

---

# SOFTWARE DESIGN DESCRIPTION

---

SOLVO - Sistema de  
información colaborativo  
para los conductores de carga  
terrestre

**SOLVO**

---

EQUIPO SOLVO  
Camilo Andrés Oviedo Lizarazo  
Juan Sebastián Sánchez López  
Sergio Enrique Plazas Merino

---

## HISTORIAL DE CAMBIOS

<b>Fecha de Modificación</b>	<b>Versión</b>	<b>Cambio(s) Realizado(s)</b>	<b>Responsable</b>
29/01/2018	0.1	Desarrollo de plantilla documento y creación del numeral 1 de la plantilla.	Juan Sebastián Sánchez L.
30/01/2018	0.2	Creación del numeral 2 de la plantilla.	Juan Sebastián Sánchez L.
03/02/2018	0.3	Modificación de Numeral 2.	Sergio Plazas Juan Sebastián Sánchez L.
10/02/2018	0.4	Creación del Numeral 3 y Modificación del mismo.	Camilo Oviedo
18/02/2018	0.5	Desarrollo de numeral 4, creación de diagrama de contexto y corrección de estilo de documento.	Camilo Oviedo
19/02/2018	0.6	Creación de Diagramas	Juan Sebastián Sánchez L.
24/02/2018	0.7	Corrección de la parte de Casos de Uso	Juan Sebastián Sánchez L.
25/02/2018	0.8	Desarrollo de los casos de uso arquitecturalmente significantes y complementación de las abreviaciones y términos	Juan Sebastián Sánchez L.

27/02/2018	0.9	Modificación del Documento con formato establecido	Camilo Oviedo
10/05/2018	1.0	Corrección errores de forma en el documento y corrección citas bibliográficas	Sergio Plazas
18/05/2018	1.1	Corrección de estilo, imágenes y finalización de Documento SDD	Camilo Oviedo
21/05/2018	1.2	Elaboración numeral 6. Interfaces de la aplicación	Sergio Plazas
21/05/2018	1.3	Finalización de documento y corrección de estilo	Camilo Oviedo
16/06/18	1.4	Adición diagramas al documento y corrección diagramas	Sergio Plazas
21/06/2018	1.5	Corrección de diagramas	Juan Sebastián Sánchez L.

## TABLA DE CONTENIDO

<b>HISTORIAL DE CAMBIOS</b> .....	1
<b>LISTA DE ILUSTRACIONES</b> .....	4
<b>LISTA DE TABLAS</b> .....	6
<b>1. Introducción</b> .....	7
<b>1.1. Descripción del sistema</b> .....	7
<b>1.1.1. Propósito</b> .....	7
<b>1.1.2. Abreviaciones y términos</b> .....	8
<b>2. Metodología de diseño</b> .....	8
<b>2.1. Análisis y Trazabilidad de Requerimientos</b> .....	8
<b>2.2. Pruebas</b> .....	14
<b>3. Arquitectura</b> .....	15
<b>3.1. Diagrama de Casos de Uso</b> .....	15
<b>3.2. Vista Lógica</b> .....	17
<b>3.3. Vista Física</b> .....	21
<b>3.4. Vista de Procesos</b> .....	23
<b>3.5. Vista de Despliegue</b> .....	32
<b>3.6. Vista de Datos</b> .....	34
<b>4. Diseño de Alto Nivel</b> .....	35
<b>4.1. Estructura del Sistema</b> .....	35
<b>4.2. Comportamiento e Interacción</b> .....	36
<b>5. Diseño de Bajo Nivel</b> .....	46
<b>5.1. Diagrama de Clases</b> .....	46
<b>6. Interfaces de la aplicación</b> .....	47
<b>REFERENCIAS</b> .....	52
<b>ANEXOS</b> .....	53

## LISTA DE ILUSTRACIONES

<i>Ilustración 1. Modelo de Vistas de Arquitectura 4+1 (Kruchten, 1995).....</i>	<i>7</i>
<i>Ilustración 2. Diagrama de casos de Uso SOLVO. Elaborado con Enterprise Architect....</i>	<i>16</i>
<i>Ilustración 3. Visión General SOLVO. Elaboración propia. ....</i>	<i>17</i>
<i>Ilustración 4. Vista lógica de SOLVO. Elaborado en Enterprise Architect .....</i>	<i>18</i>
<i>Ilustración 5. Vista Física del prototipo SOLVO. Elaboración Propia. ....</i>	<i>22</i>
<i>Ilustración 6. Vista de procesos de función "Cerrar Sesión". Elaborado en Bizagi. ....</i>	<i>23</i>
<i>Ilustración 7. Vista de procesos de función "Comentar Establecimiento de Negocio". Elaborado en Bizagi.....</i>	<i>24</i>
<i>Ilustración 8. Vista de procesos de función "Consultar Cuenta". Elaborado en Bizagi. ...</i>	<i>24</i>
<i>Ilustración 9. Vista de procesos de función "Eliminar Cuenta". Elaborado en Bizagi. ....</i>	<i>25</i>
<i>Ilustración 10. Vista de procesos de función "Editar Datos". Elaborado en Bizagi. ....</i>	<i>25</i>
<i>Ilustración 11. Vista de procesos de función "Mostrar Publicidad No Invasiva de Negocio". Elaborado en Bizagi.....</i>	<i>26</i>
<i>Ilustración 12. Vista de procesos de función "Iniciar Sesión ". Elaborado en Bizagi.....</i>	<i>27</i>
<i>Ilustración 13. Vista de procesos de función "Exhibir Publicidad No Invasiva de Negocio". Elaborado en Bizagi.....</i>	<i>27</i>
<i>Ilustración 14. Vista de procesos de función "Extraer Información de Componentes de Usuarios". Elaborado en Bizagi.....</i>	<i>28</i>
<i>Ilustración 15. Vista de procesos de función "Iniciar Sesión Administrador". Elaborado en Bizagi. ....</i>	<i>28</i>
<i>Ilustración 16. Vista de procesos de función "Cerrar Sesión Administrador". Elaborado en Bizagi. ....</i>	<i>29</i>
<i>Ilustración 17. Vista de procesos de función "Modificar Contenido". Elaborado en Bizagi. .....</i>	<i>29</i>
<i>Ilustración 18. Vista de procesos de función "Registrar Cuenta". Elaborado en Bizagi....</i>	<i>29</i>
<i>Ilustración 19. Vista de procesos de función "Trazar Ruta". Elaborado en Bizagi.....</i>	<i>30</i>
<i>Ilustración 20. Vista de procesos de función "Borrar Cuenta Administrador". Elaborado en Bizagi. ....</i>	<i>30</i>
<i>Ilustración 21. Vista de procesos de función "Calificar Establecimiento de Negocio". Elaborado en Bizagi.....</i>	<i>30</i>

<i>Ilustración 22. Vista de procesos de función "Consultar Servicio". Elaborado en Bizagi.</i>	<i>31</i>
<i>Ilustración 23. Vista de despliegue de SOLVO. Elaboración en Enterprise Architect.</i>	<i>33</i>
<i>Ilustración 24. Vista de Datos del Sistema SOLVO.</i>	<i>35</i>
<i>Ilustración 25. Diagrama de Contexto SOLVO. Elaborado con Enterprise Architect</i>	<i>35</i>
<i>Ilustración 26. Diagrama de Secuencia-Caso de uso Buscar Servicio Peaje. Elaborado con Enterprise Architect.</i>	<i>37</i>
<i>Ilustración 27. Diagrama de Secuencia-Caso de uso Buscar Estaciones de Servicio. Elaborado con Enterprise Architect.</i>	<i>38</i>
<i>Ilustración 28. Diagrama de Secuencia-Caso de uso Buscar Alojamiento. Elaborado con Enterprise Architect.</i>	<i>39</i>
<i>Ilustración 29. Diagrama de Secuencia-Caso de uso Buscar Restaurante. Elaborado con Enterprise Architect.</i>	<i>40</i>
<i>Ilustración 30. Diagrama de Secuencia-Caso de uso Buscar Parqueadero. Elaborado con Enterprise Architect.</i>	<i>41</i>
<i>Ilustración 31. Diagrama de Secuencia-Caso de uso Buscar Taller. Elaborado con Enterprise Architect.</i>	<i>42</i>
<i>Ilustración 32. Diagrama de Secuencia-Caso de uso Trazar Ruta. Elaborado con Enterprise Architect.</i>	<i>43</i>
<i>Ilustración 33. Diagrama de Secuencia-Caso de uso Comentar Servicio. Elaborado con Enterprise Architect.</i>	<i>44</i>
<i>Ilustración 34. Diagrama de Secuencia-Caso de uso Calificar Servicio. Elaborado con Enterprise Architect.</i>	<i>45</i>
<i>Ilustración 35. Diagrama de Clases del Sistema SOLVO. Elaborado con Enterprise Architect.</i>	<i>46</i>
<i>Ilustración 36. Parte "Pantallas de inicio" del árbol de seguimiento.</i>	<i>48</i>
<i>Ilustración 37. Parte "Pantalla principal" del árbol de seguimiento.</i>	<i>49</i>
<i>Ilustración 38. Parte "Pantalla de servicios" del árbol de seguimiento.</i>	<i>50</i>

## **LISTA DE TABLAS**

<i>Tabla 1. Abreviaciones y términos del documento SDD.</i> .....	8
<i>Tabla 2. Especificación de Requerimientos Funcionales.</i> .....	11
<i>Tabla 3. Especificación de Requerimientos No Funcionales.</i> .....	13

# SOLVO

# 1. Introducción

## 1.1. Descripción del sistema

### 1.1.1. Propósito

Este documento es una representación del diseño del software que se destinará para el registro de la documentación requerida del diseño y base de construcción de la aplicación SOLVO. Este documento abarca la cantidad necesaria de modelos, diagramas, y diseños destacados para la construcción del software, evidenciando cómo estará conformado el sistema al finalizar su construcción.

Además, se proporciona una visión general de arquitectura integral, presentando una serie de diferentes puntos de vista de arquitectura para interpretar diferentes aspectos del sistema. Con el fin de describir el software con la mayor precisión posible, la estructura de este documento se basa en la idea de "4 + 1" modelo de la arquitectura (Kruchten, 1995).

Siendo un modelo diseñado para “describir la arquitectura de sistemas software, basados en el uso de múltiples vistas concurrentes”. Las cuatro vistas del modelo seleccionado son: vista lógica, vista de desarrollo, vista de proceso y vista física. Además, una selección de casos de uso se utilizó para ilustrar la arquitectura sirviendo como una vista más (“4 + 1”).

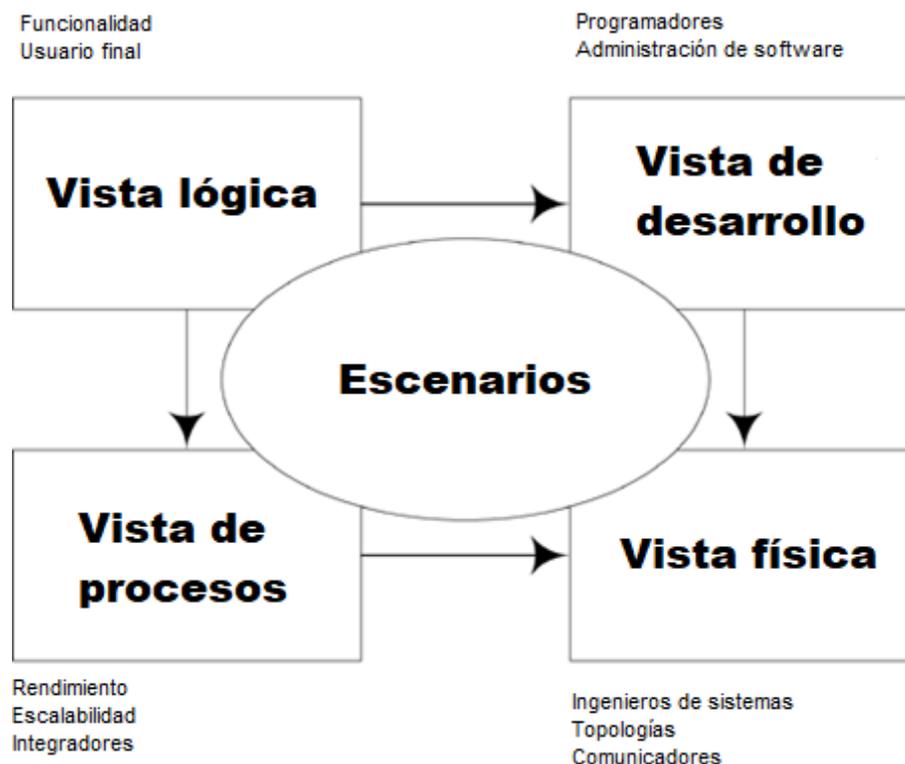


Ilustración 1. Modelo de Vistas de Arquitectura 4+1 (Kruchten, 1995).

### 1.1.2. Abreviaciones y términos

Abreviación	Término
SRS	Especificación de Requerimientos de Software
UML	Lenguaje Unificado de Modelado
APK	Tipo de aplicación informática para SO Android
SDD	Documento de Descripción del Diseño

*Tabla 1. Abreviaciones y términos del documento SDD.*

## 2. Metodología de diseño

### 2.1. Análisis y Trazabilidad de Requerimientos

Los requerimientos seleccionados se generan gracias a la percepción de la problemática descrita en el documento de Propuesta de Trabajo de Grado, a partir de lo anterior se generan diferentes formas para ofrecer servicios requeridos a los conductores de carga terrestre brindándoles confort en sus actividades diarias. Estableciendo parámetros que generan confort a los conductores de transporte de carga terrestre, se logra determinar seis servicios clave establecidos en los se empezaron a tomar decisiones enfocadas a desarrollar un prototipo valido para que se pueda cumplir el objetivo final del presente trabajo de grado.

Posteriormente, se generaron requerimientos tanto funcionales como no funcionales requeridos para dar solución al problema planteado.

Los seis servicios establecidos de manera prioritaria son: alojamiento, parqueadero, restaurante, estación de gasolina, talleres y lavaderos; los cuales serán parte del prototipo final, idealizando el diseño de la plataforma y generando ideas de qué debería contener cada servicio para el cumplimiento de su objetivo (botones, formularios, datos requeridos, lógica de negocio, etc.).

Al completar las etapas anteriormente mencionadas se comienza la presente etapa de construcción de la arquitectura que se explica más adelante en el presente documento; una vez culminado el diseño del sistema SOLVO se dará inicio a la etapa de construcción del prototipo.

### 2.1.1. Requerimientos Funcionales

A continuación, en la siguiente tabla de requerimientos funcionales, se presenta el identificador, tipo y descripción de los requerimientos funcionales; realizando su respectiva priorización más adelante.

<b>REQUERIMIENTOS FUNCIONALES</b>		
<b>ID</b>	<b>Tipo</b>	<b>Descripción</b>
FU1	Usuario	El sistema debe permitir al usuario registrarse
FU2	Usuario	El sistema debe permitir al usuario iniciar sesión
FU3	Usuario	El sistema debe permitir al usuario editar su cuenta
FU4	Usuario	El sistema debe permitir al usuario eliminar su cuenta
FU5	Usuario	El sistema debe permitir al usuario consultar alojamientos
FU6	Usuario	El sistema debe permitir al usuario consultar establecimientos de servicio
FU7	Usuario	El sistema debe permitir al usuario consultar peajes
FU8	Usuario	El sistema debe permitir al usuario consultar puntos de alimentación
FU9	Usuario	El sistema debe permitir al usuario consultar puntos de talleres
FU10	Usuario	El sistema debe permitir al usuario consultar parqueaderos
FU11	Usuario	El sistema debe permitir al usuario realizar comentarios a servicios ejecutados
FU12	Usuario	El sistema debe permitir al usuario realizar calificaciones a servicios ejecutados
FA1	Administrador	El sistema debe permitir al administrador iniciar sesión
FA2	Administrador	El sistema debe otorgar al Administrador un nivel de acceso mayor para ver un listado de las cuentas que están registradas en el sistema
FA3	Administrador	El sistema debe permitir al administrador monitorear contenido en cuanto a las cuentas de los usuarios registrados,

		y el contenido de los establecimientos
FA4	Administrador	El sistema debe permitir al administrador eliminar cuentas de los usuarios registrados
FS1	Sistema	El sistema debe contar con al menos un Administrador con capacidad de administración y control de cuentas de usuario
FS2	Sistema	El sistema debe soportar uno a muchos usuarios
FS3	Sistema	El sistema debe desplegar información sobre alojamientos registrados
FS4	Sistema	El sistema debe desplegar información sobre establecimientos de servicio registrados
FS5	Sistema	El sistema debe desplegar información sobre peajes registrados
FS6	Sistema	El sistema debe desplegar información sobre puntos de alimentación registrados
FS7	Sistema	El sistema debe desplegar información sobre puntos de talleres registrados
FS8	Sistema	El sistema debe desplegar información sobre parqueaderos registrados
FS9	Sistema	El sistema debe no permitir a los usuarios eliminar cuentas de otros usuarios
FS10	Sistema	El sistema debe no permitir al usuario registrarse con un correo electrónico no válido
FS11	Sistema	El sistema debe no permitir a los usuarios ingresar a una cuenta que ya se eliminó
FS12	Sistema	El sistema debe no permitir registrarse si no se cuenta con una conexión a internet
FS13	Sistema	El sistema debe poder ser accedido desde cualquier celular con dispositivo Android versión 4.2 Jelly Bean en adelante
FS14	Sistema	El sistema debe validar la sesión al ser iniciada

FS15	Sistema	El sistema no debe permitir registrar una cuenta con un correo electrónico usado en una cuenta ya existente
FS16	Sistema	El sistema debe impedir que se puedan crear dos servicios con el mismo nombre
FS17	Sistema	El sistema debe sólo mostrar resultados que correspondan a lo que busque el usuario
FS18	Sistema	El sistema debe impedir el acceso al sistema si la contraseña dada no corresponde al usuario
FS19	Sistema	El sistema debe verificar que los datos que el usuario va a editar sean correctos
FS20	Sistema	El sistema debe poder ser accedido desde cualquier lugar que tenga Internet
FS21	Sistema	El sistema debe permitir cancelar una cuenta creada
FS22	Sistema	El sistema debe eliminar la cuenta de la base de datos cuando esto sea solicitado
FS23	Sistema	El sistema debe mostrar un mapa geográfico de la ubicación del servicio

*Tabla 2. Especificación de Requerimientos Funcionales.*

### 2.1.2. Requerimientos No Funcionales

A continuación, en la siguiente tabla de requerimientos no funcionales, se presenta el identificador, atributo de calidad correspondiente y descripción de los requerimientos no funcionales; realizando su respectiva priorización más adelante.

