

**Rediseño y Modernización de Página Web**

**Versión 1.0**

**Fabian Albornoz**

**Alonso Arellano**

**Paula Arancibia**

**Identificación de Documento**

| **Identificación** | MDLimitada |
| --- | --- |
| **Proyecto** | Rediseño y Modernización de Página Web |
| **Versión** | 1.0 |

| **Documento mantenido por** | Alonso Arellano |
| --- | --- |
| **Fecha de última revisión** | 03/10/2024 |
| **Fecha de próxima revisión** | 06/10/2024 |

| **Documento aprobado por** | Fabian Albornoz |
| --- | --- |
| **Fecha de última aprobación** | 06/10/2024 |

**Historia de cambios**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 03/10/2024 | 1.0 | Se inicia el desarrollo del DAS | Alonso Arellano |
| 06/10/2024 | 1.1 | Se termina el desarrollo de DAS | Alonso Arellano |
|  |  |  |  |
|  |  |  |  |

**Tabla de Contenidos**

[**Introducción 2**](#_heading=h.g1ee2pzhungc)

[**Alcance 3**](#_heading=h.3znysh7)

[**Referencias 3**](#_heading=h.2et92p0)

[**Arquitectura de Software 3**](#_heading=h.tyjcwt)

[Vista de Casos de Uso y Escenarios de Calidad: 3](#_heading=h.qam6iucapllk)

[Vista de Metas y Restricciones: 4](#_heading=h.56qdqptmdgfa)

[Vista Lógica: 4](#_heading=h.zbzkzeyor34z)

[Vista de Procesos: 5](#_heading=h.42k76cqn2cbz)

[Vista de Implementación: 5](#_heading=h.mylp39dbb8fc)

[**Objetivos y Restricciones de la Arquitectura 6**](#_heading=h.7f6dud7jeyn9)

[Objetivos de la arquitectura 6](#_heading=h.1t3h5sf)

[Restricciones de la Arquitectura 6](#_heading=h.4d34og8)

[**Arquitectura de Procesos 7**](#_heading=h.26in1rg)

[**Arquitectura de desarrollo 8**](#_heading=h.lnxbz9)

[**Arquitectura física 9**](#_heading=h.35nkun2)

[**Escenarios 10**](#_heading=h.44sinio)

[Modelo de Casos de Uso 10](#_heading=h.2jxsxqh)

[Especificación de Casos de Uso Relevantes 13](#_heading=h.z337ya)

[Especificación de los Escenarios de Calidad Relevantes 14](#_heading=h.3j2qqm3)

[**Tamaño y desempeño 16**](#_heading=h.1y810tw)

# Introducción

METAL DISEÑO CHILE SPA, fundada el 8 de junio de 2009, surge como una extensión de la reconocida MAESTRANZA Y CARPINTERÍA METÁLICA RAS, consolidando una tradición que abarca desde 1990 hasta el 2009.

Como empresa familiar, ha heredado no solo la experiencia sino también el sólido sistema de trabajo en el ámbito de la carpintería metálica. Ubicados en Santa Inés 01559, Quinta Normal, Santiago, se especializan en la reparación y fabricación de piezas metálicas para ascensores, montacargas y escalas mecánicas, así como en la fabricación e instalación de estructuras de acero inoxidable.

En conjunto con un intermediario de la empresa se desarrollará un proyecto el cual consiste en modernizar y rediseñar, en su mayoría, la app web que este negocio ofrece, con la finalidad de optimizar tiempo y procesos, para crear nuevas oportunidades de negocio dentro de la empresa.

# Alcance

El alcance del proyecto se define y dirige, con la creación, rediseño y modernización completa de la página web de Metal Diseño, que genere opciones y funciones que faciliten y optimicen el proceso tanto de brindar información, como de gestión de pedidos y registro de clientes, como también gestión de colaboradores y sus cotizaciones, todo esto pre establecido y definido en métodos y lógica de procesos. Se espera que, con la realización y desarrollo del sistema, se logre establecer un escenario de funcionamiento idóneo o con los requisitos mínimos para el sistema, donde pueda cumplir y realizar con las tareas esperadas de este mismo, junto a el despliegue correcto de todas sus funcionalidades según lo estimado por el Metal Diseño.

