2. Volver al paso 1.</p>	

5.2 Diagramas de Casos de Uso



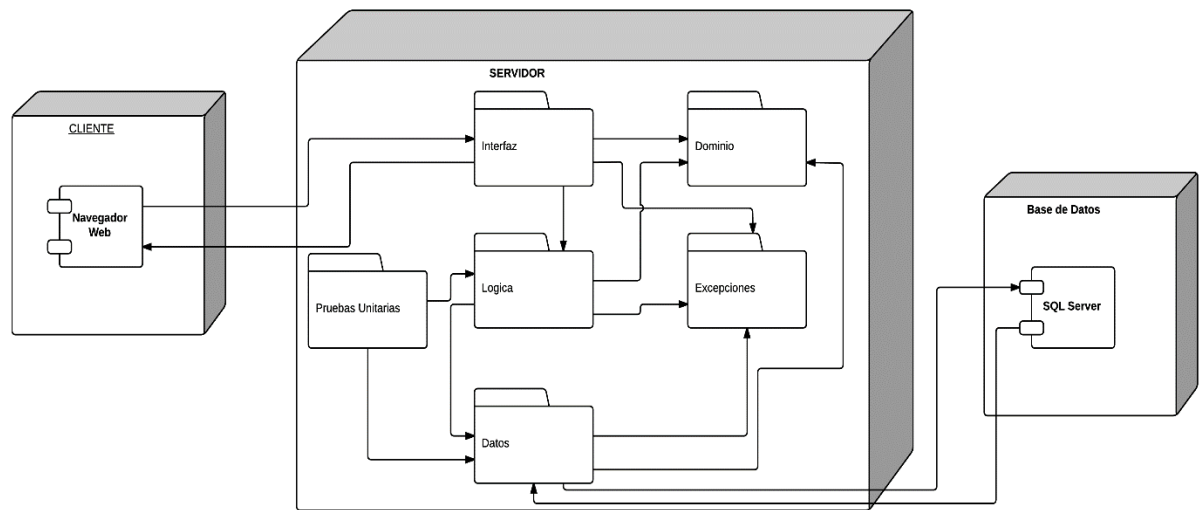
5.3 Diagramas de Secuencia

- [Diagrama de Secuencia de Inicio de Sesión](#)
- [Diagrama de Secuencia de Recuperación de Clave](#)
- [Diagrama de Secuencia de Cierre de Sesión](#)

6 Vista de Despliegue

El sistema está hecho de tal forma de que las capas de Interfaz, Lógica y Datos puedan funcionar desde lugares diferentes con tal de que exista un web service que permita la conexión entre Interfaz y Lógica y entre Lógica y Datos, para lo cual se tendrían que hacer pocas modificaciones de código, lo ideal siendo que todas las capas se encuentren en un mismo servidor. Cuando el servidor acceda a través del navegador web, este se debe conectar con la interfaz, y la capa de Datos debe tener acceso a una base de datos, la cual debería estar en un servidor aparte.

7 Vista de Implementación



8 Vista de Datos

- [Diagrama E-R de la base de datos](#)

9 Vista de Seguridad

Como medidas de seguridad se implementó el uso de expresiones regulares para el filtrado de datos de ingreso, específicamente en los campos que requieren el ingreso de correo. Otra medida que se implementó, es establecer un máximo de caracteres posibles en los campos de inputs, se establecieron métodos de encriptamiento con Rijndael para el paso de datos por el URL, encriptado con hash para el almacenamiento de contraseñas en la base de datos, y por último los queries de base de datos fueron hechos con stored procedures parametrizados. Con todo lo anterior podemos garantizar que el sistema desarrollado tiene protección contra ataques cross-scripting, con el filtrado de datos, también se asegura el ataque con SQL-Injection al hacer queries parametrizado y ataques a fuerza bruta con el encriptado de los datos.

10 Calidad

10.1 RNF

Desempeño

La aplicación garantizará que los usuarios tendrán una eficiencia de un 98% al usar la aplicación.

Seguridad

La arquitectura de desarrollo planteada para el sistema de n capas brinda más seguridad.

El sistema tiene implementadas medidas de seguridad que lo protegen contra ataques comunes al mismo.