ID	QA	Requerimiento
NF-1	Rendimiento	El sistema debe responder a una petición en 30 segundos o menos
NF-2	Rendimiento	El sistema debe desplegar la información de los establecimientos en 20 segundos o menos
NF-3	Rendimiento	El sistema debe desplegar la comprobación de registro de un alojamiento en 5 minutos o menos
NF-4	Escalabilidad	El sistema debe poder soportar al menos 50 usuarios

NF-5	Usabilidad	El sistema debe dar la opción de iniciar sesión
NF-6	Usabilidad	El sistema debe dar la opción de registrar cuenta
NF-7	Usabilidad	El sistema debe dar la opción de asociar una imagen con el servicio
NF-8	Usabilidad	El sistema debe dar la opción de asociar una imagen con el usuario
NF-9	Usabilidad	El sistema debe mostrar un calendario para seleccionar la fecha de nacimiento del usuario.
NF-10	Integridad Conceptual	El sistema debe mostrar un cuadro de texto donde se pueda digitar el usuario
NF-11	Integridad Conceptual	El sistema debe mostrar un cuadro de texto, donde se pueda digitar la contraseña de usuario
NF-12	Integridad Conceptual	El sistema debe ser implementado con un paradigma orientado a objetos
NF-13	Integridad Conceptual	El sistema debe permitir poner la dirección del lugar donde se presta el servicio
NF-14	Integridad Conceptual	El sistema debe tener los colores definidos de SOLVO
NF-15	Integridad Conceptual	El sistema debe mostrar las notificaciones con un color que resalte con respecto a la vista de la página
NF-16	Integridad Conceptual	El sistema debe tener un botón para aceptar los cambios realizados
NF-17	Integridad Conceptual	El sistema debe tener una sección para editar cuentas
NF-18	Integridad Conceptual	El sistema debe desplegar un campo de texto donde se pueda agregar un comentario en una publicación
NF-19	Seguridad	El sistema debe crear una copia de seguridad
NF-20	Seguridad	El sistema debe guardar información necesaria en el dispositivo móvil en caso de alguna pérdida de señal

NF-21	Seguridad	El sistema debe validar el registro del usuario por medio de un correo electrónico
NF-22	Seguridad	El sistema debe encriptar las contraseñas de las cuentas de los usuarios almacenadas
NF-23	Confiabilidad	El sistema debe notificarle al usuario si cometió un error al ingresar su correo electrónico
NF-24	Confiabilidad	El sistema debe validar que el correo electrónico dado por el usuario corresponde a la estructura normal de un correo electrónico.
NF-25	Confiabilidad	El sistema debe verificar que la contraseña del usuario tenga mínimo 8 caracteres.
NF-26	Confiabilidad	El sistema debe notificarle al usuario si el correo ingresado se encuentra en uso.
NF-27	Confiabilidad	El sistema debe notificarle al usuario si ha realizado su registro correctamente.
NF-28	Confiabilidad	El sistema debe mostrar un enlace a los términos y condiciones de SOLVO durante el proceso de registro
NF-29	Confiabilidad	El sistema debe mostrar que durante la creación de la cuenta, se aceptan los términos y condiciones
NF-30	Confiabilidad	El sistema debe tener una vista que muestre a los términos y condiciones de agregar un servicio a la cuenta del usuario.

*Tabla 3. Especificación de Requerimientos No Funcionales.*

### 2.1.3. Priorización

Para la priorización se determinó por medio del número de dependencias para cada uno de los requerimientos que se tienen, y así poder obtener un grafo para lograr visualizar cuál requerimiento debería ser empleado antes y cuál requerimiento después, dando importancia de precedencia y procedencia para los mismos.

Esta priorización se llevó a cabo en el documento de *Especificación de Requerimientos de Software (SRS)* en la sección *3.2.1.3. Prioridad en desarrollo* para los requerimientos funcionales, y en la sección *3.3.1.3. Prioridad en desarrollo* para los

requerimientos no funcionales, el documento se puede encontrar en la página web de donde se extrajo el presente documento.

## **2.2. Pruebas**

Las pruebas de software son aquellos procesos de análisis de un sistema que brindan la detección de diferencias, con el objetivo de maximizar la cantidad de defectos descubiertos. A continuación se expondrán las distintas pruebas que se llevarán a cabo (*Peña Melo, 2007*).

### **2.2.1. Pruebas Unitarias**

Las pruebas unitarias son aquellas realizadas sobre las funciones del programa, y tienen el cargo de probar por unidades individuales. (*Microsoft, 2015*) Estas pruebas se pueden realizar sin una intervención manual, se pueden repetir cuantas veces se quiera (*Universidad de Alicante*). Estas pruebas deben enfocarse en probar la funcionalidad de objetos o métodos (*Sommerville, 2011*).

Dichas pruebas brindan beneficios debido a que la vida del desarrollador se facilita, refiriéndose a la calidad del código, se fomenta el cambio y la refactorización, se disminuyen los problemas y tiempos dedicados a la integración, y probar el sistema por partes sin necesidad de tenerlo completo (*Universidad de Alicante*).

### **2.2.2. Pruebas de Integración**

Las pruebas de integración se centran en detectar nuevos defectos en pequeños grupos de componentes. Estas pruebas se componen de dos sub grupos:

- **Pruebas de gran explosión:** Se enfocan en realizar las pruebas individualmente por cada componente, y finalmente se realiza una juntado todos los componentes.
- **Pruebas de abajo hacia arriba:** Se enfoca en realizar una prueba de forma individual de la capa inferior, y luego ir realizando pruebas conjuntas con capas superiores. Esto se repite hasta que se combinen todos los componentes. (*Peña Melo, 2007*)

### **2.2.3. Pruebas de Validación**

Las pruebas de validación implican pruebas con las entradas correctas. Estas estimulan el sistema para generar los resultados correctos esperados. Además, el objetivo es validar que el sistema cumpla con sus requisitos y que sea lo suficientemente bueno para uso externo. Por medio de estas pruebas se espera que el sistema funcione correctamente

utilizando un conjunto determinado de casos de prueba que reflejen el uso esperado del sistema (Sommerville, 2011).

#### 2.2.4. Pruebas Funcionales

Las pruebas funcionales están enfocadas en verificar que las funciones en cada nivel de usuario trabajen según lo diseñado. En este nivel de prueba se encarga que los datos estén en el estado correcto, actualizados a la base de datos apropiada, entre otras cosas (Craig & Jaskiel, 2002).

Las pruebas funcionales se dividen en dos categorías:

- **Prueba funcional positiva:** Esta prueba implica el ejercicio de las funciones de la aplicación con una entrada válida y verificando que las salidas son correctas.
- **Prueba funcional negativa:** esta prueba implica el ejercicio de la aplicación funcionalidad que utiliza una combinación de entradas no válidas, condiciones de operación inesperadas y otros escenarios fuera de límites (Agarwal, Tayal, & Gupta, 2009).

### 3. Arquitectura

*"Una arquitectura de software de un programa o un sistema computacional es la estructura del sistema, la cual comprende elementos de software, las propiedades externamente visibles de esos elementos, y las relaciones entre ellos" (Bass, Clements, & Kazman, 2003).*

Por medio de esta arquitectura del sistema, se busca mostrar su comportamiento en términos generales, y lograr presentar un diseño abstracto del sistema. La siguiente arquitectura presenta un modelo de "Vista 4+1", debido a que provee un Marco de Trabajo (*framework*) de diferentes puntos de vista para especificar los sistemas de software orientados a objetos. Este modelo de arquitectura permite relacionar las 4 vistas (vista lógica, vista de desarrollo, vista de proceso y vista física) las cuales se diferencian por un metalenguaje que tiene cada una, y esta relación ayuda a entender a nivel macro de cómo surgiría el sistema idealizado (Albin, 2003).

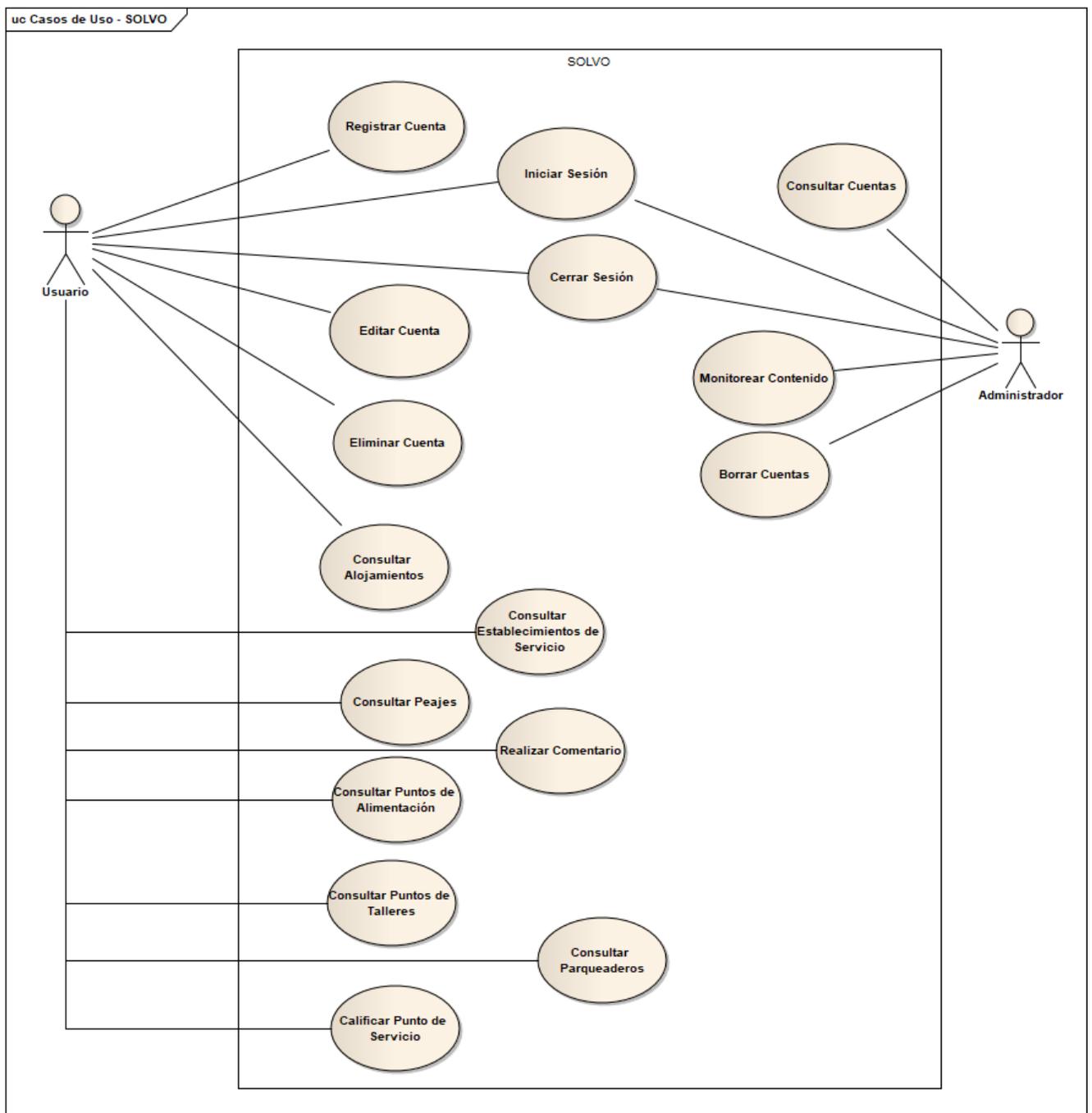
#### 3.1. Diagrama de Casos de Uso

Este diagrama de casos de uso junta las otras vistas, representando los requerimientos funcionales clave del sistema, y actúa como una especie de guía que une los elementos de las diferentes vistas (Albin, 2003).

En la siguiente ilustración, se muestra el diagrama de casos de uso del sistema SOLVO, en el cual se visualiza cada caso de uso descrito en el documento SRS al igual que las relaciones con los actores del sistema. Además se muestran los actores que interactúan con el sistema SOLVO y cómo estos se relacionan con los casos de uso (Los casos de uso se encuentran especificados a groso modo en [Anexo Documento CU SOLVO 0.7](#)).

En el diagrama de casos de uso hay 2 actores los cuales son:

- Usuario
- Administrador



*Ilustración 2. Diagrama de casos de Uso SOLVO. Elaborado con Enterprise Architect*

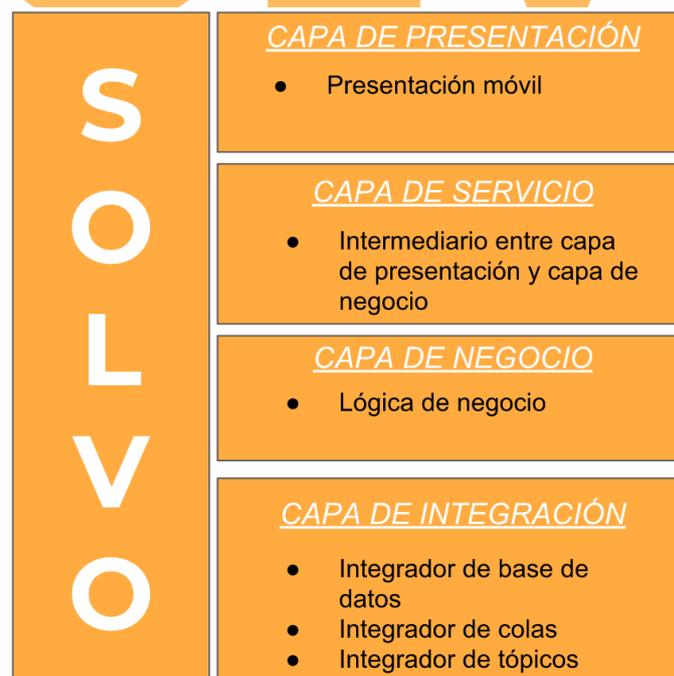
## 3.2. Vista Lógica

### 3.2.1. Visión General

La vista lógica del sistema describe los elementos arquitectónicamente significativos de la arquitectura y las relaciones entre ellos. Además captura esencialmente la estructura de la aplicación por medio de un análisis en el modelo de lógica de negocio usando diagramas de clase o sus equivalentes (Albin, 2003; Gorton, 2011).

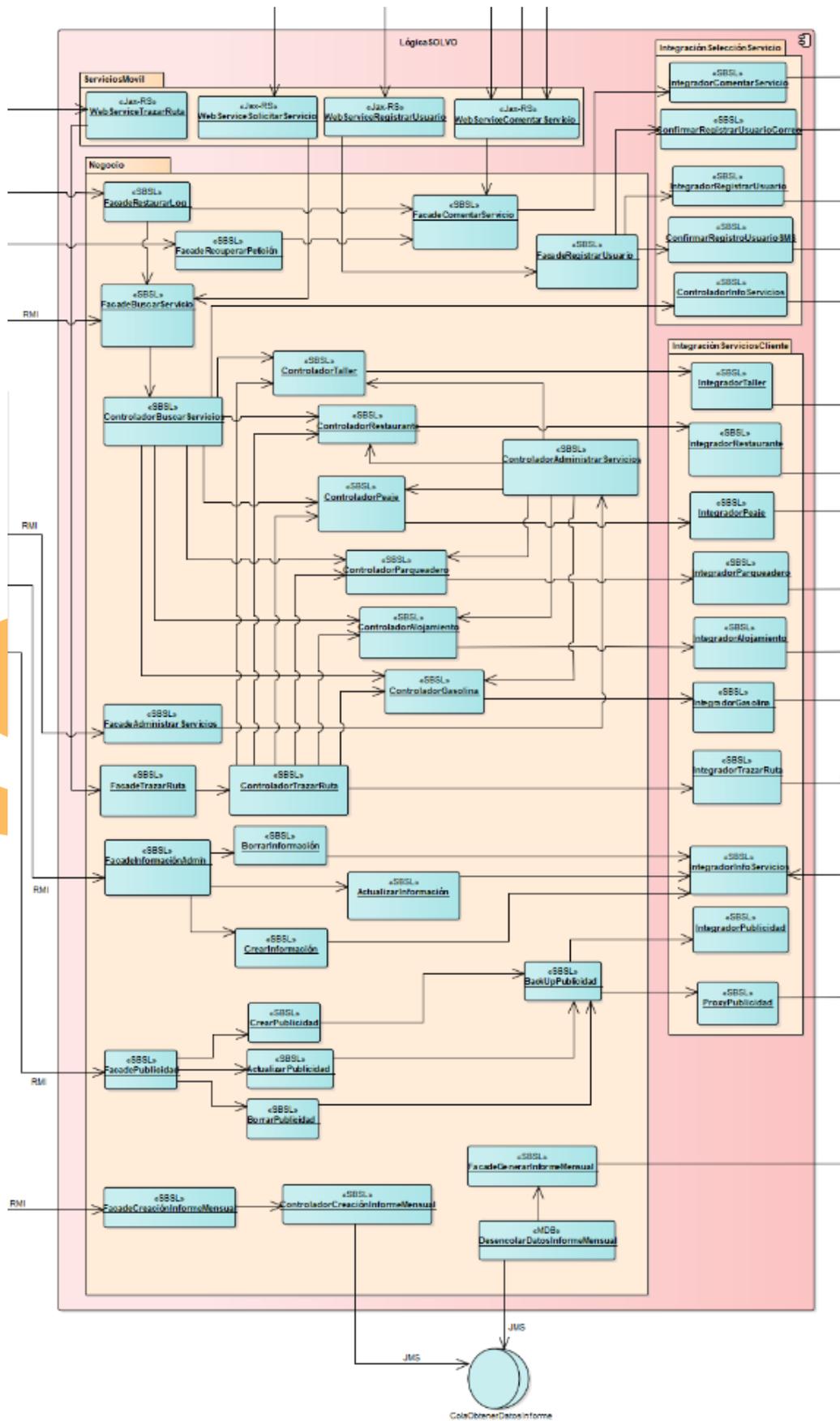
La arquitectura principal del sistema SOLVO es la de un sistema dividido en capas. El sistema SOLVO está basado en un estilo de capas, el cual descompone un sistema y de esta forma cada una de las capas tiene una responsabilidad asignada. A continuación, se detalla el contenido de dichas capas:

- **Capa de presentación:** En esta capa se prestan los servicios al usuario, además de la presentación de la información.
- **Capa de negocio:** En esta capa se encuentra la lógica del sistema.
- **Capa de integración:** Esta capa se encarga de la comunicación con los sistemas externos de bases de datos, los sistemas de mensajería, los sistemas de transacciones y demás sistemas externos.
- **Capa de servicio:** En esta capa se encuentran los controladores de servicio que permiten conectar algunas capas externas con la capa de negocio.