# Referencias

A continuación se listan las referencias a documentos relacionados:

* Product Backlog
* Plan de Trabajo
* Historias de User

# Arquitectura de Software

El patrón arquitectónico designado para ese proyecto es una arquitectura MVC (Modelo Vista Controlador), pues es el que más se adapta al desarrollo de una solución web.

La arquitectura del sistema de la app web está representada siguiendo el enfoque del framework 4+1. Las vistas incluidas en esta versión del documento son:

### **Vista de Casos de Uso y Escenarios de Calidad:**

Describe los casos de uso más significativos, presenta los actores y una descripción de sus casos de uso asociados. De igual forma describe los escenarios de calidad más relevantes para la arquitectura.

### **Vista de Metas y Restricciones:**

Describe restricciones tecnológicas, normativas, estándares, etc., los cuales influyen sobre las decisiones arquitectónicas, del producto y del proceso de desarrollo.

**Metas:**

* Los usuarios podrán hacer uso de la app web según su clasificación en el sistema.
* Reducir tiempo en el desarrollo de procesos relacionados a la gestión de clientes y empleados, y al brinde de información de la empresa.
* Permitirá una visualización rápida de distintos casos y métricas del proceso.

**Restricciones:**

* El sistema de empleados, posteo y gestión de clientes solo funcionara en la vista de usuarios con sesiones válidas.
* Debe existir una separación de funcionalidades entre los usuarios de secretaría y gerencia.
* El sistema funcionará con Oracle 18 XE como motor de base de datos.

### 

### **Vista Lógica:**

Describe la arquitectura del sistema presentando varios niveles de refinamiento. Indica los módulos lógicos principales, sus responsabilidades y dependencias. Usa el view type Módulos para representar la estructura lógica y el view type Componentes y Conectores para representar el comportamiento.



### **Vista de Procesos:**

Describe los procesos involucrados para darle sentido a la ejecución del sistema, así como sus relaciones de comunicación y sincronización.

### **Vista de Implementación:**

Describe los componentes de deployment construidos y sus dependencias.

# Objetivos y Restricciones de la Arquitectura

A continuación se revisan las metas y restricciones de la arquitectura.

## Objetivos de la arquitectura

De acuerdo a las reuniones y al análisis de los requerimientos, se listan los principales conductores iniciales de la arquitectura los cuales corresponden a las metas arquitectónicas iniciales:

* **Seguridad**: El sistema usará variadas maneras para resguardar la seguridad y continuo funcionamiento del programa, estas siendo por ejemplo,módulos de seguridad integrados en el workframe, protección de hosting y la aplicación de sistemas de seguridad del sistema.
* **Mantenibilidad:** El programa será capaz de ser fácilmente modificado y usado para otros usos, debido a que este seguirá un estricto protocolo para la uniformidad del código escrito, lo cual ayudaría masivamente para que el programa pueda ser usado en ambientes con otras funcionalidades.
* **Adecuación funcional**: El sistema tiene que seguir los requerimientos no funcionales implicados anteriormente, para así estar al pie de la letra con lo que el cliente necesita.
* **Usabilidad**: El sistema tiene que ser sencillo de usar, ya que tiene que poder ser dominado de manera rápida e intuitiva por los usuarios de todas las edades.