Robustez

A través del manejo de excepciones y errores el sistema se comporta de manera robusta, lo que convierte al sistema en un sistema tolerante a fallas.

Mantenibilidad

Gracias a la arquitectura de desarrollo planteada para el sistema el mantenimiento perfectivo y adaptativo, se puede realizar de una manera muy sencilla.

Portabilidad

El sistema TOTEM corre basado en Web lo que hace que pueda ser utilizado desde cualquier dispositivo que tenga conexión a la red y corra un explorador como Chrome, Firefox o Safari convirtiéndola en una aplicación multiplataforma.

Disponibilidad

La disponibilidad del sistema estará sujeta a la disponibilidad del servidor de producción en donde se aloje.

10.2 Estándares

Tipo	Estandar	Ejemplo
Clase	PascalCase	Usuario
Interfaz	PascalCase, inicio con I	ICliente
Excepción	PascalCase, termina con Exception	UsuarioException
Métodos	PascalCase	ToString()
Parámetro	Camel	CalcularValor(Valor valorConseguido)
Atributos	Camel	nombreUsuario
Variables	Camel	Int contadorCicloUsuario
Constantes	Mayusculas	final String archivo XML_PATH
Enumeration	PascalCase	DiasDeSemana

Acrónimos:

Dos y tres caracteres mayúscula:

-BDRegistro -DAOUsuario

Más de tres minúscula:

-htmlWriter

11 Tamaño y Performance

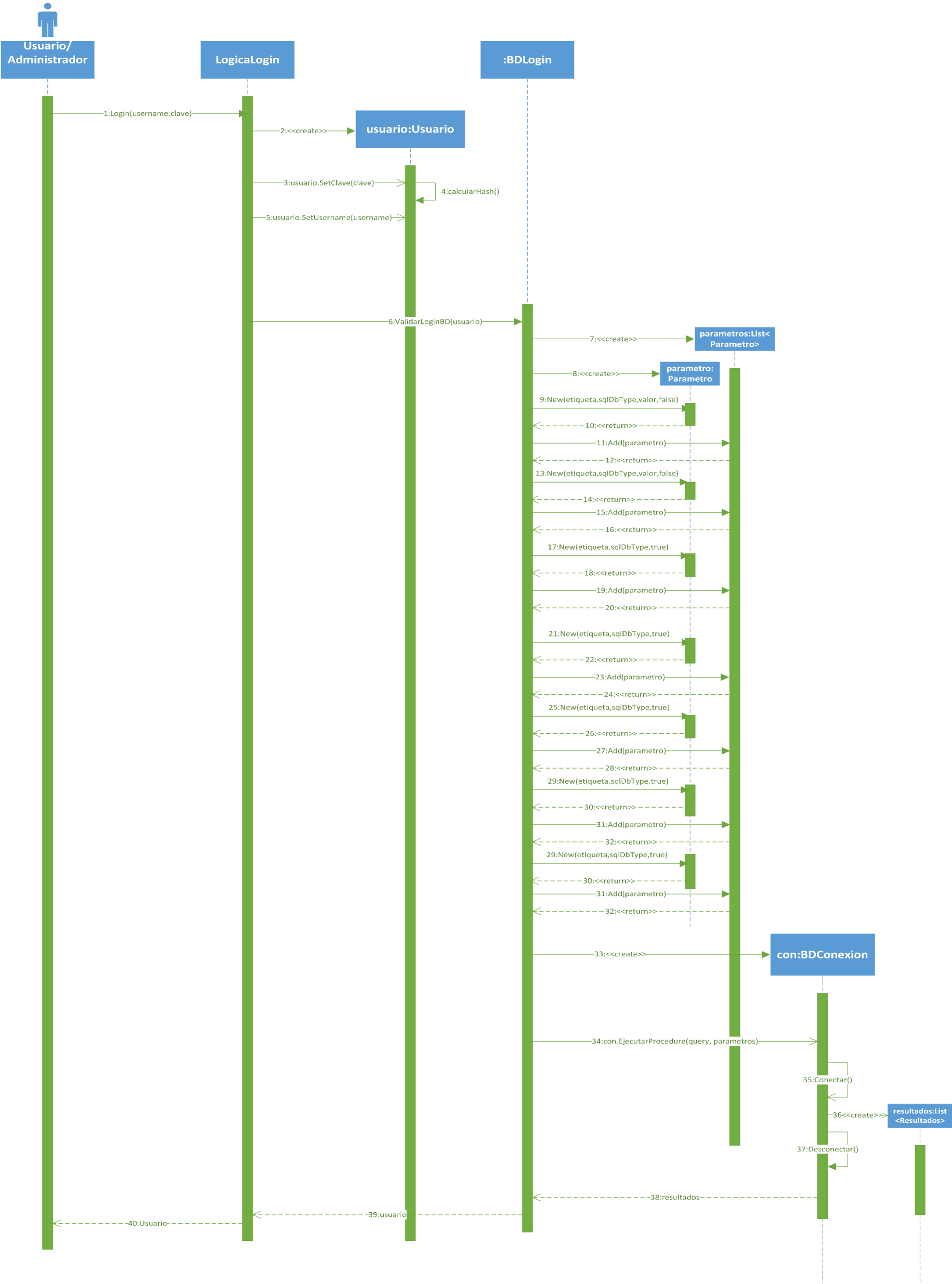
El sistema TOTEM está previsto para ser utilizado por pocos usuarios, aunque la arquitectura diseñada permite el crecimiento de la aplicación a grandes escalas. Todo dependerá del equipo donde se aloje la base de datos y la aplicación.