*Ilustración 3. Visión General SOLVO. Elaboración propia.*





*Ilustración 5 Lógica SOLVO*

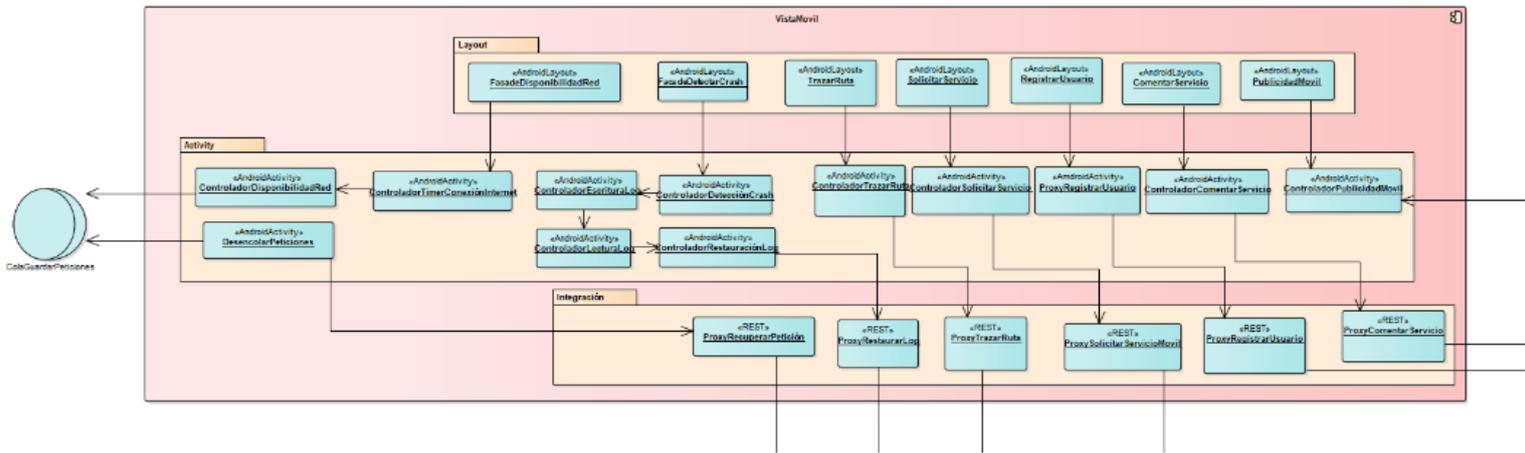


Ilustración 6 Vista Móvil

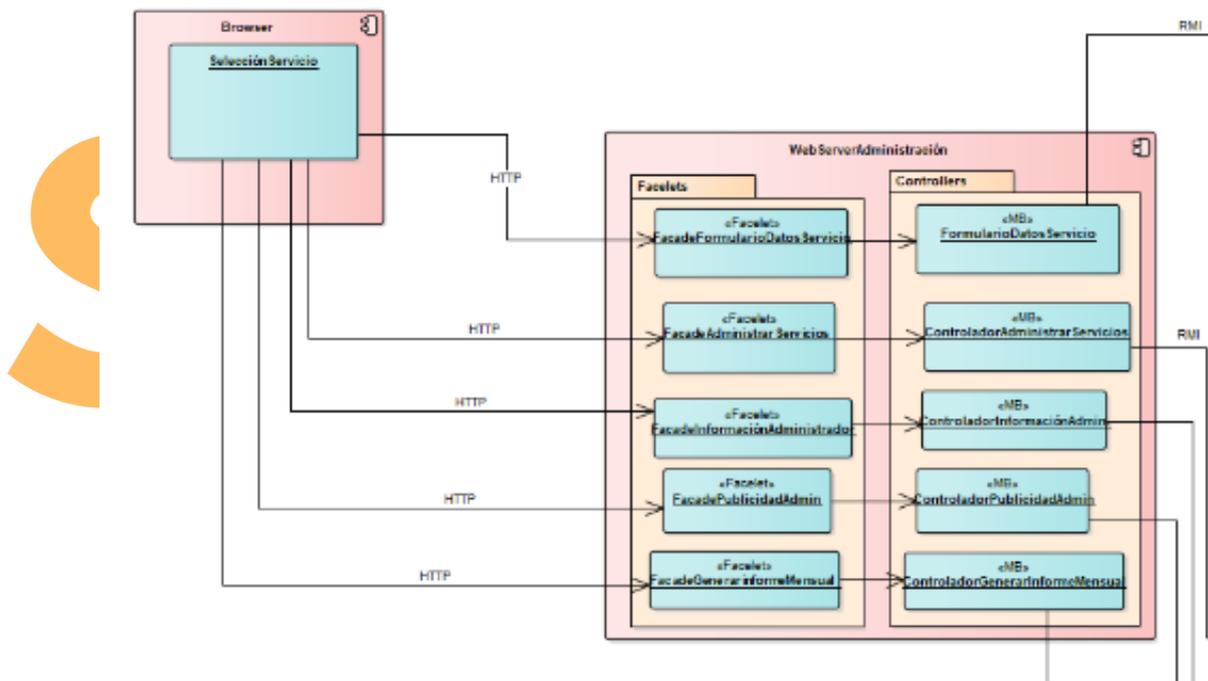


Ilustración 7 Administrador SOLVO

### 3.2.2. Casos de Uso Arquitecturalmente Significantes

Ver anexo de Casos de Uso Arquitecturalmente Significantes en donde se encuentran los 6 casos más importantes de SOLVO, brindando el diagrama de tipo lógico y explicando las capas compuestas en el diagrama. ([Anexo Documento CU ArqSig 0.2](#))

### 3.3. Vista Física

La vista física muestra cómo se mapean los principales procesos y componentes en el hardware del sistema. Podría mostrar, cómo se distribuyen la base de datos y los servidores web para la aplicación a través de varias máquinas servidoras. (*Gorton, 2011*)

Además, en la vista física se provee un contexto global del sistema, teniendo en cuenta los componentes físicos, las conexiones, y los nodos que constituyen el sistema.

La vista física indica requerimientos no funcionales, y brinda una importancia referida a la implementación y el despliegue de configuraciones que indicarán el desempeño de los requerimientos relacionados (*Albin, 2003*). El diagrama de la vista física está en el siguiente diagrama:

# SOLVO

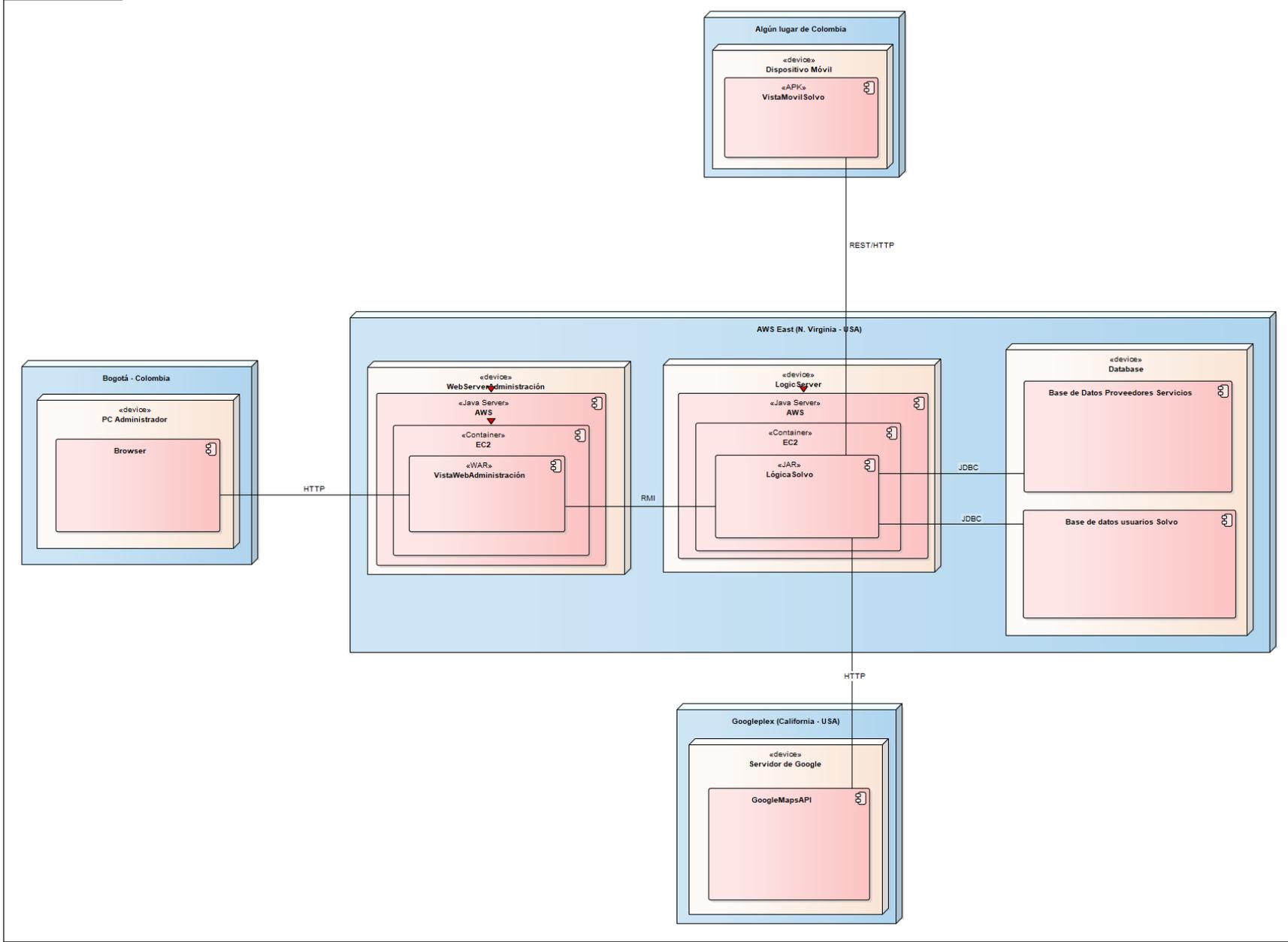


Ilustración 8. Vista Física del prototipo SOLVO. Elaboración Propia.

### 3.4. Vista de Procesos

La vista de procesos se enfoca en describir la concurrencia y las comunicaciones elementos de una arquitectura. En aplicaciones semejantes, las principales preocupaciones son la descripción de componentes multiproceso o replicados, y los mecanismos de comunicación sincrónicos o asincrónicos utilizados, además de enfocarse en la concurrencia, la sincronización y la distribución en aspectos del diseño. (Albin, 2003; Gorton, 2011)

Asimismo, en la vista de procesos se presenta de manera general, cuales son las acciones que el usuario puede realizar durante la utilización del sistema. Esto es, mostrar que cuáles son las posibles operaciones realizadas por el usuario y cuál es la salida o resultado de esta acción.

La vista de procesos está descrita en muchos niveles de abstracción, e indica cómo los objetos lógicos interactúan en orden de producir un funcionamiento requerido del sistema (autonomía, persistencia, subordinación, y distribución). (Albin, 2003)

Para esta sección se decidió realizar el modelo BPM de los casos de uso del sistema, para así exponer el proceso que llevará a cabo al poner en funcionamiento a cada uno de estos. Estos diagramas se encuentran a continuación:

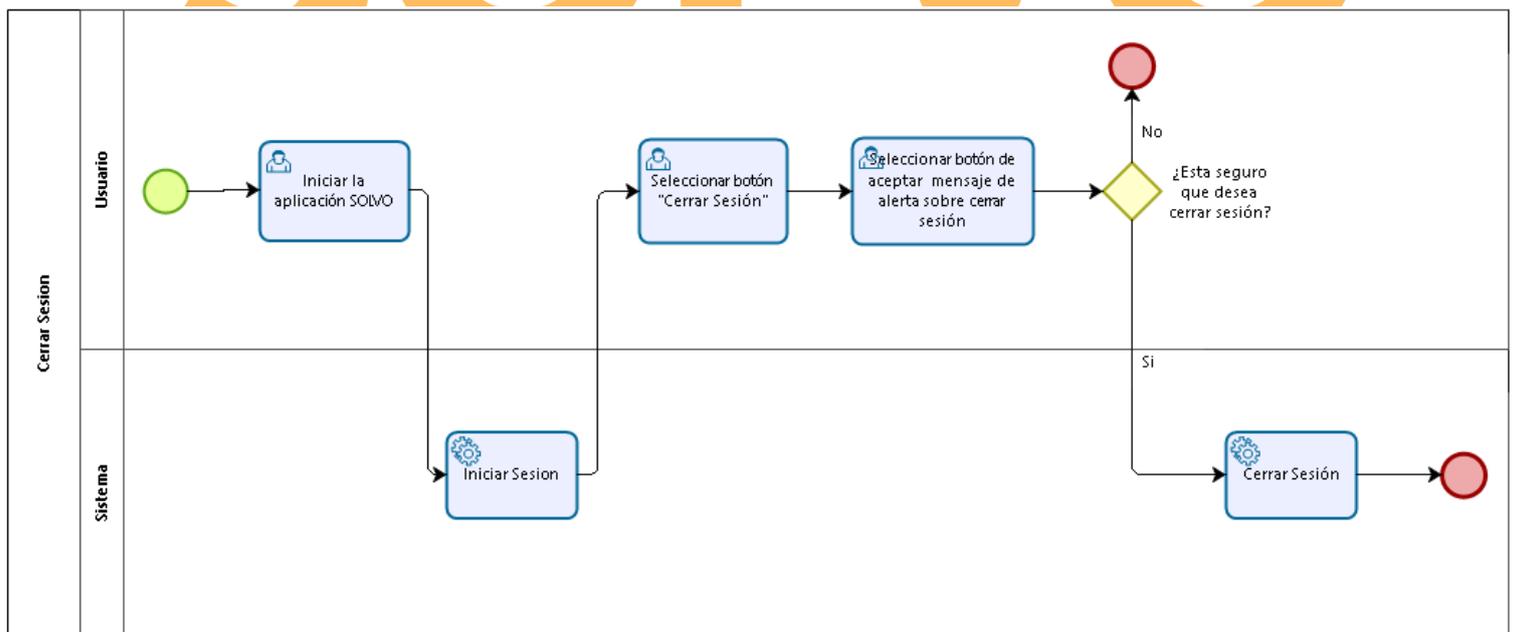
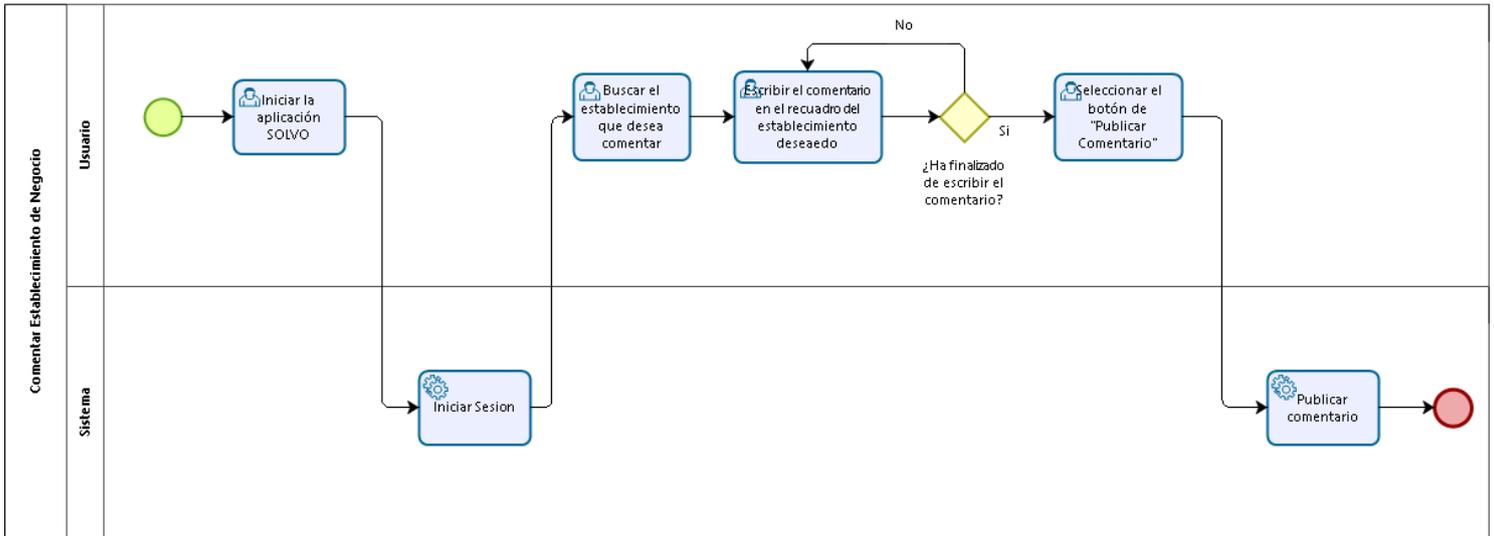
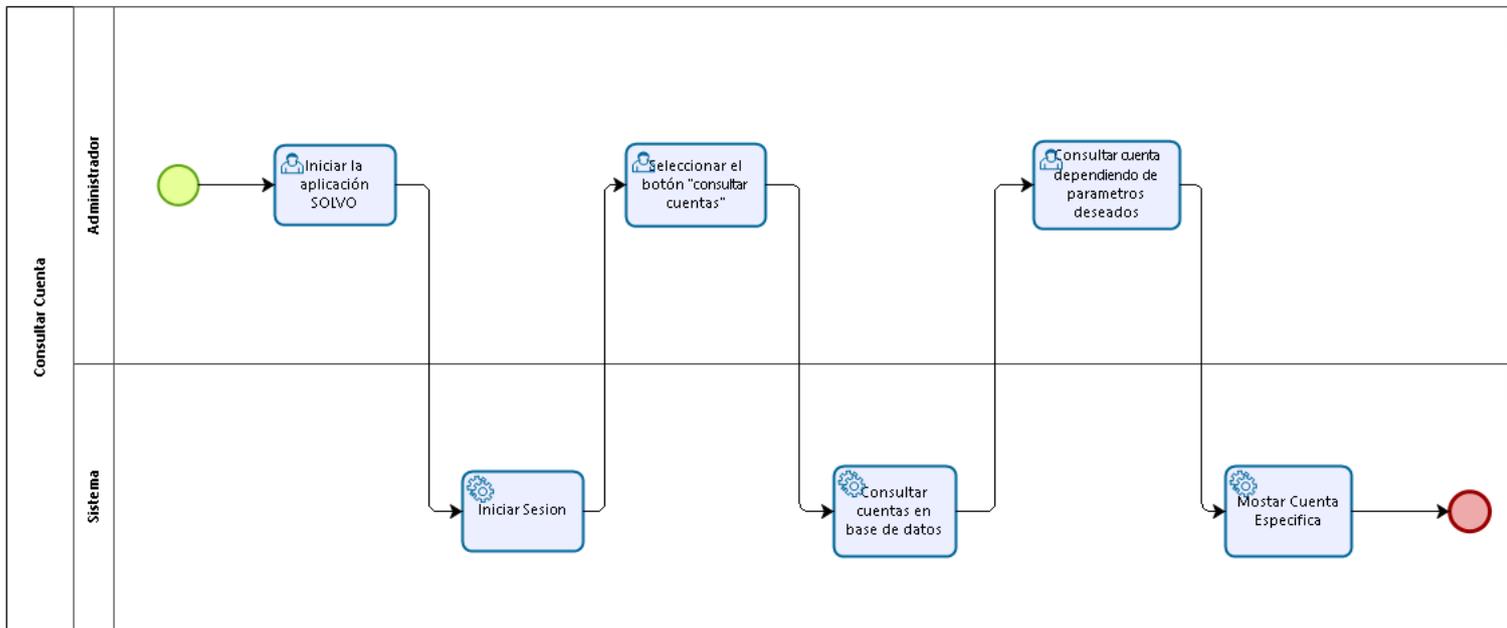


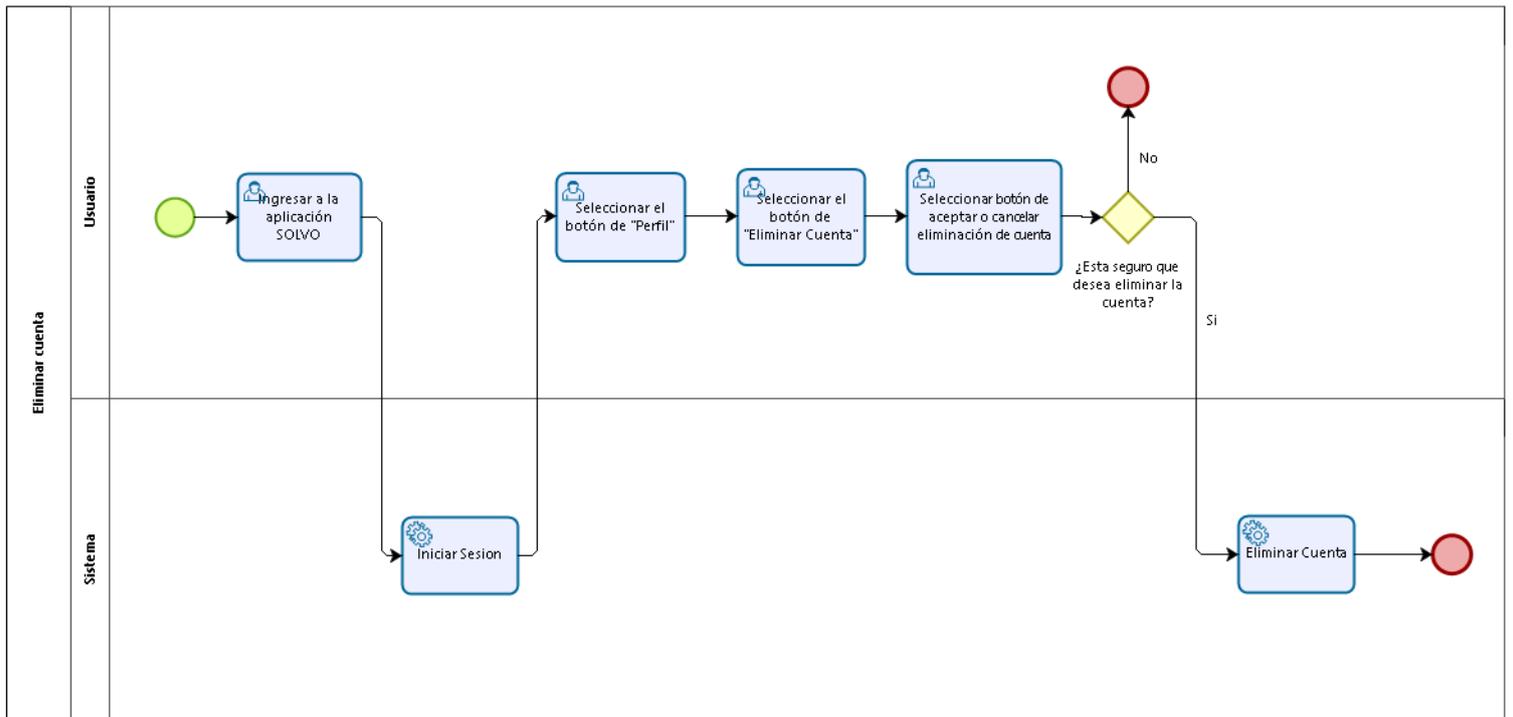
Ilustración 9. Vista de procesos de función "Cerrar Sesión". Elaborado en Bizagi.