## Restricciones de la Arquitectura

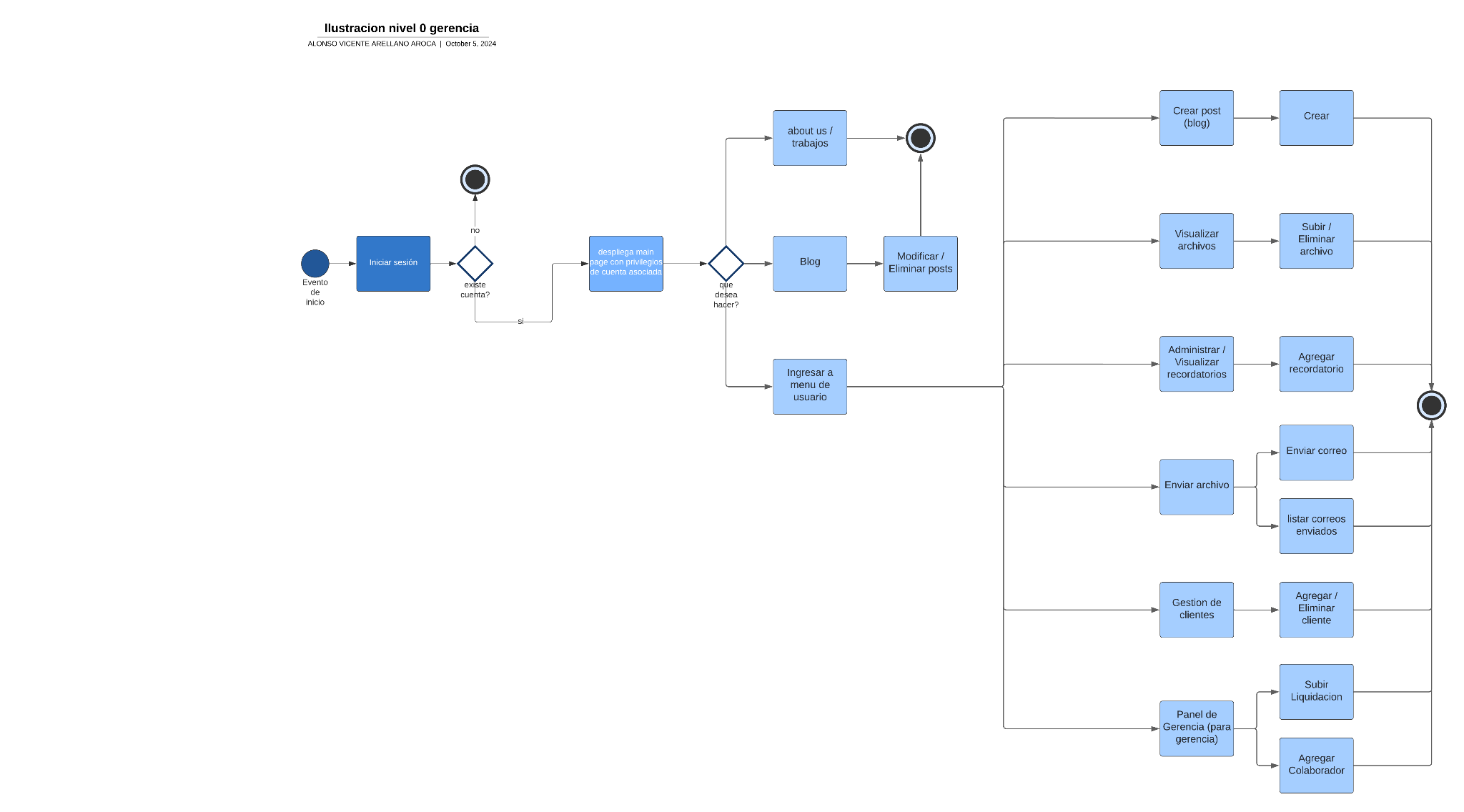
Existen restricciones que han sido levantadas con los stakeholders, las cuales se presentan a continuación:

* **Tiempo de construcción**: se cuenta con un plazo estrecho de tiempo para su construcción, 15 semanas según la planificación.
* **Infraestructura**: se cuenta con servidores de aplicación replicados por parte de un host web y con balanceadores de carga pre integrados, futuramente con una base de datos robusta.