Del lado del cliente como es una aplicación web el crecimiento puede ser masivo y el manejo de la concurrencia en la base de datos está basado en el manejador SQL Server, que permite a los usuarios trabajar de manera continua.

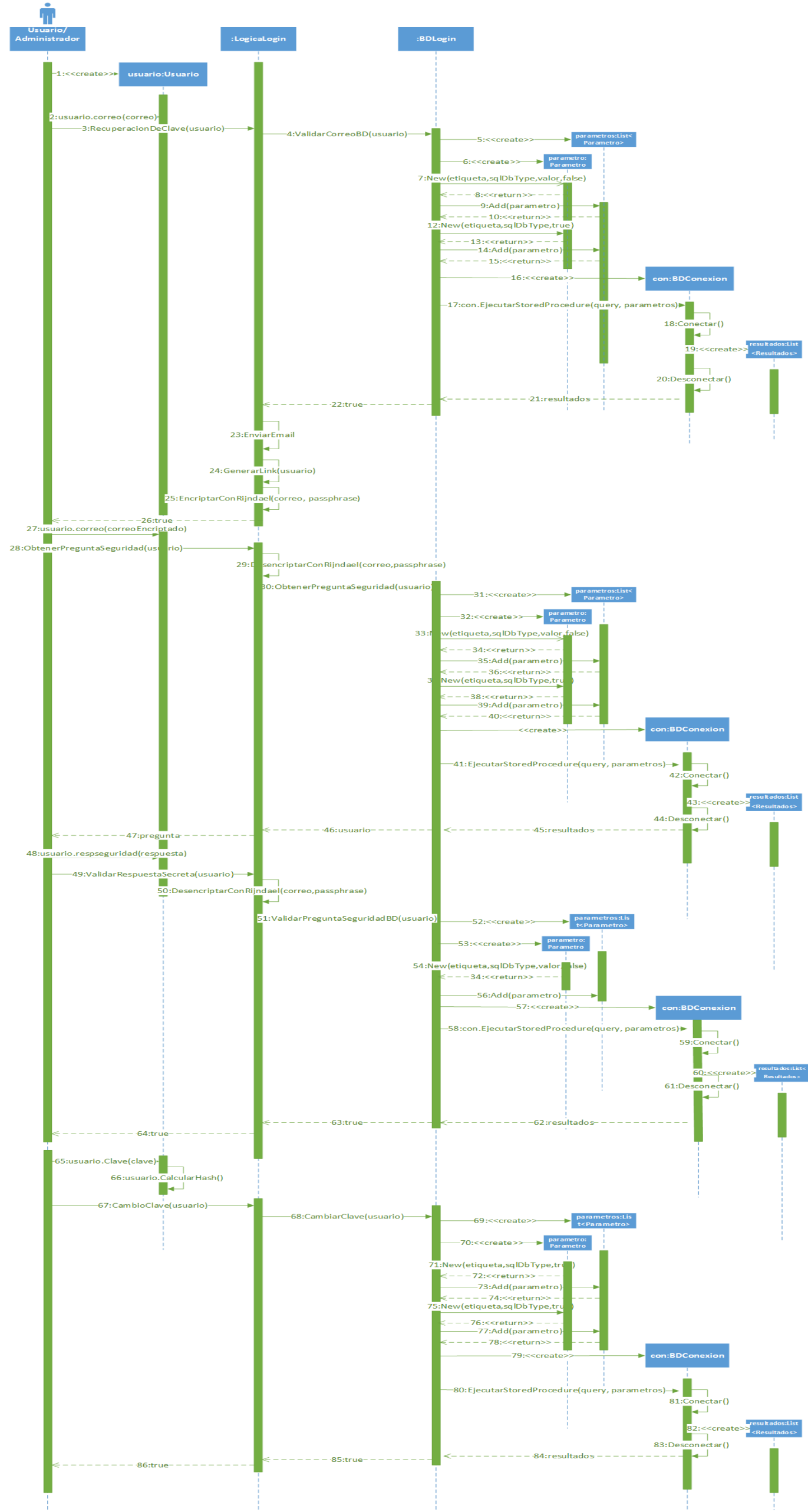
Se requiere que el equipo donde será instalada la aplicación tenga al menos 40mb de espacio en disco duro y 4gb de RAM para correr los servicios necesarios.