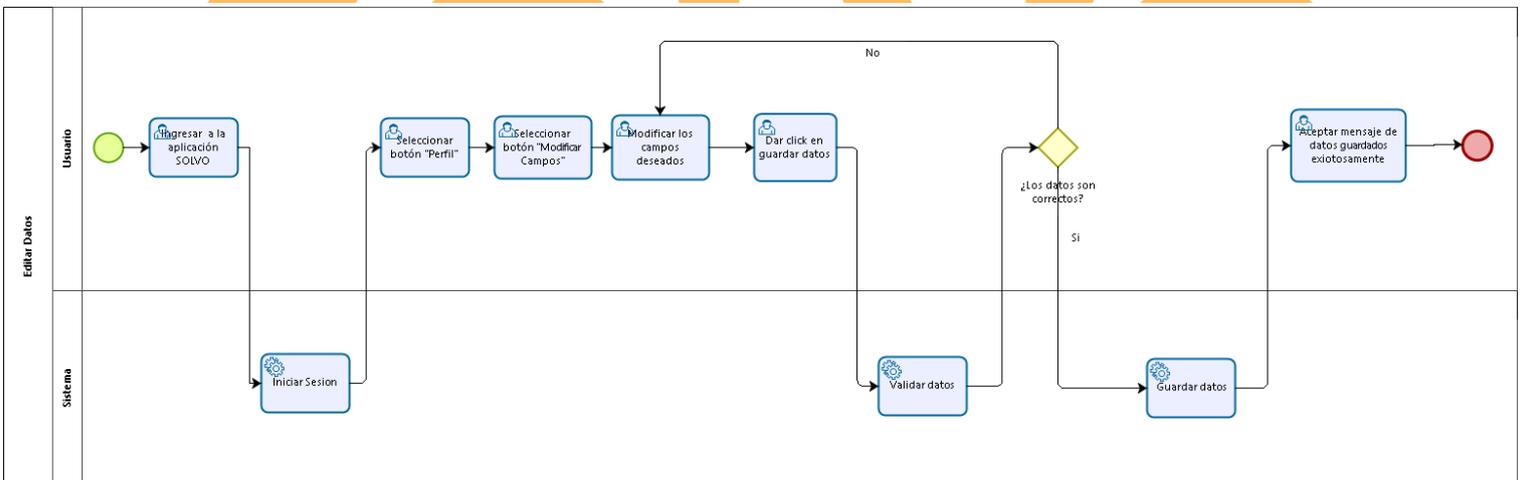
**Ilustración 10.** Vista de procesos de función "Comentar Establecimiento de Negocio". Elaborado en Bizagi.



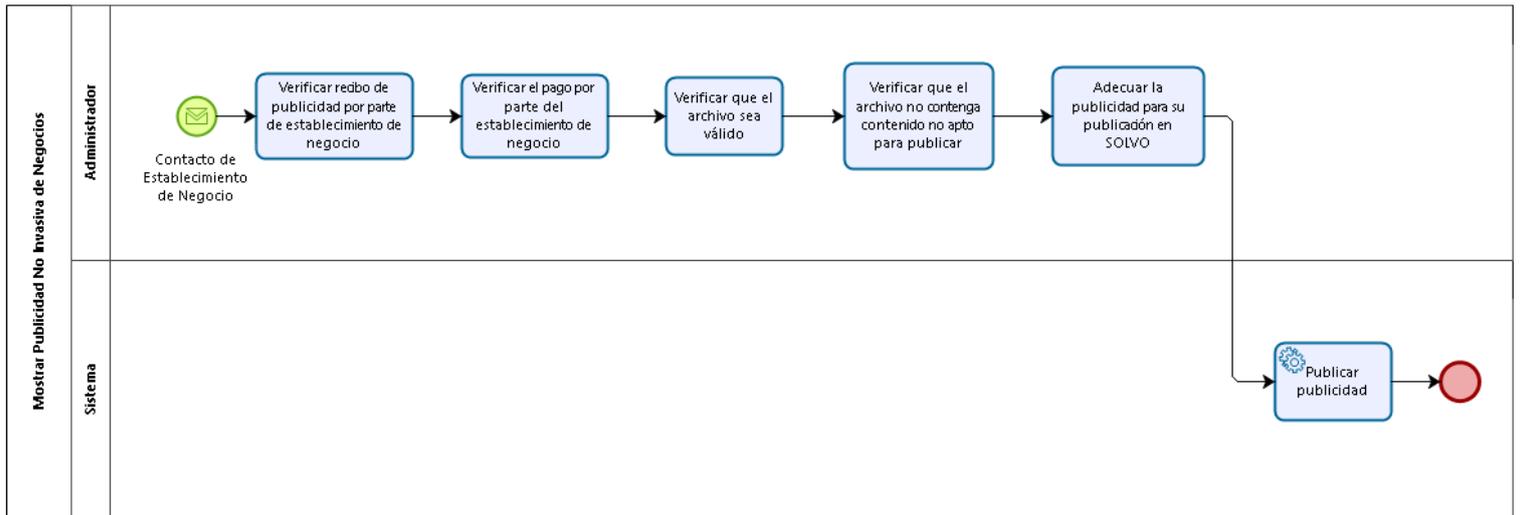
**Ilustración 11.** Vista de procesos de función "Consultar Cuenta". Elaborado en Bizagi.



**Ilustración 12.** Vista de procesos de función "Eliminar Cuenta". Elaborado en Bizagi.

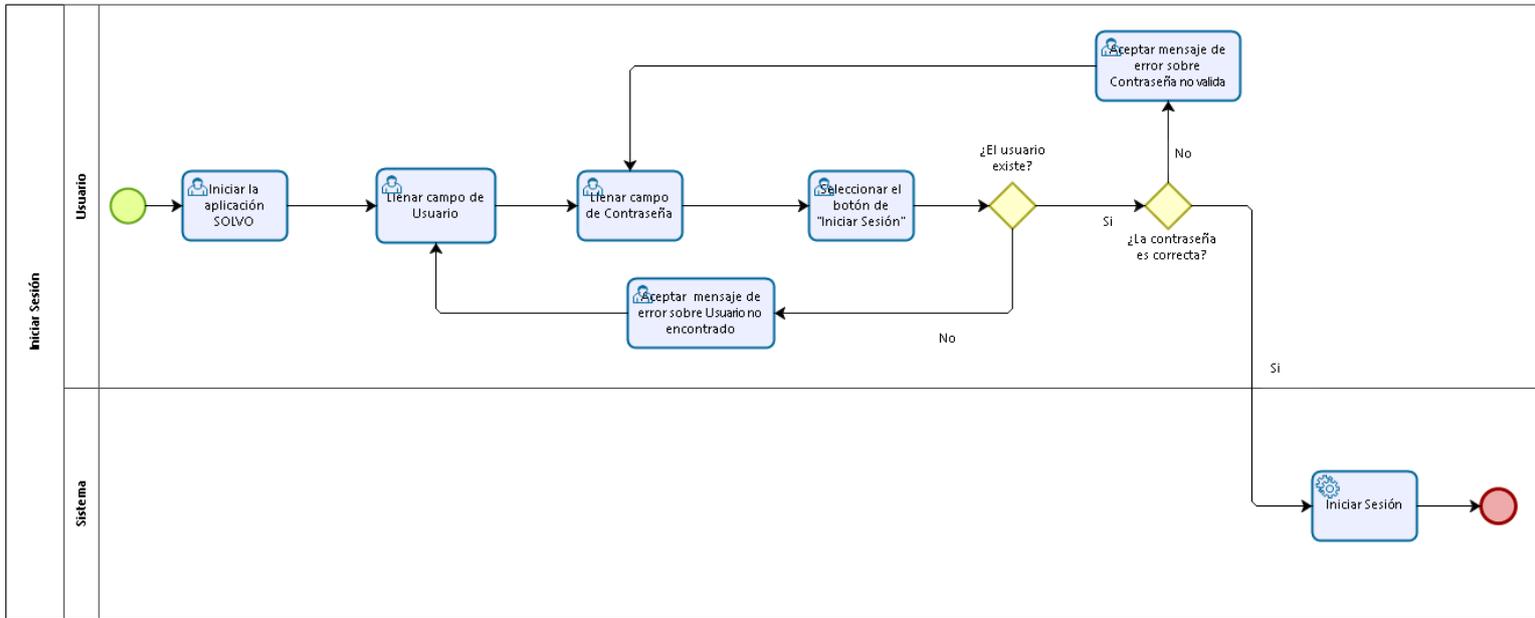


**Ilustración 13.** Vista de procesos de función "Editar Datos". Elaborado en Bizagi.

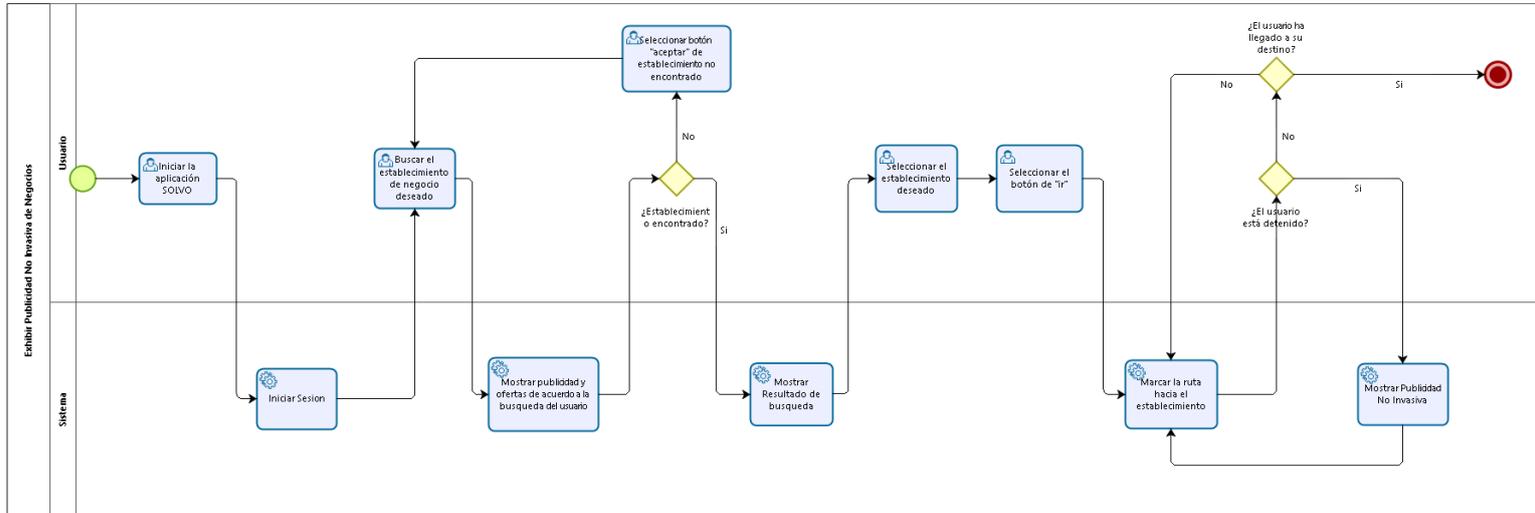


*Ilustración 14. Vista de procesos de función "Mostrar Publicidad No Invasiva de Negocio". Elaborado en Bizagi.*

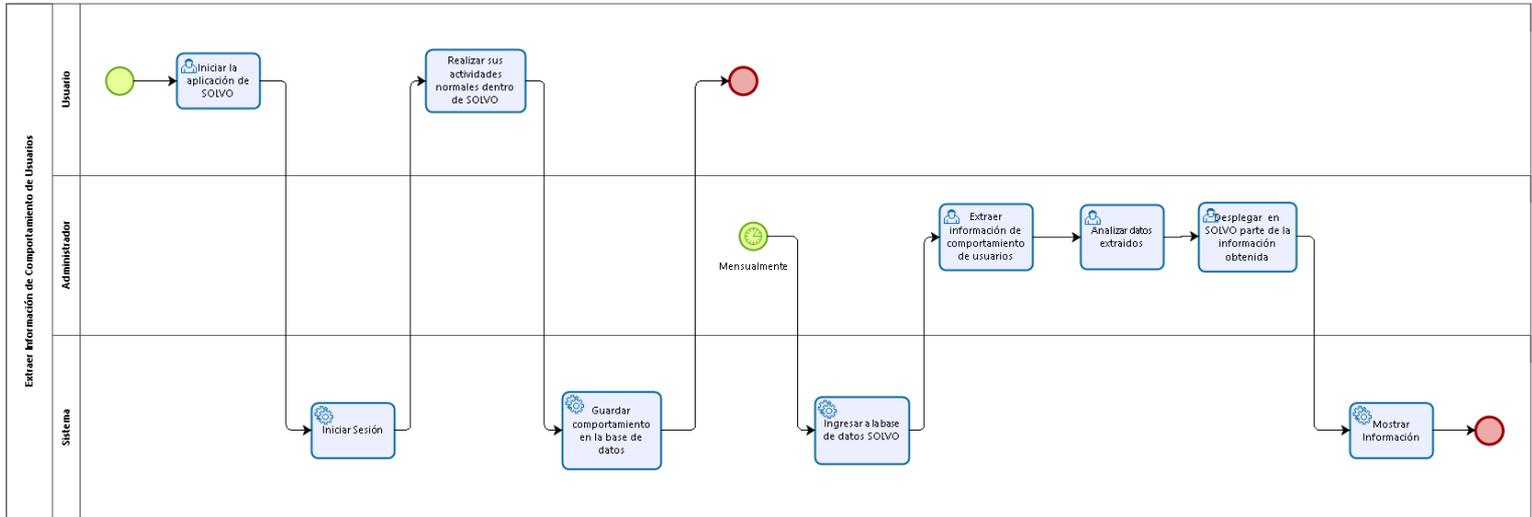
# SOLVO



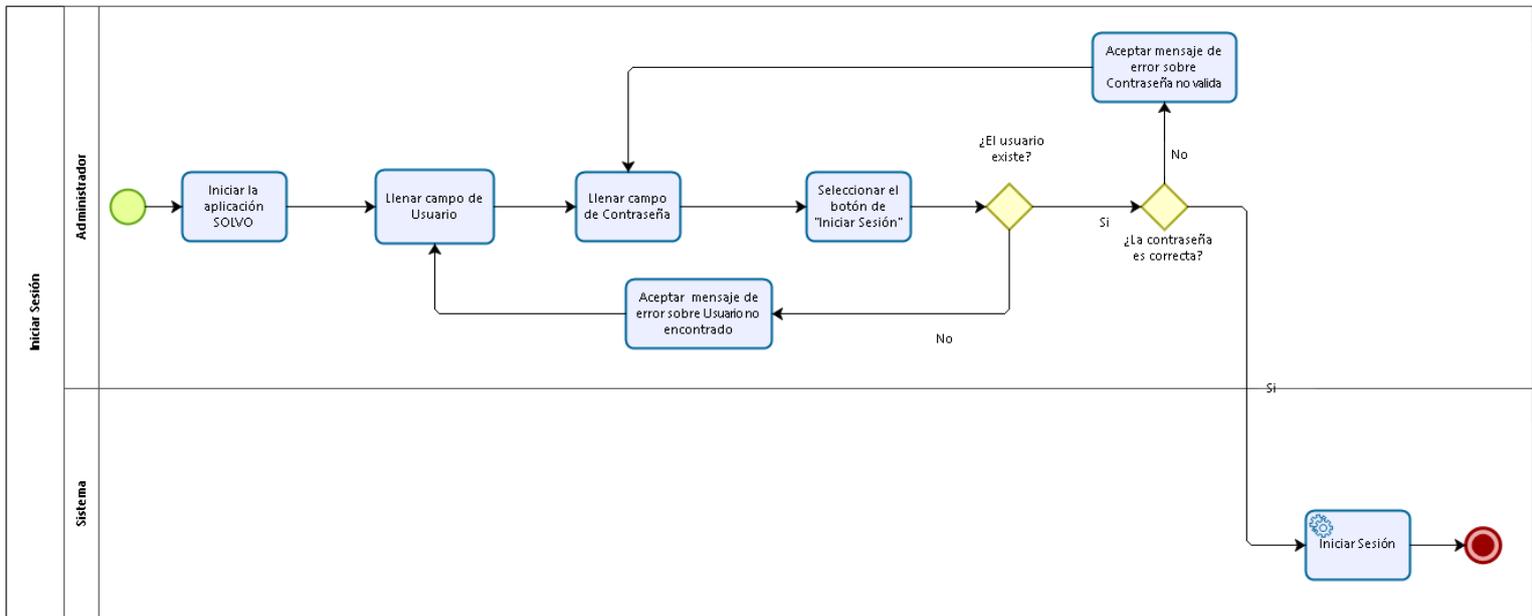
**Ilustración 15.** Vista de procesos de función "Iniciar Sesión". Elaborado en Bizagi.