# Arquitectura de Procesos

A continuación se muestra una vista de procesos

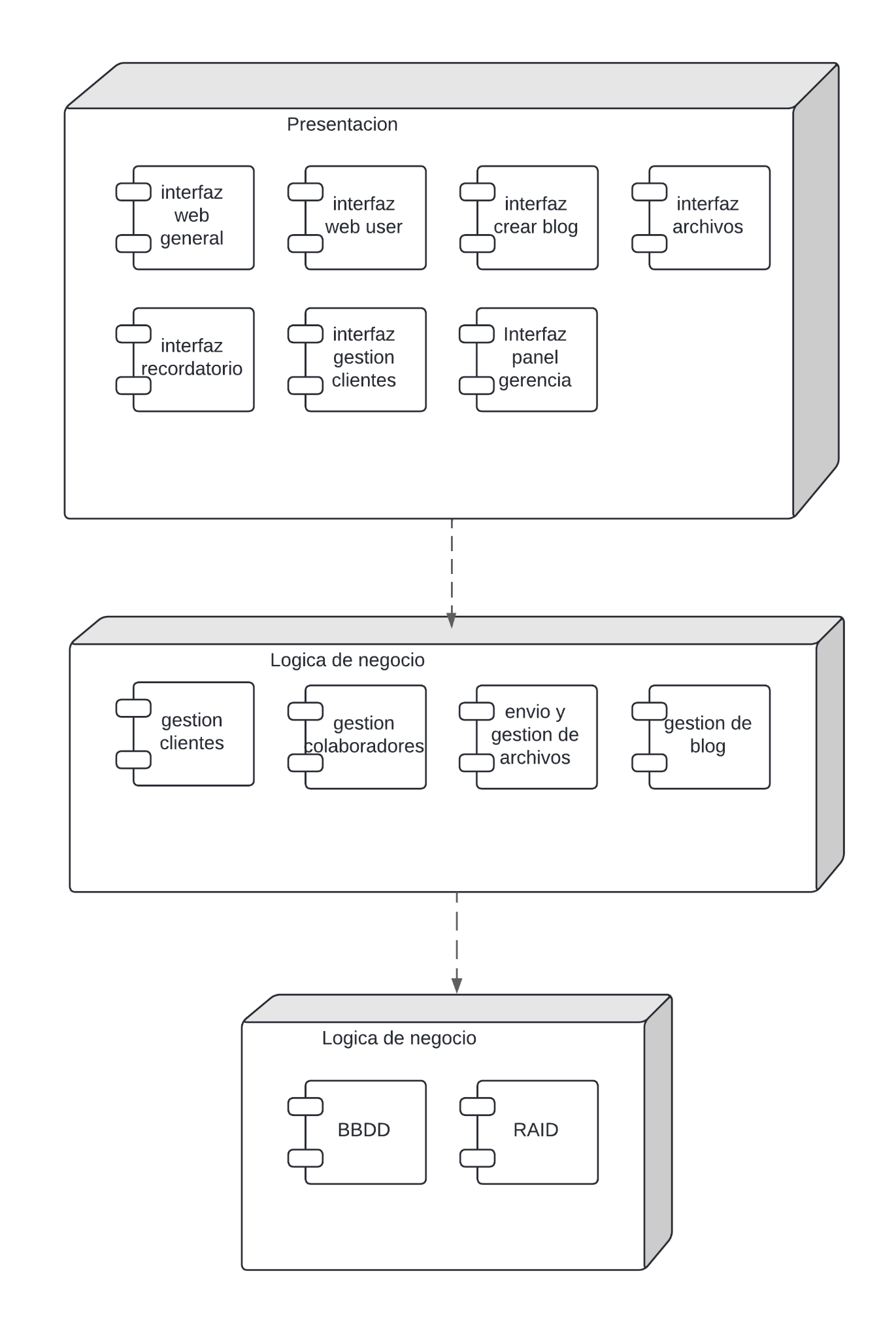
**Diagrama de actividad**

****

# Arquitectura de desarrollo

En esta vista se aprecia que existirán dos módulos principales que contendrán distintas funcionalidades de la aplicación. A continuación se describen:

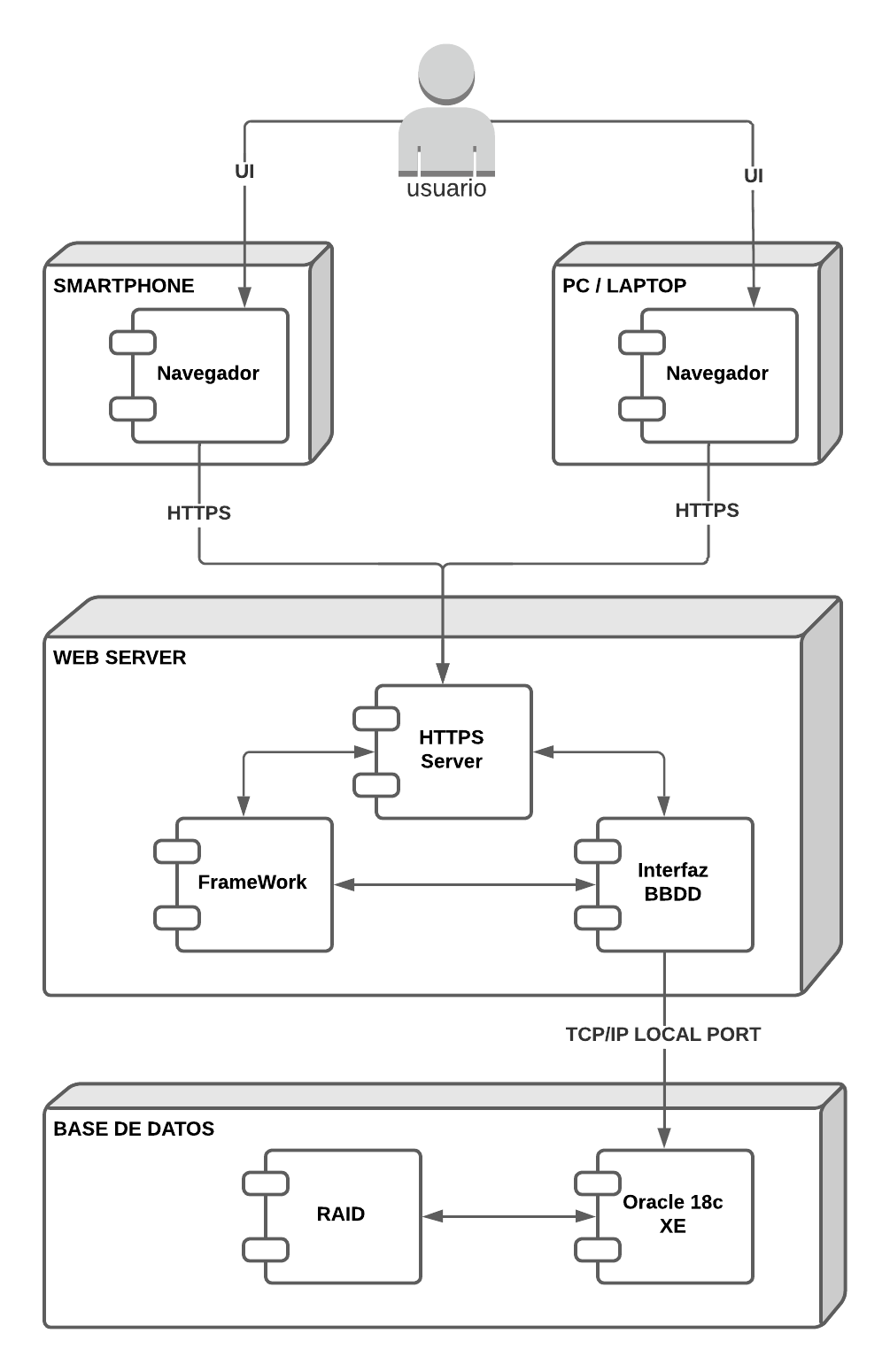
**Vista de Implementación (Diagramas de componentes o paquetes)**



# Arquitectura física

En esta vista se despliegan los nodos que participan con el sistema. Los nodos principales son los nodos Servidor de Integración. Características a continuación:

**Diagrama de Despliegue**

****

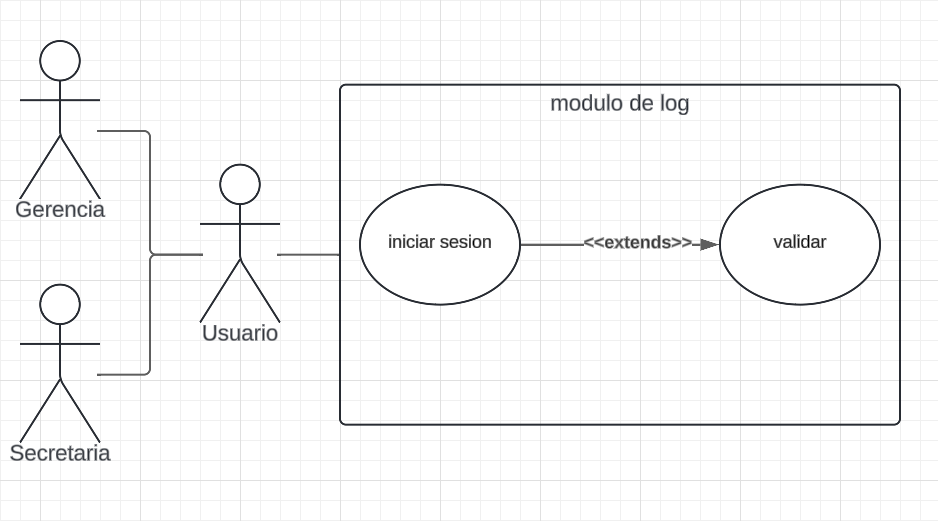
# Escenarios

Esta sección describe en detalle el conjunto de escenarios funcionales y no funcionales que obtuvieron la mayor prioridad en el análisis. Para esto se presenta y describe el diagrama de casos de uso y los casos de uso prioritarios, así como los escenarios en que uno o más atributos de calidad se ven involucrados de manera significativa.

## Modelo de Casos de Uso

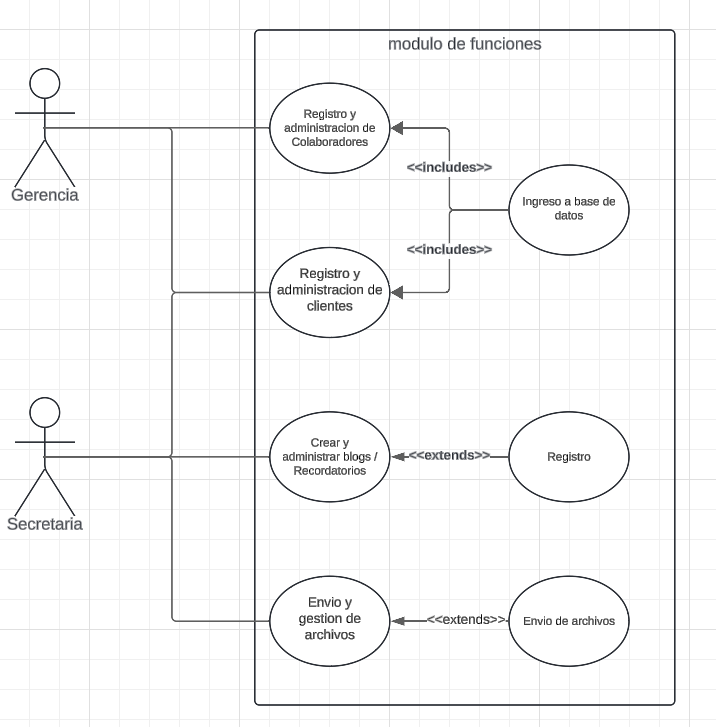
**Caso de uso 1: Login**

CU-001-001



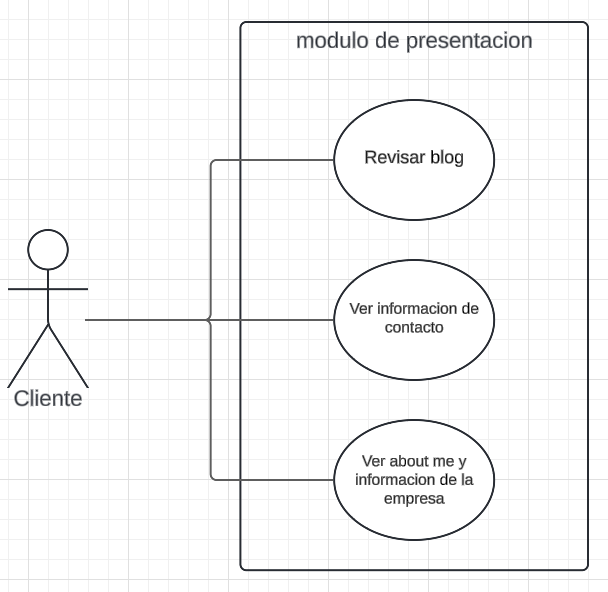
**Caso de uso 2: Funciones para Gerencia y Secretaria**

CU-002-002



**Caso de uso 3: Funciones para Usuario**

CU-003-003



## Especificación de Casos de Uso Relevantes

Los casos de uso considerados los más relevantes para el desarrollo de la arquitectura fueron determinados. Los criterios usados para dicha determinación fueron:

* Su implementación implica varios nodos de la vista de despliegue.
* Su implementación es de alto riesgo.
* Incluye muchos conceptos y relaciones del dominio.
* Incluye posibles escenarios críticos de calidad.