Anexos

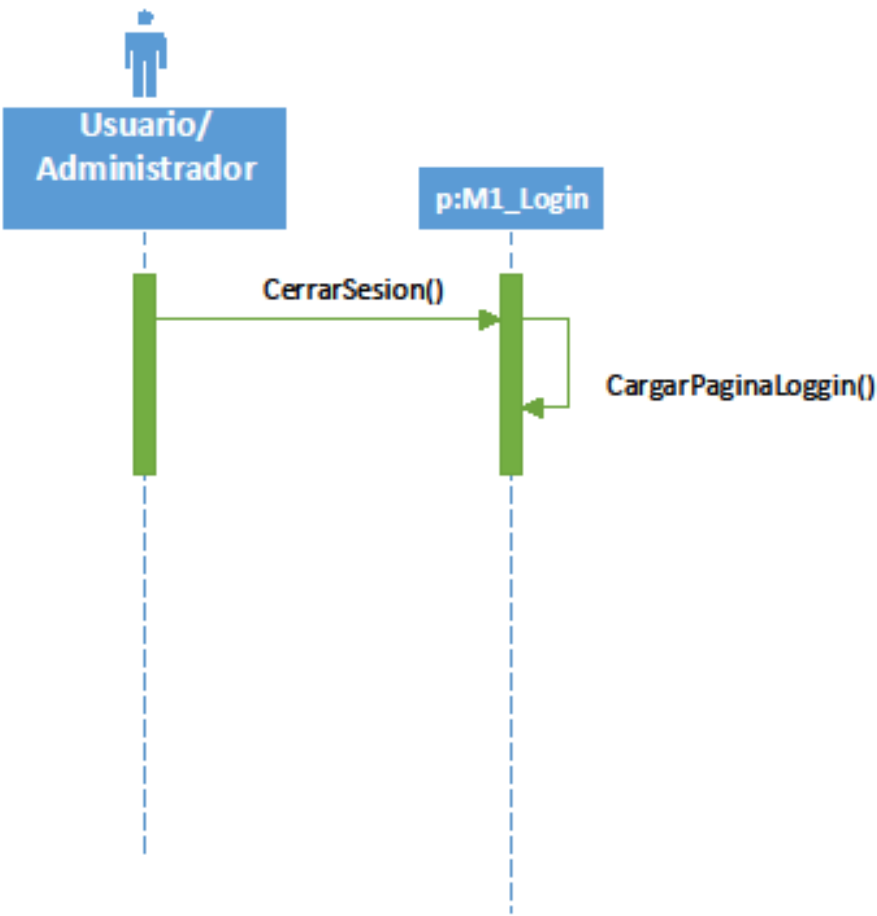
12Diagramas de Secuencia Inicio de Sesión