**Ilustración 16.** Vista de procesos de función "Exhibir Publicidad No Invasiva de Negocio". Elaborado en Bizagi.



**Ilustración 17.** Vista de procesos de función "Extraer Información de Componentes de Usuarios". Elaborado en Bizagi.



**Ilustración 18.** Vista de procesos de función "Iniciar Sesión Administrador". Elaborado en Bizagi.

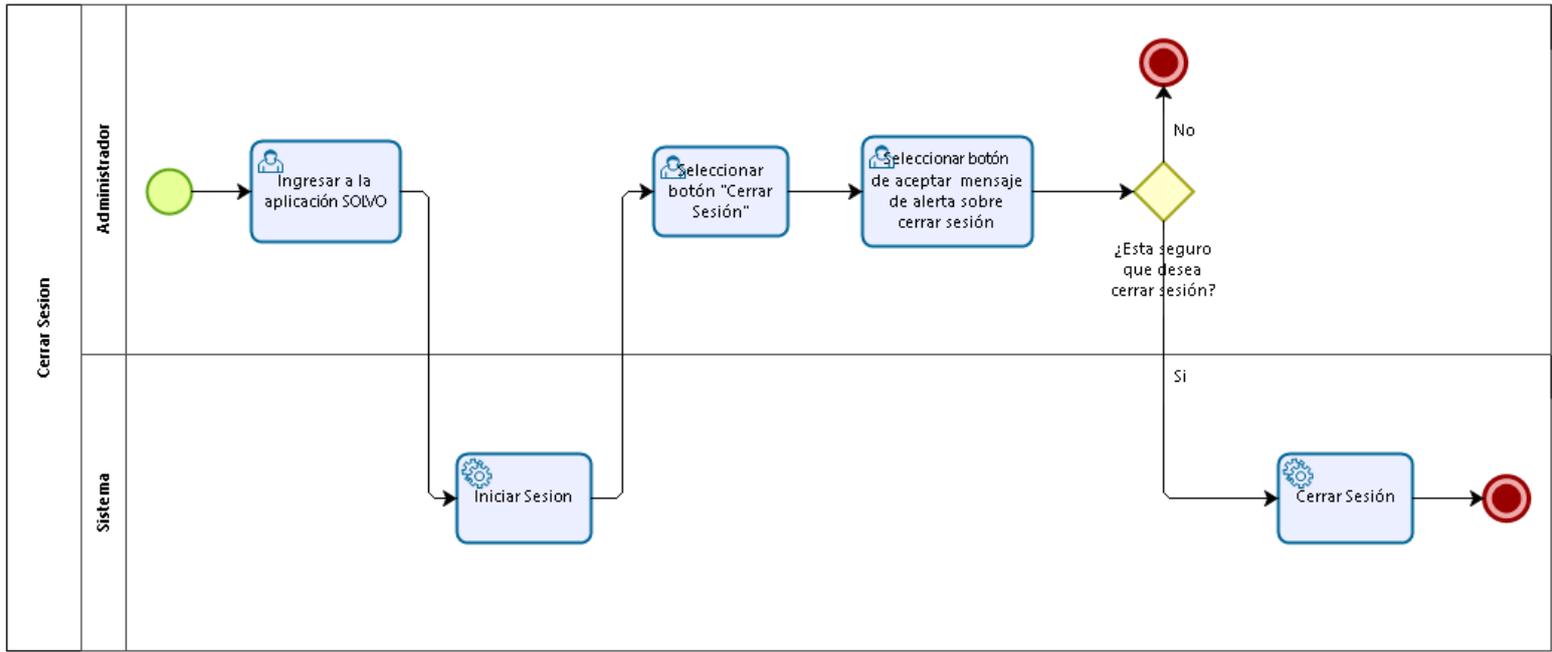


Ilustración 19. Vista de procesos de función "Cerrar Sesión Administrador". Elaborado en Bizagi.

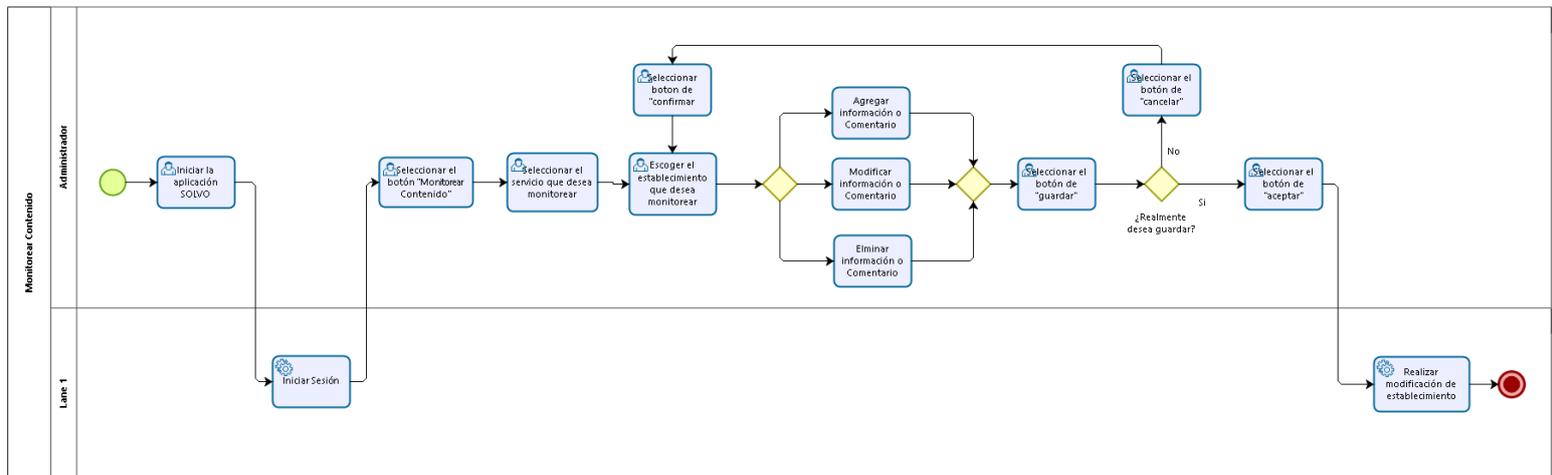


Ilustración 20. Vista de procesos de función "Modificar Contenido". Elaborado en Bizagi.

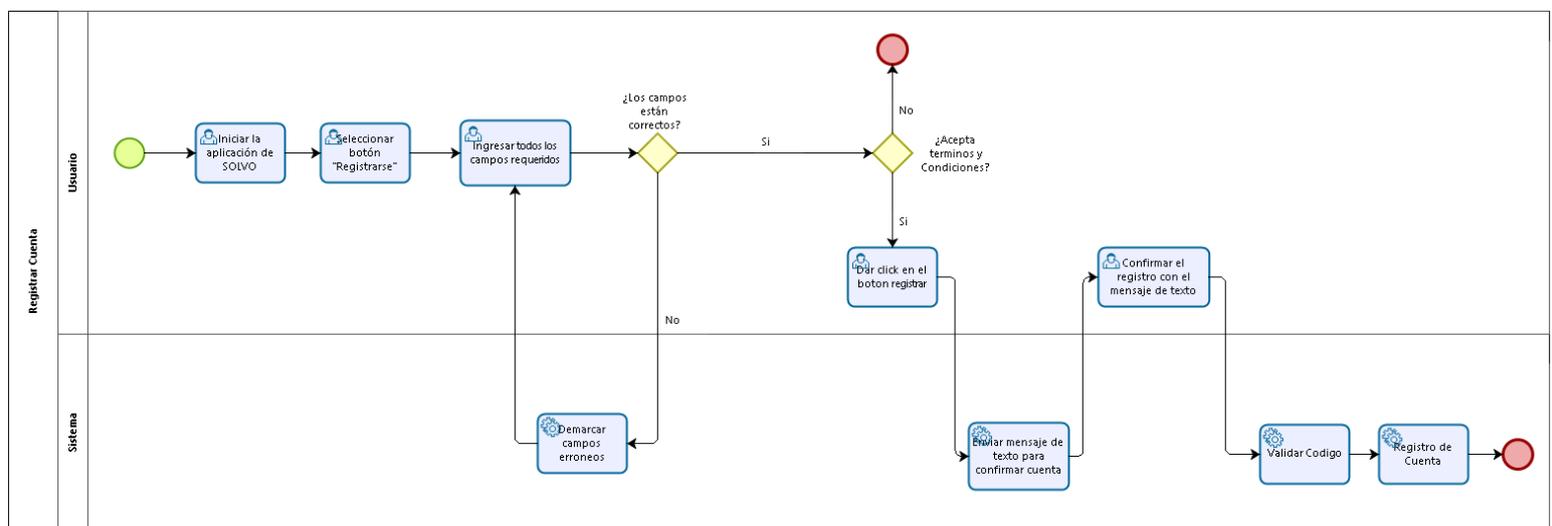
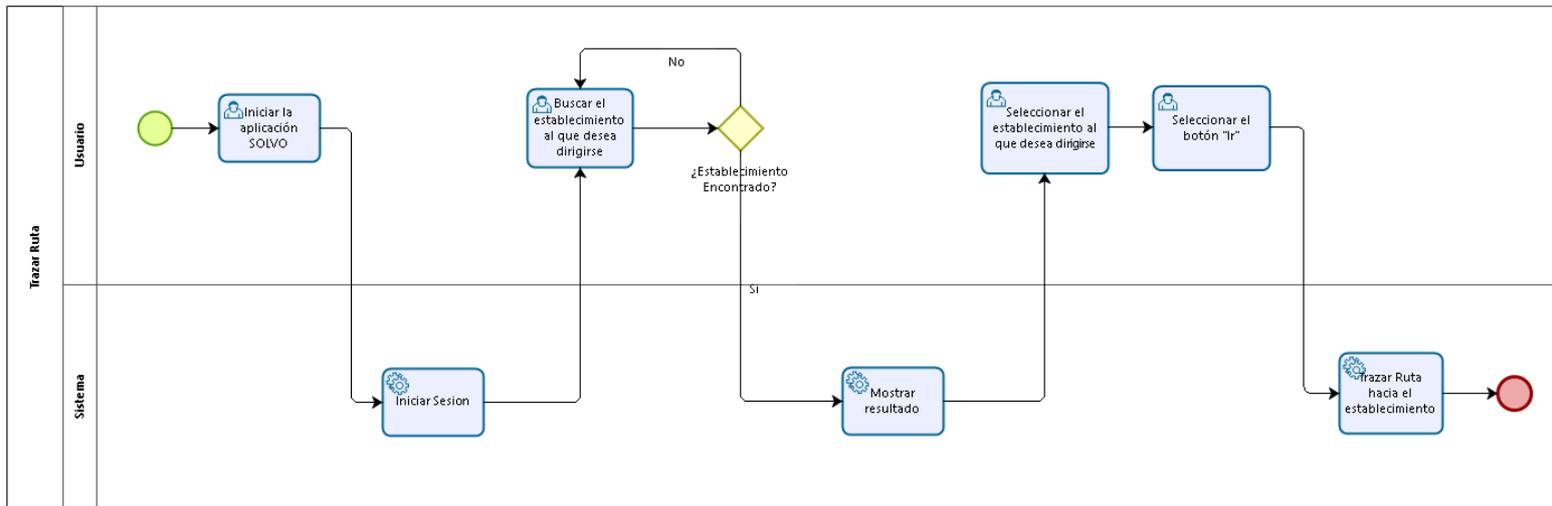
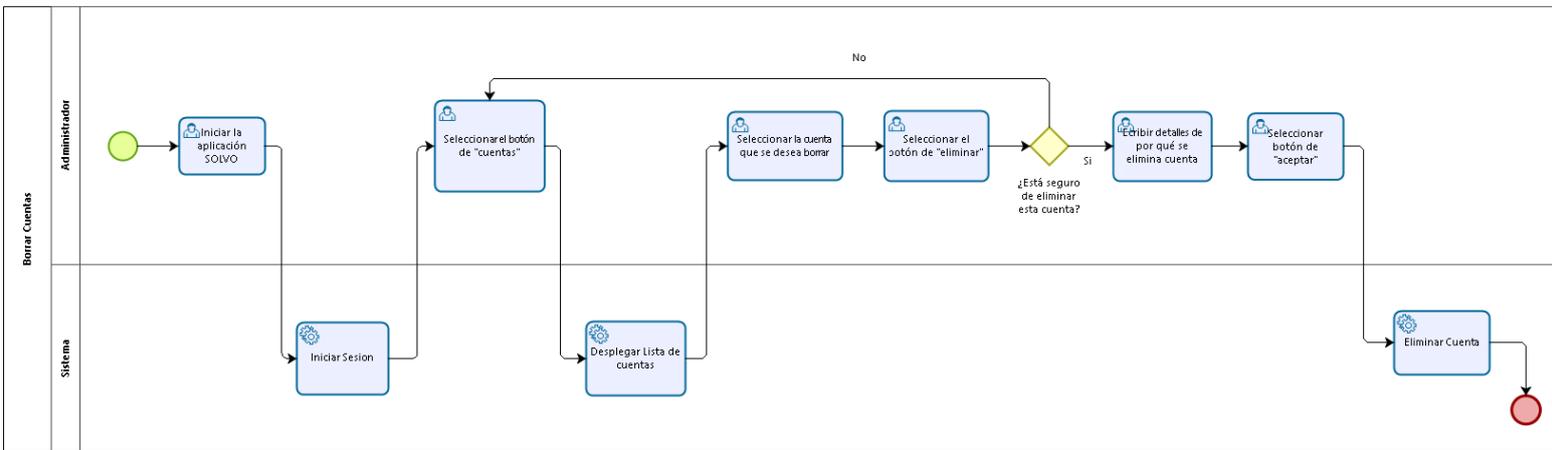


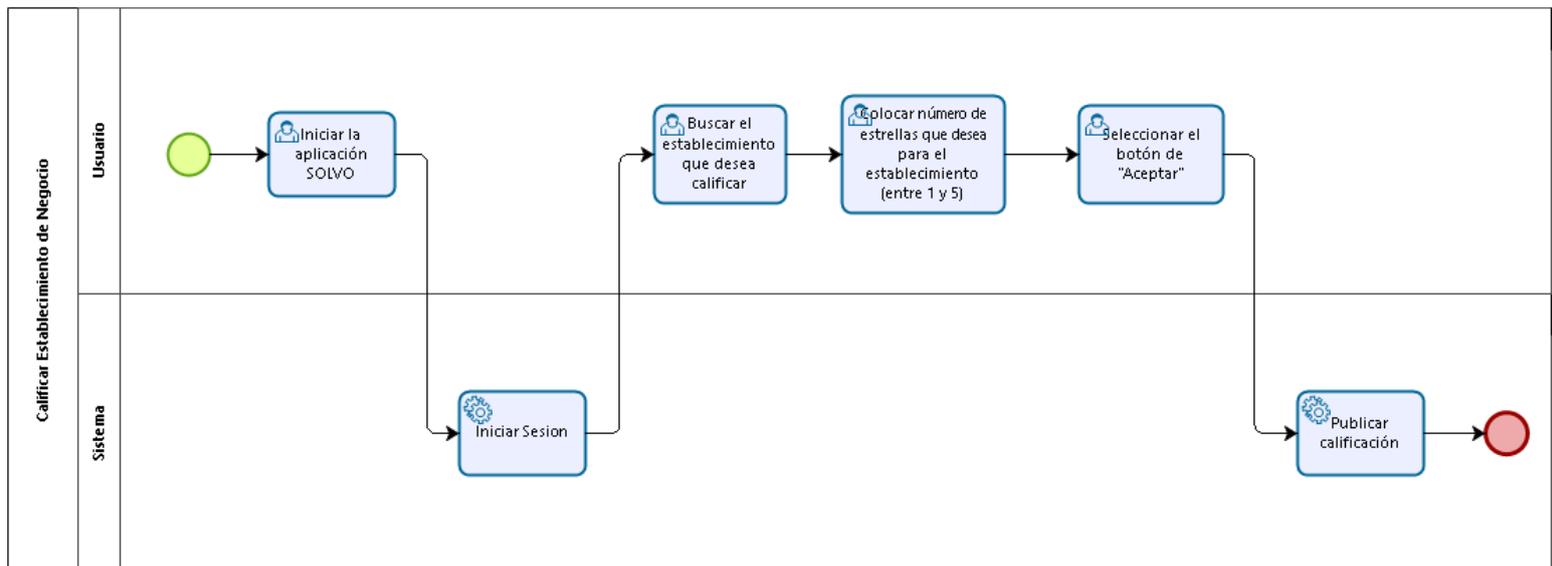
Ilustración 21. Vista de procesos de función "Registrar Cuenta". Elaborado en Bizagi.



**Ilustración 22.** Vista de procesos de función "Trazar Ruta". Elaborado en Bizagi.



**Ilustración 23.** Vista de procesos de función "Borrar Cuenta Administrador". Elaborado en Bizagi.



**Ilustración 24.** Vista de procesos de función "Calificar Establecimiento de Negocio". Elaborado en Bizagi.

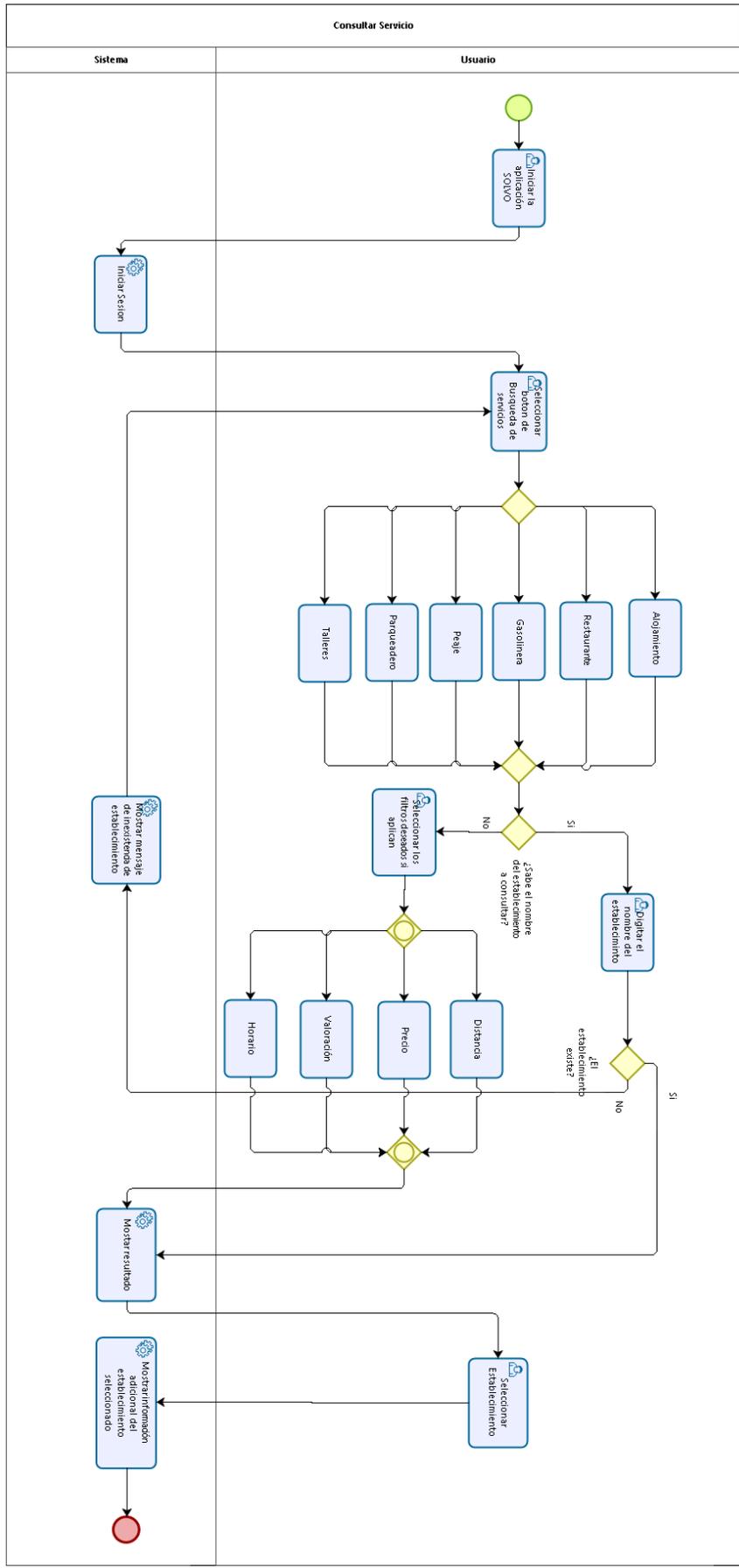


Ilustración 25. Vista de procesos de función "Consultar Servicio". Elaborado en Bizagi.