A continuación se listan los casos de uso relevantes, los cuales pueden ser encontrados con su especificación detallada en el documento “Casos de Uso”.

| **Código** | **Nombre** | **Actores** | **Prioridad** |
| --- | --- | --- | --- |
| CU-001-001 | Login de Sistema | Cliente, Gerente, Secretaria y sistema | Alta |
| CU-002-002 | Funciones de secretaría y gerencia | Gerente, Secretaria y Sistema | Alta |
| CU-002-003 | Funciones de Usuario | Cliente, Sistema | Alta |

## Especificación de los Escenarios de Calidad Relevantes

Después de un análisis en conjunto con los stakeholders, los escenarios de calidad se expresan a continuación:

**Escenario de calidad N°1**

**Atributo de Calidad Asociado (Característica):** Seguridad  
**Subcaracterística:** Control de Acceso  
**Descripción:** Solo los usuarios con credenciales válidas deben poder acceder al sistema de gestión de empleados y clientes.  
**Fuente del Estímulo:** Usuario  
**Estímulo:** Intento de acceder al sistema sin una sesión válida.  
**Artefacto:** Sistema  
**Ambiente:** Producción  
**Respuesta:** El sistema deniega el acceso y redirige al usuario a la página de inicio de sesión.  
**Medida de Respuesta:** El 100% de los intentos no autorizados son denegados.

**Escenario de calidad N°2**

**Atributo de Calidad Asociado (Característica):** Mantenibilidad  
**Subcaracterística:** Facilidad de Modificación  
**Descripción:** El sistema debe permitir la actualización de la lógica de gestión de pedidos sin afectar otras funcionalidades.  
**Fuente del Estímulo:** Desarrollador  
**Estímulo:** Solicitud de cambio en la lógica de gestión de pedidos.  
**Artefacto:** Código del Sistema  
**Ambiente:** Desarrollo  
**Respuesta:** Se modifica la lógica de gestión de presupuesto sin alterar las funcionalidades relacionadas con la gestión de empleados o clientes.  
**Medida de Respuesta:** El tiempo de implementación del cambio no debe exceder 10 horas, y el sistema debe pasar todas las pruebas unitarias.

**Escenario de calidad N°3**

**Atributo de Calidad Asociado (Característica):** Usabilidad  
**Subcaracterística:** Facilidad de Aprendizaje  
**Descripción:** Los usuarios deben poder navegar el sistema y realizar una cotización sin necesidad de capacitación avanzada.  
**Fuente del Estímulo:** Usuario  
**Estímulo:** Un nuevo usuario intenta realizar una cotización.  
**Artefacto:** Interfaz de Usuario  
**Ambiente:** Producción  
**Respuesta:** El sistema guía al usuario a través de la interfaz con mensajes claros e intuitivos para completar la cotización.  
**Medida de Respuesta:** El 90% de los usuarios nuevos deben poder gestionar clientes y realizar tareas asociadas a su rol en menos de 5 minutos.

# Tamaño y desempeño

Las principales decisiones arquitectónicas se tomaron teniendo en cuenta la restricción de tiempo para la construcción del proyecto. Dado que el desarrollo debe completarse en un plazo ajustado y sin margen de retrasos, se optó por una arquitectura bien conocida y de bajo riesgo en cuanto a su implementación.

El escenario de calidad relacionado con la mantenibilidad nos lleva a diseñar con un enfoque en la separación de responsabilidades de los componentes. También se utiliza el patrón "provider", lo que permite que el sistema delegue sus solicitudes de información a sistemas externos a través de componentes de software desacoplados, favoreciendo así su capacidad de extensión futura.

Además, la arquitectura se dividió en módulos con el objetivo de separar las responsabilidades, lo que permite trabajar en paralelo en diferentes componentes. Esto también facilita la realización de pruebas unitarias en cada módulo, asegurando que cada parte tenga una baja probabilidad de fallos.