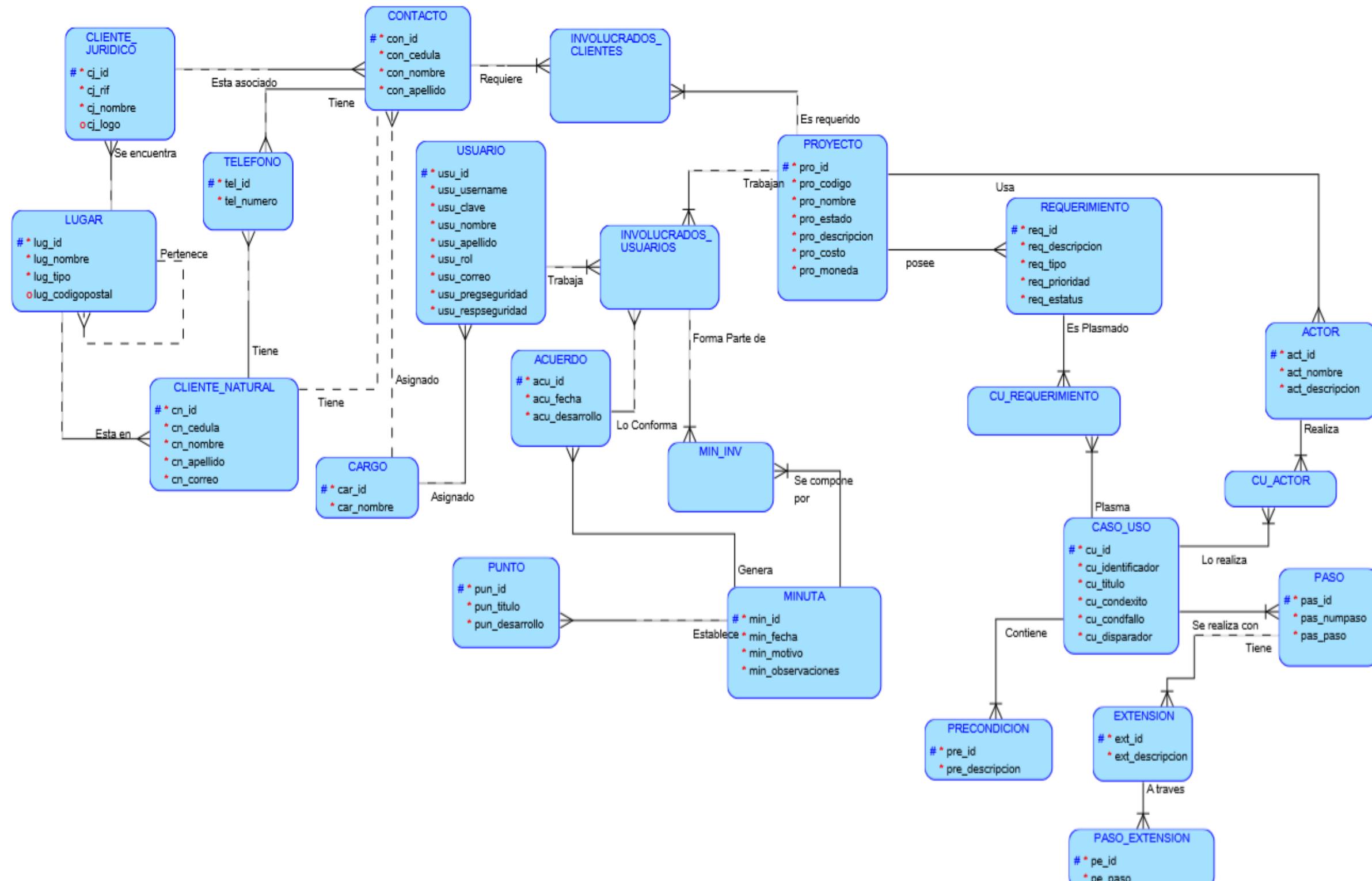
13 Diagrama de Secuencia de Recuperación de Clave



14 Diagrama de Secuencia de Cierre de Sesión



15 Diagrama ER



16 Diagrama de Clases Módulo 1