### **3.5. Vista de Despliegue**

Esta vista representa la organización estática del software con respecto al ambiente de desarrollo del software. Así mismo aborda la gestión de la configuración del software y las preocupaciones, como capacidad de construcción, mantenimiento, reutilización y la gestión de la configuración de las versiones del sistema, además se enfoca en la división de funcionalidades entre subsistemas.

Estos subsistemas están organizados internamente en capas jerárquicas en donde cada capa provee una interface a la siguiente capa superior, y el propósito de usar las capas es para minimizar dependencias entre cada subsistema o módulo (*Albin, 2003*).

# SOLVO

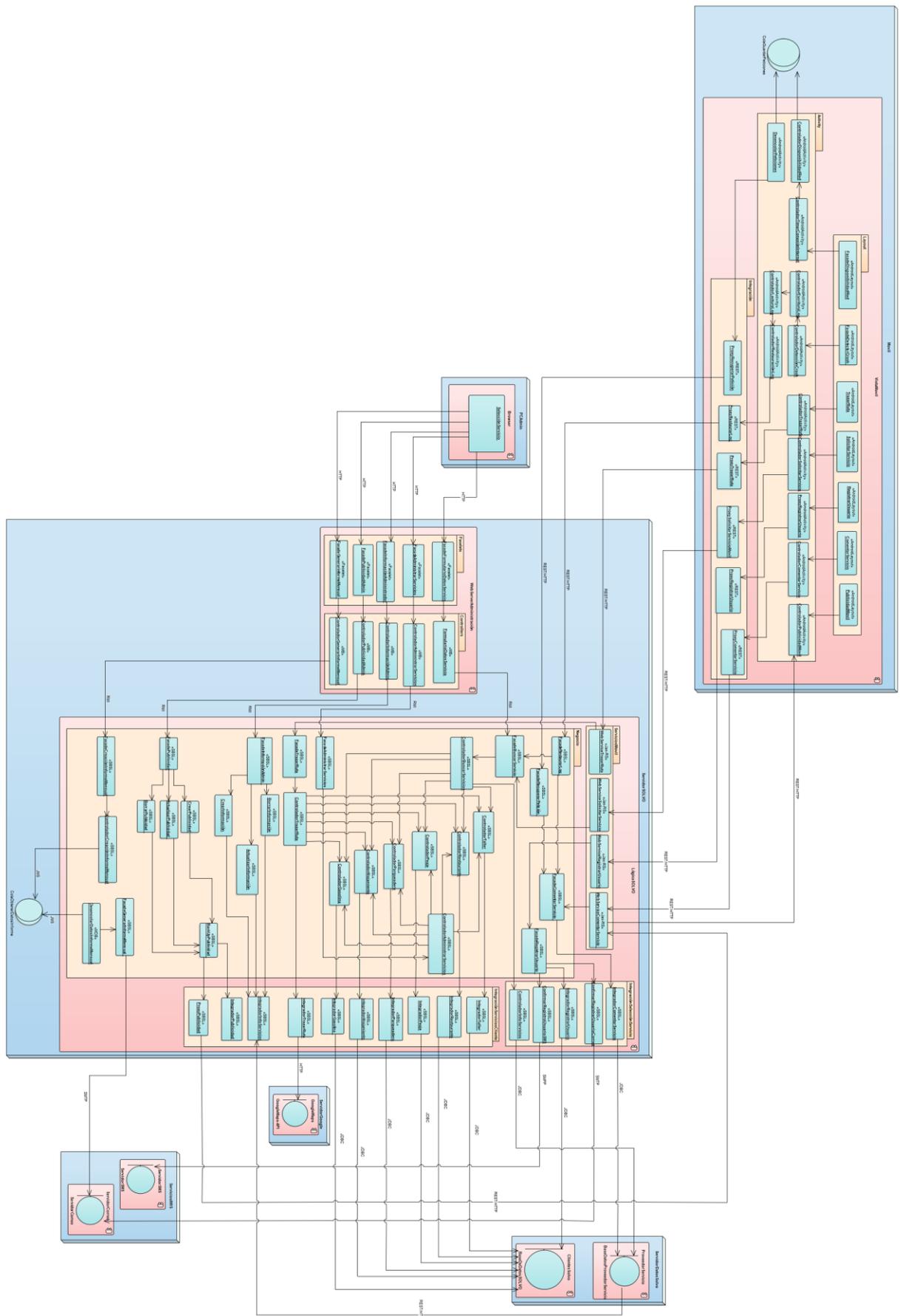


Ilustración 26. Vista Física de SOLVO. Elaboración en Enterprise Architect.

### 3.6. Vista de Datos

Esta vista brinda una descripción de la estructura que tendrá la base de datos del sistema, ofreciendo representación de datos de forma organizada, aumento de rendimiento del sistema interno, alta disponibilidad de los datos, reducción del espacio de almacenamiento, e inter relación y estructuración de los datos. (*Piattini Velthuis, 1998*)

Asimismo, esta vista brinda una abstracción de los datos, siendo esto el cómo los datos son almacenados y mantenidos. Para poder brindar una complejidad menor se debe emplear los tres niveles para su buen funcionamiento:

- **Nivel físico:** Este es el nivel más bajo de abstracción el cual describe el “cómo” los datos están almacenados actualmente. Este nivel describe la complejidad de nivel bajo de los datos estructurados en detalle.
- **Nivel lógico:** Este nivel es el siguiente más alto de abstracción el cual describe el “qué” datos son los almacenados en la base de datos y qué relaciones existen entre esos datos.
- **Nivel de vista:** Este es el nivel más alto de abstracción el cual describe solo la parte de la base de datos completa. Y así el nivel lógico use estructuras simples, la complejidad permanece debido a la variedad de información almacenada en una gran base de datos. Su abstracción existe para simplificar su interacción con el sistema. (*Silberschatz, Korth, & Sudarshan, 2006*)

A continuación, se presenta una ilustración referida a la vista de datos que se emplea en el sistema SOLVO, y a continuación se brinda una descripción de las entidades propuestas en el sistema.

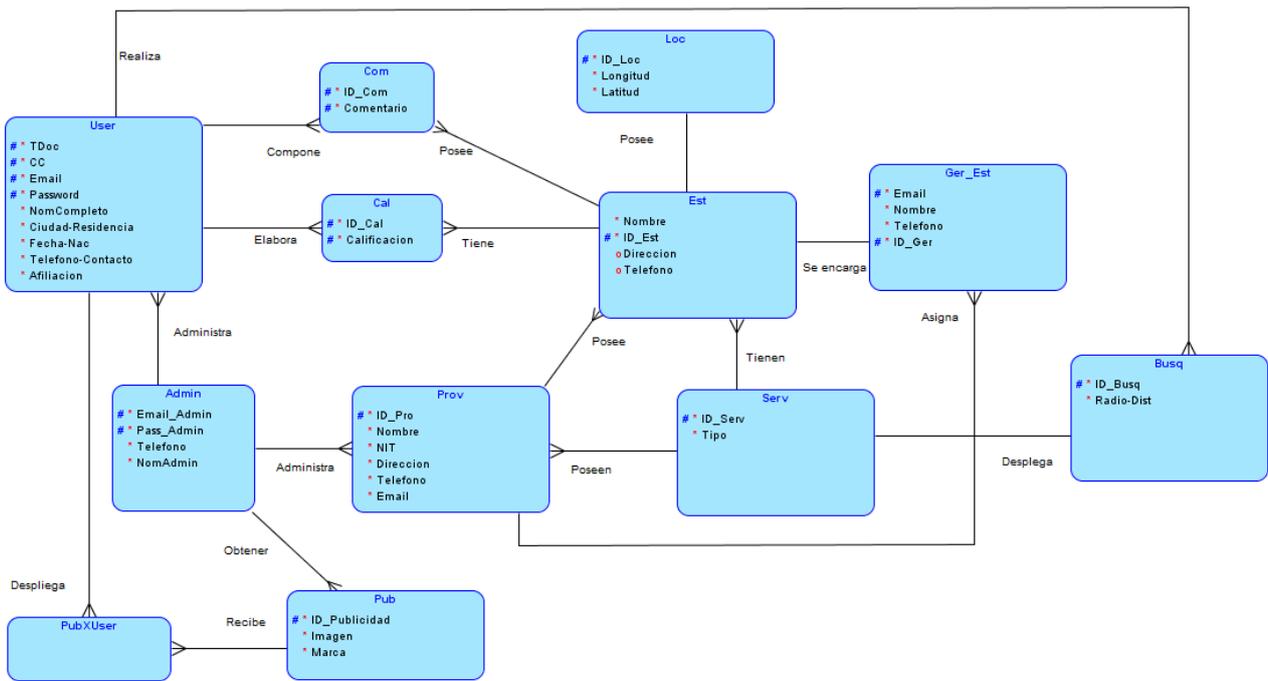


Ilustración 27. Vista de Datos del Sistema SOLVO.

# SOLVO

## 4. Diseño de Alto Nivel

### 4.1. Estructura del Sistema

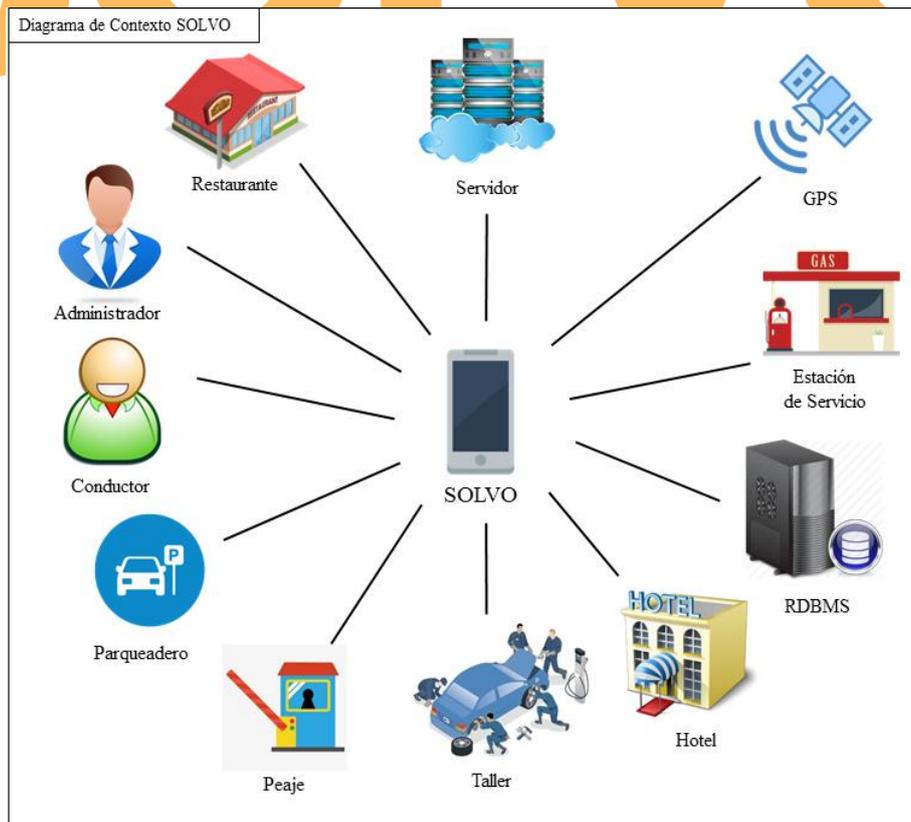


Ilustración 28. Diagrama de Contexto SOLVO. Elaborado con Enterprise Architect

En el diagrama de contexto se puede observar la estructura del sistema en alto nivel. En el centro del diagrama se encuentra SOLVO, a su alrededor, los actores y las entidades externas que interactúan con él. Se identificaron dos actores que tendrán interacción con el sistema, estos son:

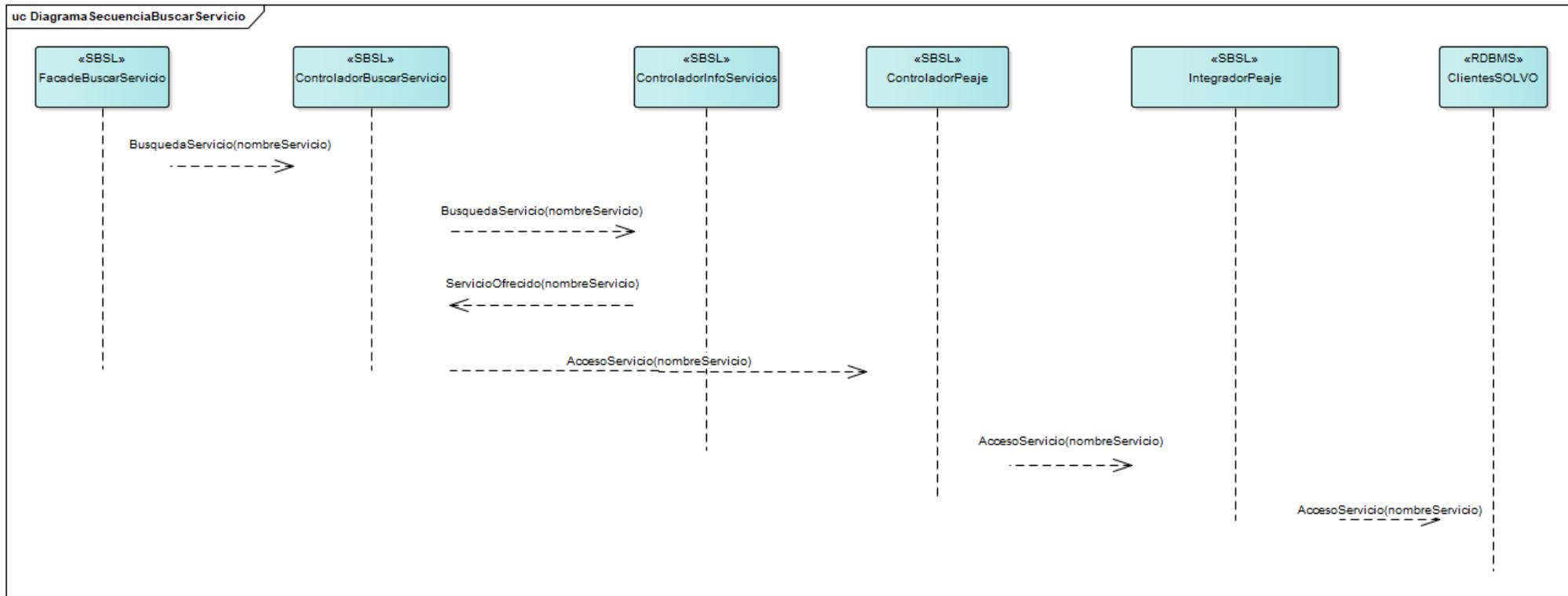
1. El conductor de transporte de carga terrestre.
2. El administrador del sistema.

Por otra parte, las entidades externas al sistema se identificaron nueve, las cuales son:

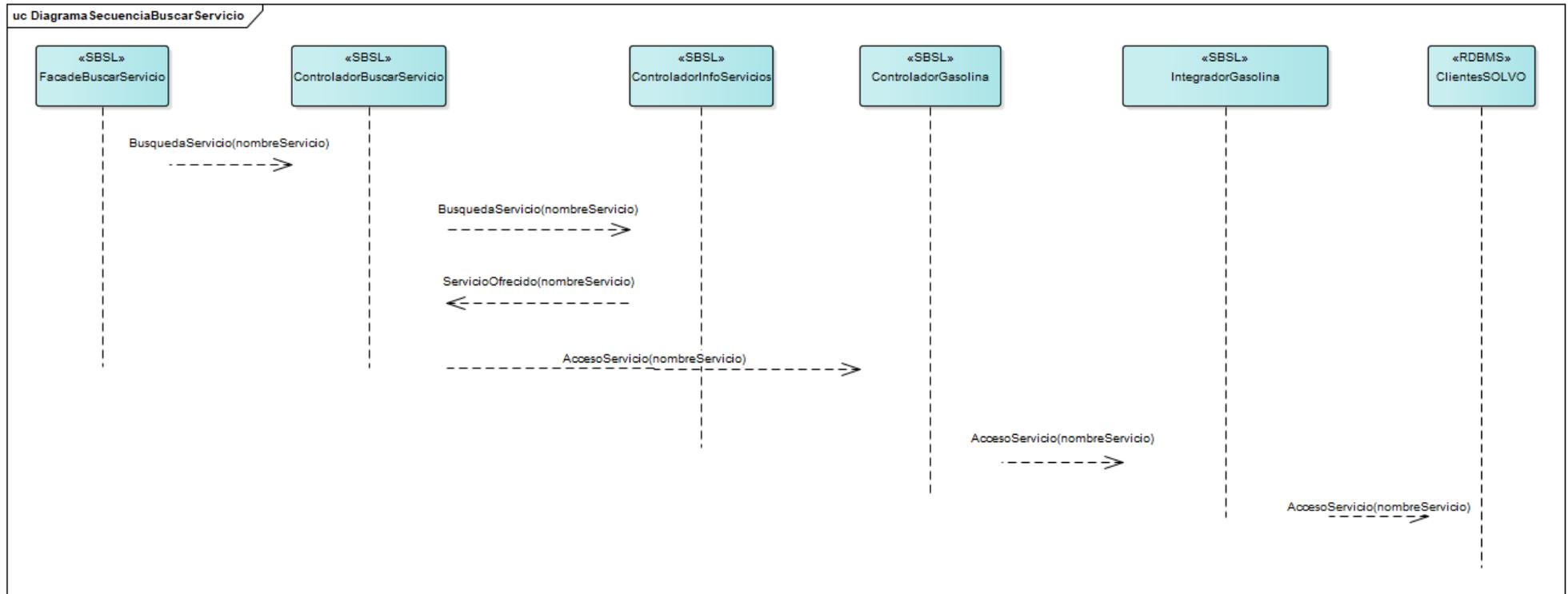
1. **Servidor:** Se albergará la lógica de negocio de SOLVO con el fin de poder escalar la aplicación a otros dispositivos en el futuro y que la lógica no quede únicamente en el dispositivo Android.
2. **Punto de Alojamiento:** Permitiendo informar al conductor sobre este tipo de establecimientos cercanos a su ubicación.
3. **Punto de Alimentación:** Permitiendo informar al conductor sobre este tipo de establecimientos cercanos a su ubicación.
4. **Taller:** Permitiendo informar al conductor sobre este tipo de establecimientos cercanos a su ubicación.
5. **Gasolinera:** Permitiendo informar al conductor sobre este tipo de establecimientos cercanos a su ubicación.
6. **Parqueadero:** Permitiendo informar al conductor sobre este tipo de establecimientos cercanos a su ubicación.
7. **Peaje:** Permitiendo informar al conductor sobre este tipo de establecimientos cercanos a su ubicación.
8. **RDBMS:** Es la base datos en la cual se guardan los datos que requiera SOLVO.
9. **GPS:** Es para la funcionalidad principal de SOLVO que es proporcionar información relevante a los conductores dependiendo de su ubicación.

## 4.2. Comportamiento e Interacción

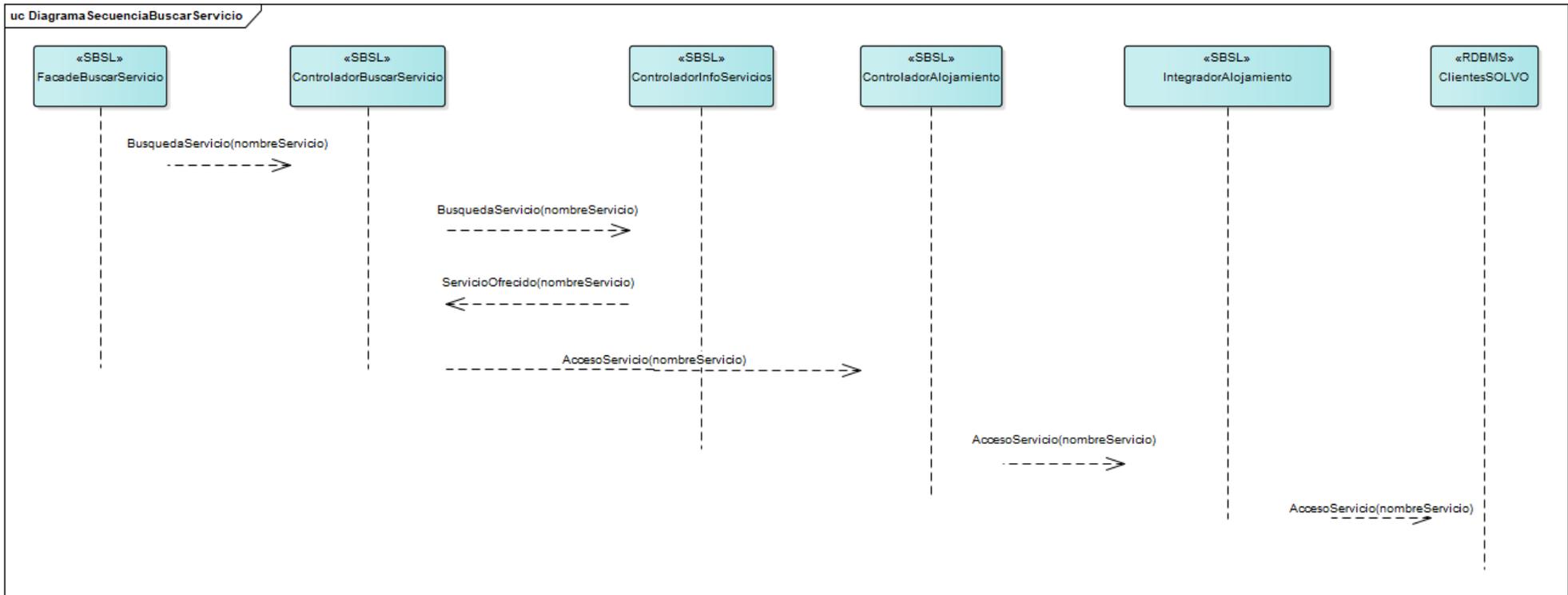
Esta sección mostrará los diagramas de secuencia de los casos de uso más relevantes en el desarrollo de la aplicación y cómo es la interacción entre los componentes que conforman cada caso de uso.



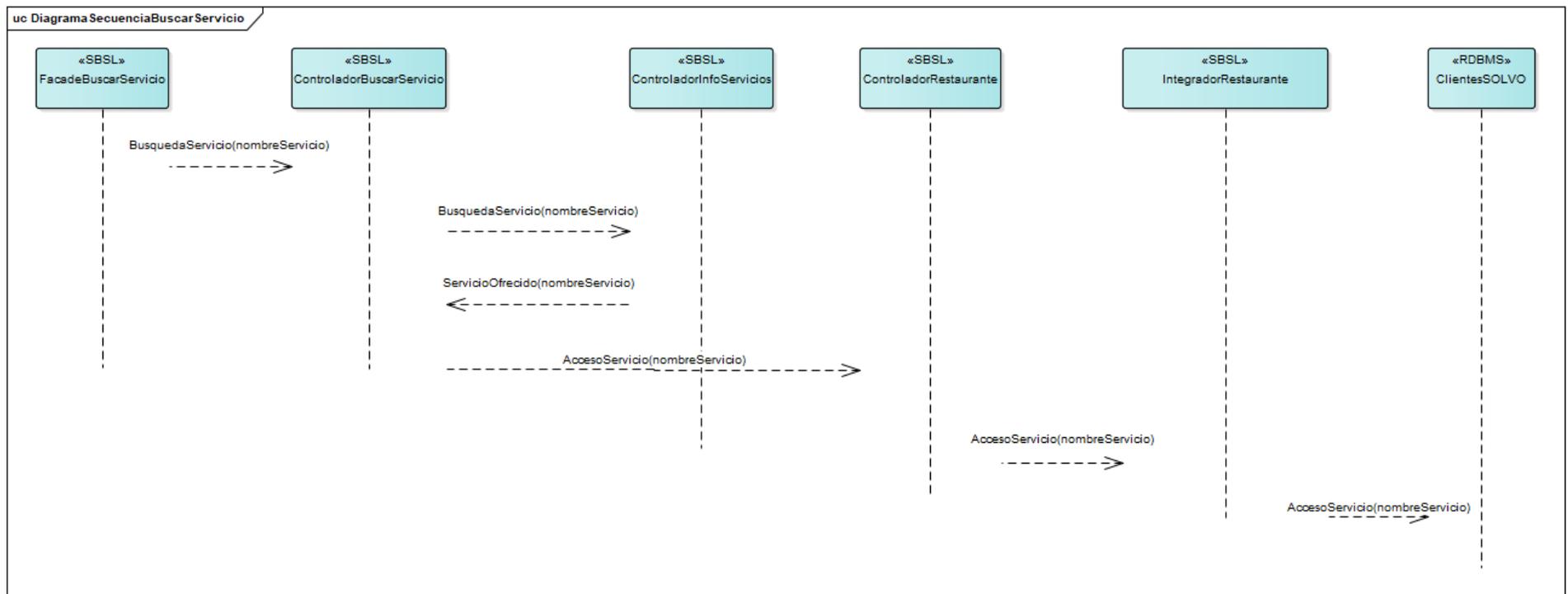
*Ilustración 29. Diagrama de Secuencia-Caso de uso Buscar Servicio Peaje. Elaborado con Enterprise Architect.*



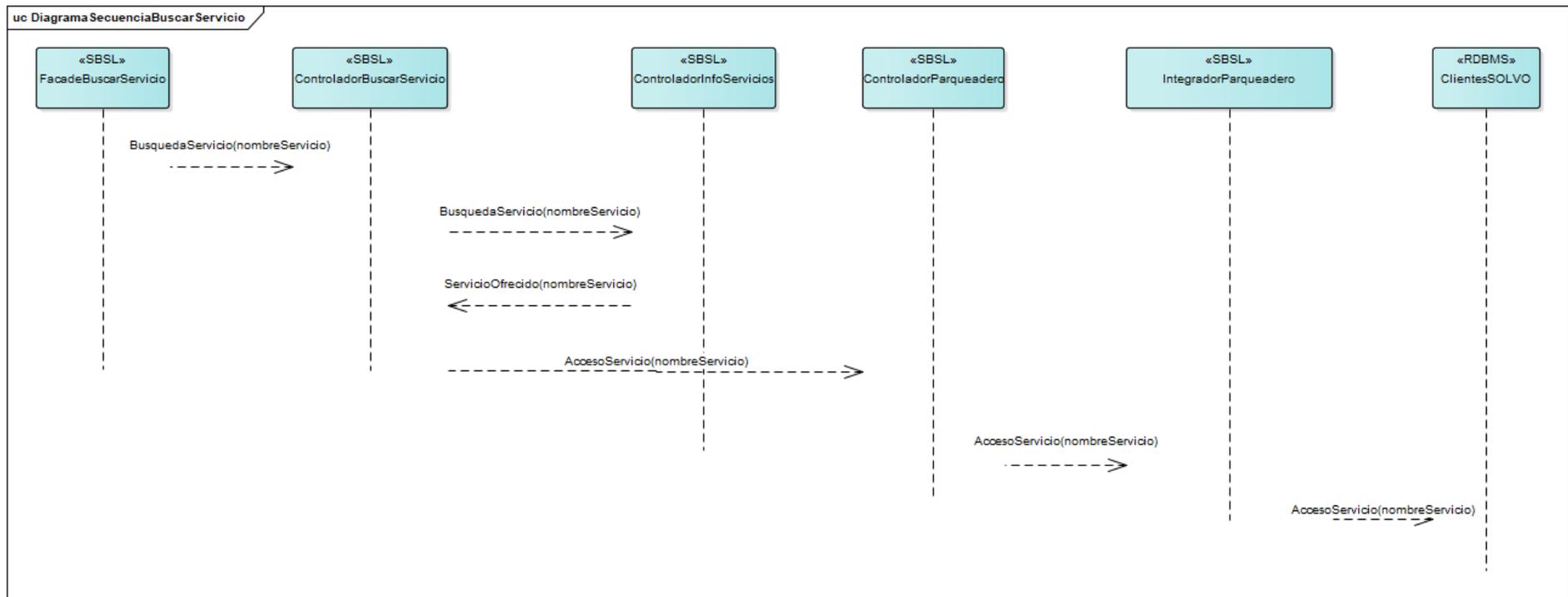
**Ilustración 30.** Diagrama de Secuencia-Caso de uso Buscar Estaciones de Servicio. Elaborado con Enterprise Architect.



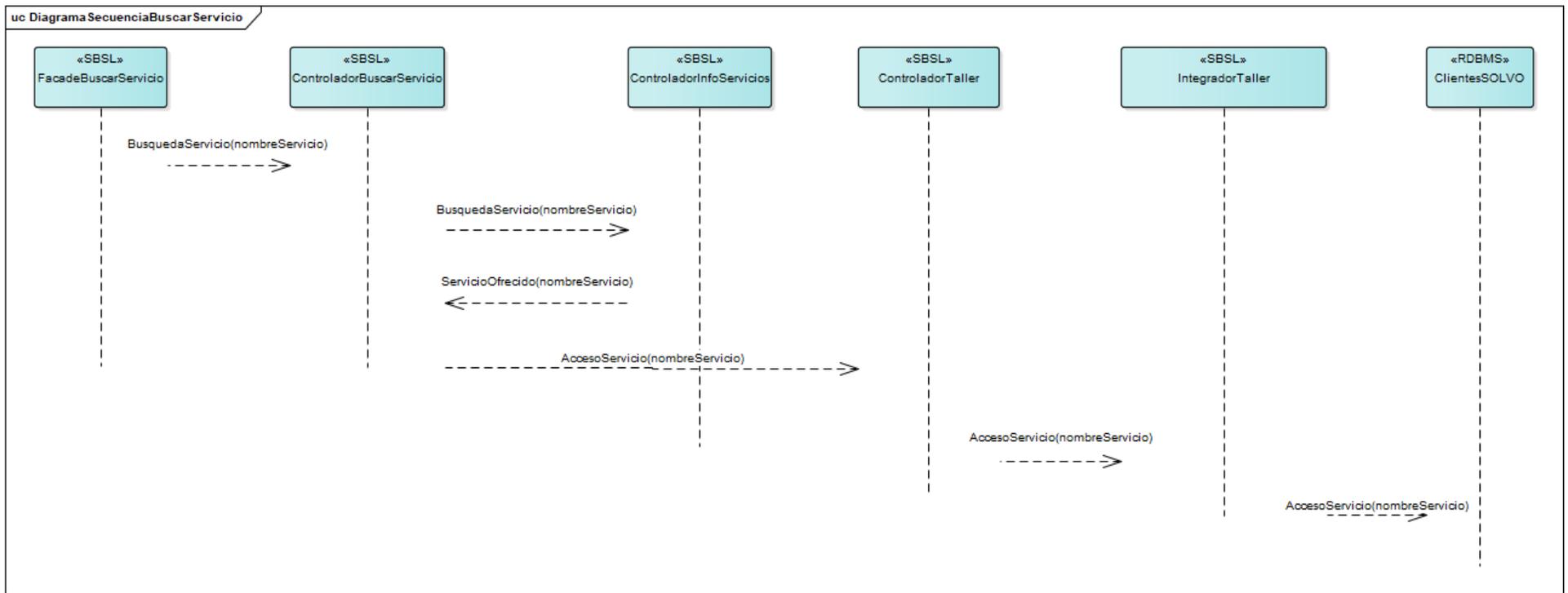
*Ilustración 31. Diagrama de Secuencia-Caso de uso Buscar Alojamiento. Elaborado con Enterprise Architect.*



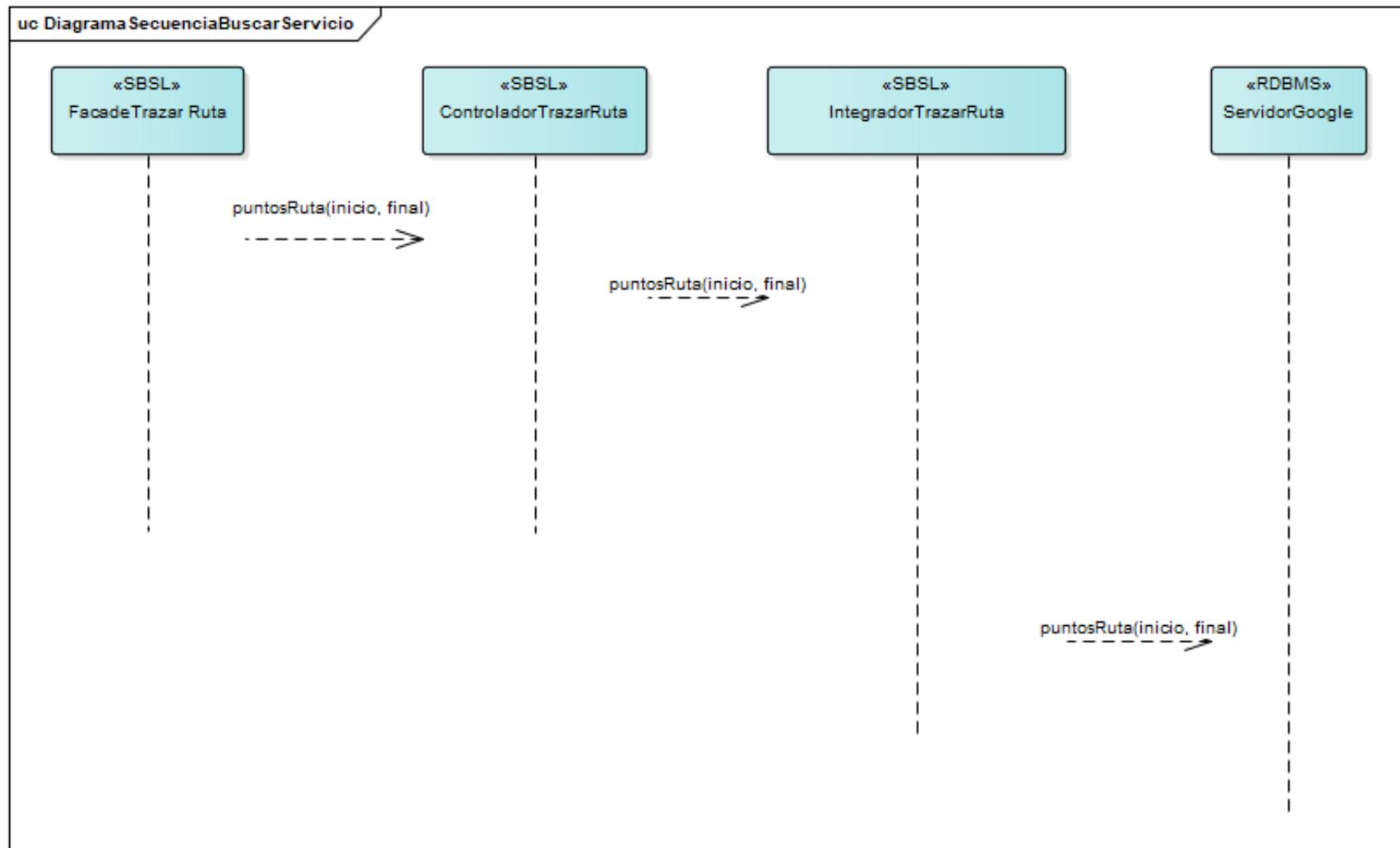
**Ilustración 32.** Diagrama de Secuencia-Caso de uso Buscar Restaurante. Elaborado con Enterprise Architect.



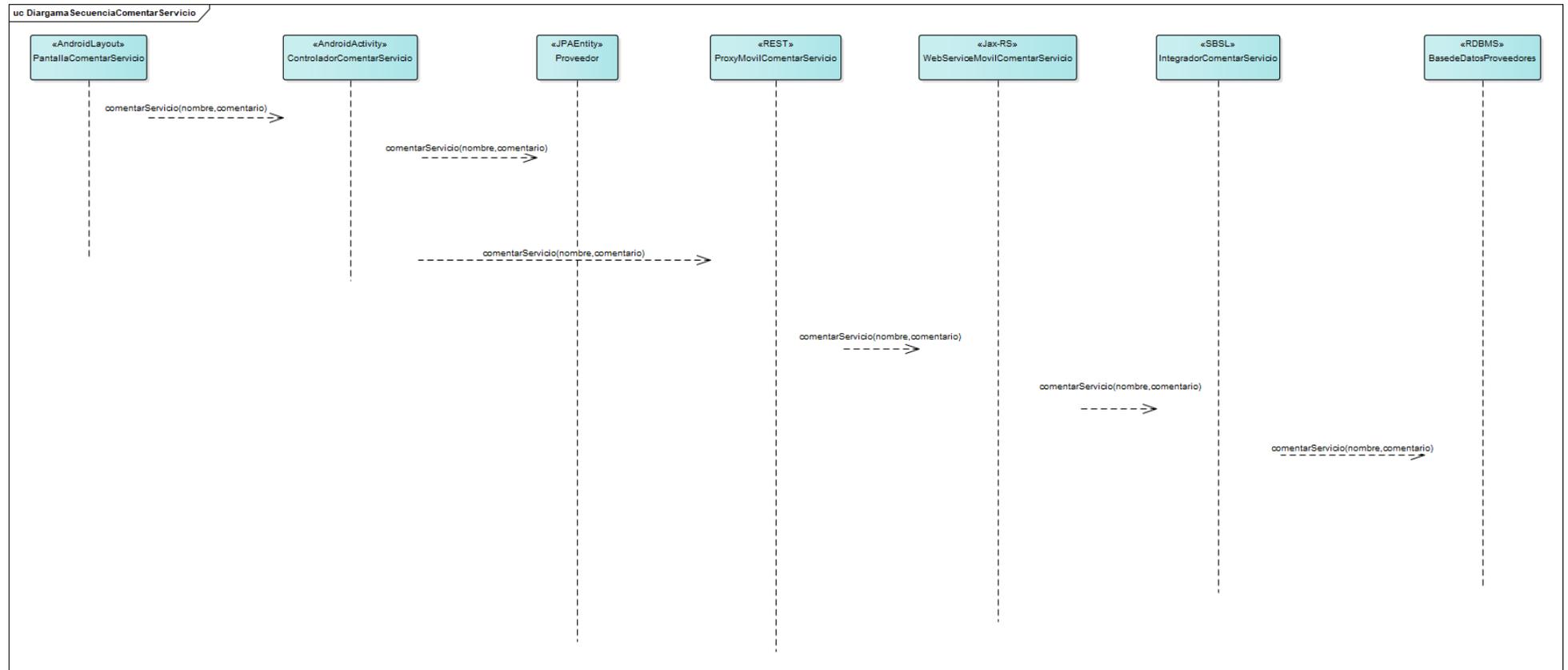
**Ilustración 33.** Diagrama de Secuencia-Caso de uso Buscar Parqueadero. Elaborado con Enterprise Architect.



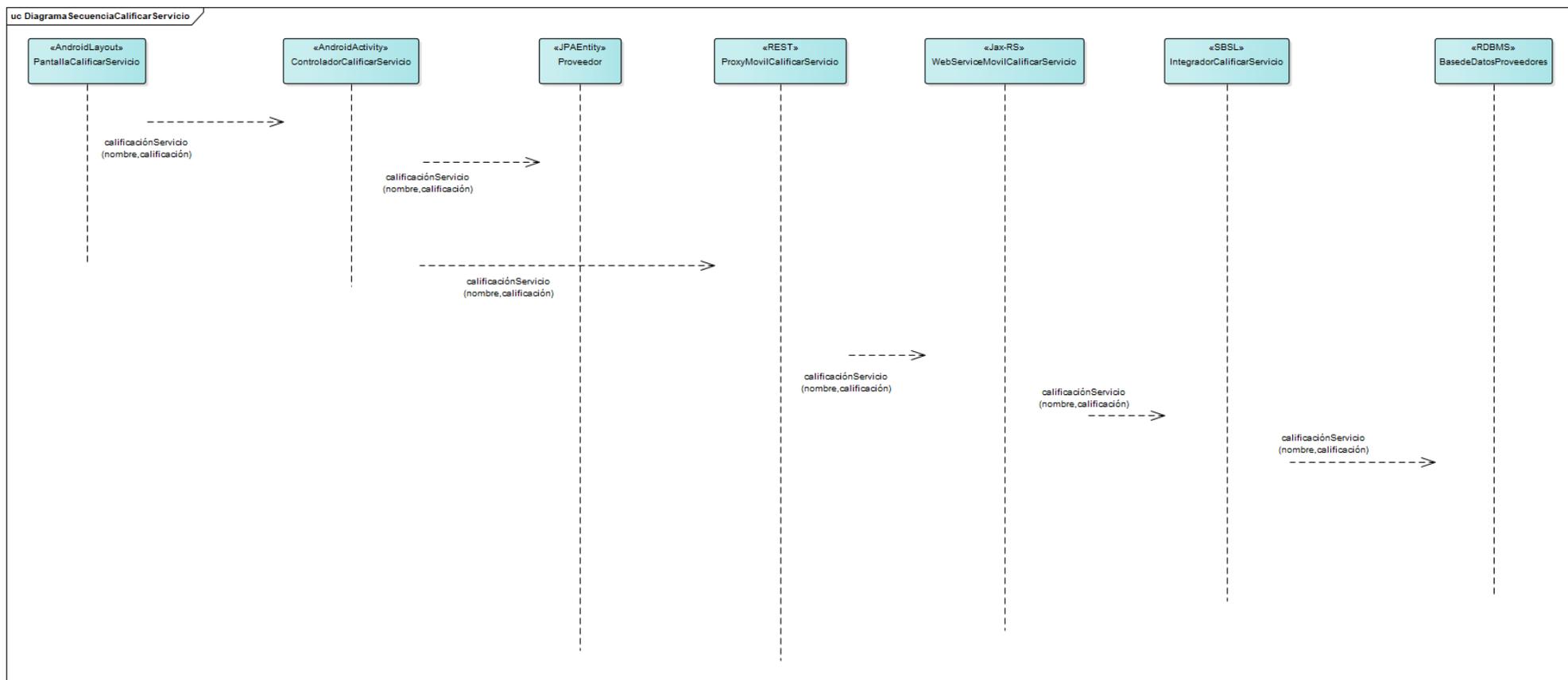
*Ilustración 34. Diagrama de Secuencia-Caso de uso Buscar Taller. Elaborado con Enterprise Architect.*



*Ilustración 35. Diagrama de Secuencia-Caso de uso Trazar Ruta. Elaborado con Enterprise Architect.*



*Ilustración 36. Diagrama de Secuencia-Caso de uso Comentar Servicio. Elaborado con Enterprise Architect.*

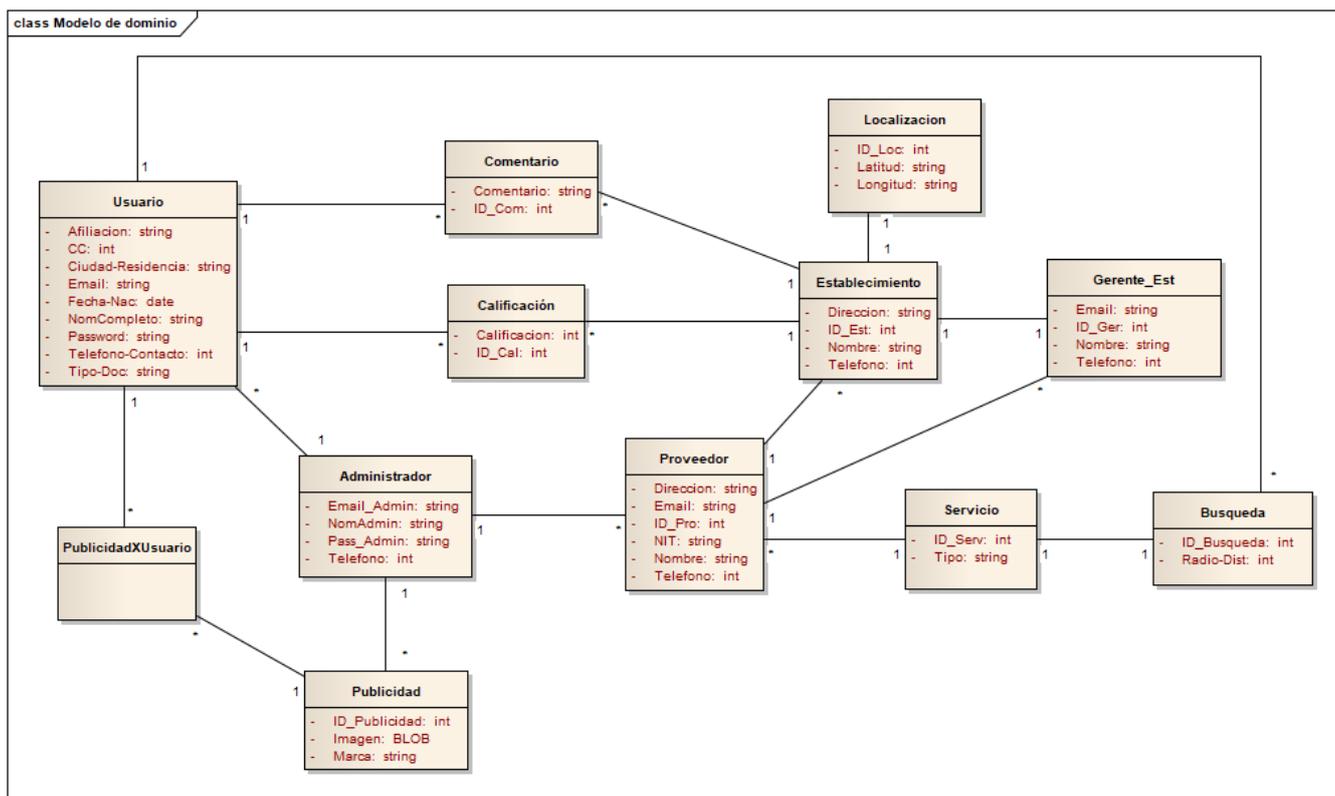


*Ilustración 37. Diagrama de Secuencia-Caso de uso Calificar Servicio. Elaborado con Enterprise Architect.*

## 5. Diseño de Bajo Nivel

### 5.1. Diagrama de Clases

A continuación, se presenta en la *ilustración 15*, el diagrama de clases de la plataforma SOLVO, el cual se compone de 12 clases, teniendo en cuenta que la clase que posee más información es el usuario; esto teniendo en cuenta el manejo de información que se realizará posteriormente cuando los conductores usen la plataforma.



*Ilustración 38. Diagrama de Clases del Sistema SOLVO. Elaborado con Enterprise Architect.*

## 6. Interfaces de la aplicación

Para este realizar el árbol de seguimiento de la aplicación, primero hay que definir las interfaces que existen y cómo se relacionan entre ellas. Las interfaces usadas, las cuales serán mostradas textual y gráficamente son:

- SOLVO App
  - Pantalla de inicio
    - Iniciar sesión
    - Registrar usuario
      - Confirmar registro
  - Pantalla principal
    - Menú de opciones
      - Redes sociales
      - Perfil del usuario
      - Puntos SOLVO
      - Cerrar sesión
    - Menú de configuraciones
      - Configuración radio de búsqueda
    - Buscar servicio
  - Pantalla de servicios
    - Calificar/Comentar servicio
    - Mostrar/Navegar ruta



*Ilustración 39. Parte "Pantallas de inicio" del árbol de seguimiento.*

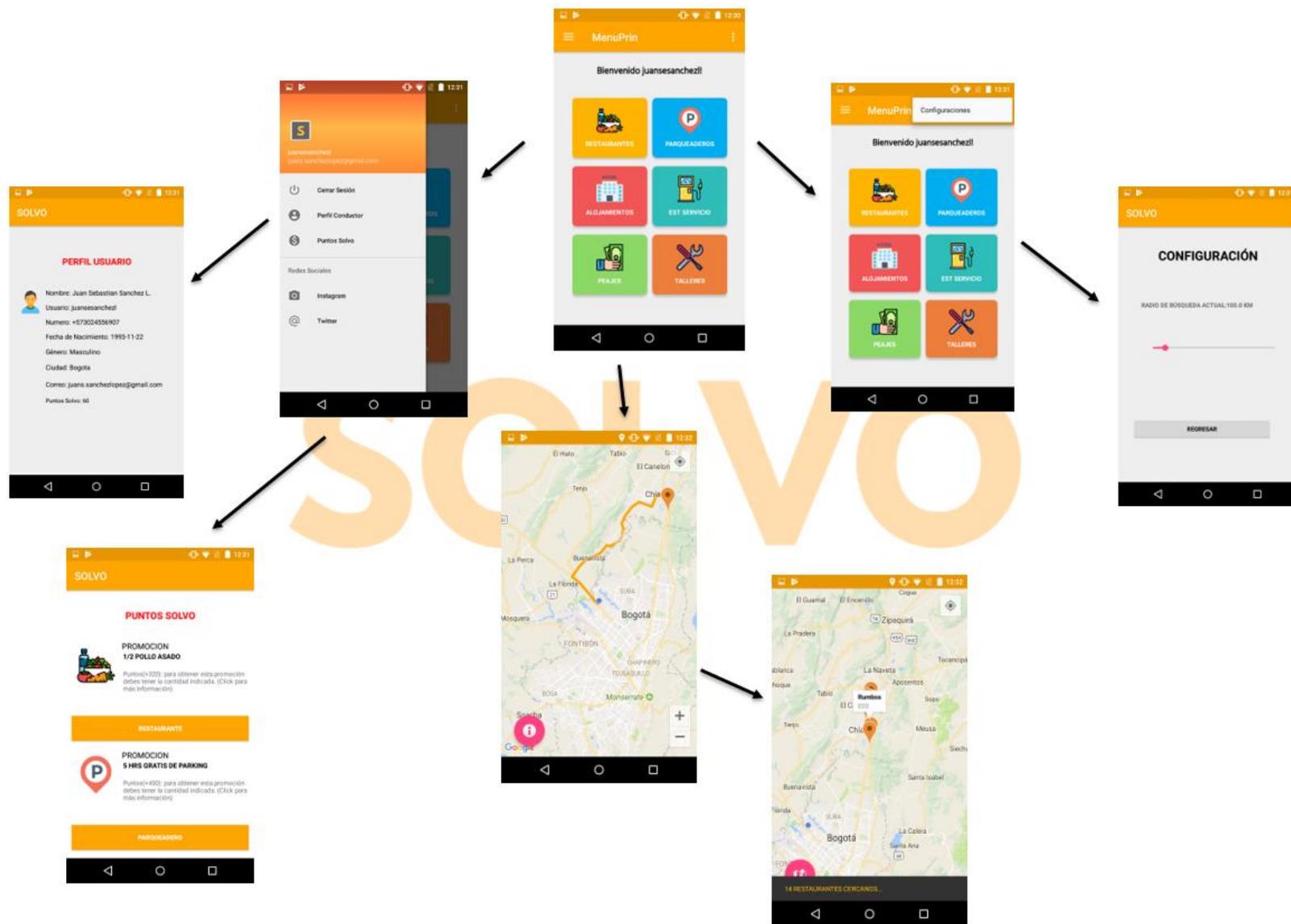
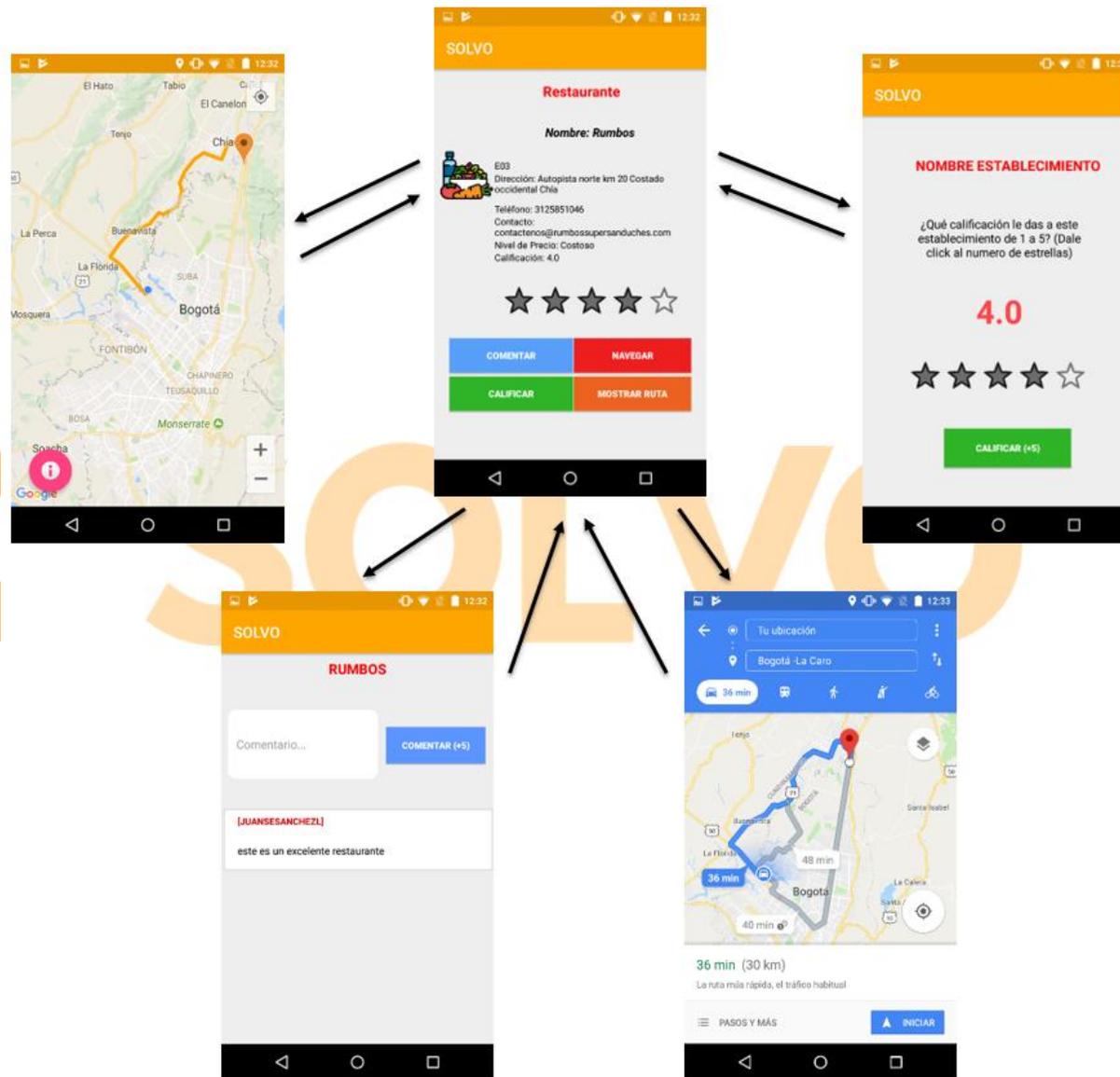


Ilustración 40. Parte "Pantalla principal" del árbol de seguimiento.



*Ilustración 41. Parte "Pantalla de servicios" del árbol de seguimiento.*

Para más detalles sobre el prototipo se puede acceder al anexo ([ver Manual de Usuario e Instalación - SOLVO](#)).

# SOLVO

## REFERENCIAS

- Agarwal, B. B., Tayal, S. P., & Gupta, M. (2009). *Software Engineering and Testing*. Jones & Bartlett Publishers.
- Albin, S. T. (2003). *The Art of Software Architecture - Design Methods and Techniques*. Wiley Publishing, Inc.
- Bass, L., Clements, R., & Kazman, R. (2003). *Software Architecture in Practice, SEI Series in Software Engineering* (Segunda ed.). Addison Wesley.
- Craig, R. D., & Jaskiel, P. S. (2002). *Systematic Software Testing*. Artech House.
- Gorton, I. (2011). *Essential Software Architecture* (Segunda ed.). Springer.
- Kruchten, P. (1995). *Architectural Blueprints-The "4+1" View Model of Software Architecture*. Recuperado el 2018, de <https://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf>.
- Microsoft. (2015). *Visual Studio*. Obtenido de Conceptos básicos de prueba unitaria: <https://msdn.microsoft.com/es-co/library/hh694602.aspx>
- Peña Melo, P. (Mayo de 2007). *Pruebas de Software*. Obtenido de [https://sophia.javeriana.edu.co/~lcdiaz/ingSw2007-1/EXP\\_PruebasSoftware\\_pPena.pdf](https://sophia.javeriana.edu.co/~lcdiaz/ingSw2007-1/EXP_PruebasSoftware_pPena.pdf)
- Piattini Velthuis, M. G. (1998). *Fundamentos y modelos de bases de datos*. Madrid, España: Alfaomega Grupo Editor.
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2006). *Database System Concepts* (Quinta ed.). Mc Graw Hill.
- Sommerville, I. (2011). *Software Engineering* (Novena ed.). Pearson Education.
- Universidad de Alicante. (s.f.). *Curso .NET con C#*. Obtenido de <https://si.ua.es/es/documentacion/c-sharp/documentos/pruebas/07pruebasunitarias.pdf>

## ANEXOS

- [Anexo BPMs SOLVO](#)
- [Vista Logica CUArq](#)
- [Anexo Documento CU ArqSig 0.2](#)
- [Anexo Documento CU SOLVO 0.7](#)
- [Anexo Vista Despliegue-SOLVO](#)
- [Anexo Vista Fisica-SOLVO](#)
- [Anexo Vista Logica-SOLVO](#)
- [Casos de Uso-SOLVO](#)
- [Diagramas de Contexto – SOLVO](#)
- [DiagramaDeClases - SOLVO](#)
- [Vista Datos - SOLVO](#)
- [Manual de Usuario - SOLVO](#)

# SOLVO