

---

# VALIDACIÓN DE LA ARQUITECTURA ATAM

---

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

---

## EQUIPO SOLVO

Camilo Andrés Oviedo Lizarazo

Juan Sebastián Sanchez 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>
30/03/2018	0.1	Inicio documentación de fase 1 del proceso de validación	Sergio Plazas
31/03/2018	0.2	Especificación de fase 1 del proceso de validación. Desarrollo de la sección Presentación de la fase 1.	Sergio Plazas
02/04/2018	0.3	Desarrollo de la sección Investigación y análisis de la fase 1.	Sergio Plazas
03/04/2018	0.4	Verificación referencias e imágenes usadas.	Sergio Plazas
08/04/2018	0.5	Desarrollo de la sección Pruebas de la Fase 2.	Sergio Plazas
18/05/2018	0.6	Corrección de estilo y adecuación final de documento	Camilo Oviedo

# TABLA DE CONTENIDO

HISTORIAL DE CAMBIOS .....	1
LISTA DE FIGURAS .....	4
LISTA DE TABLAS .....	5
1. Introducción.....	6
1.1. Descripción de la herramienta.....	6
1.1.1. Propósito .....	6
1.1.2. Abreviaciones y términos .....	6
2. Validación ATAM.....	6
2.1. Primera Fase.....	6
2.1.1. Presentación .....	6
2.1.1.1. Presentación ATAM.....	6
2.1.1.2. Presentación condiciones de negocio .....	7
Ambiente de Negocio.....	7
Limitantes técnicas.....	8
Limitantes de negocio .....	9
Atributos a evaluar .....	9
Presentación de arquitectura.....	9
2.1.1.3. Investigación y análisis.....	10
Identificación de decisiones arquitecturales.....	11
Árbol de atributos de calidad .....	11
Analizar decisiones arquitecturales .....	13
2.2. Segunda Fase.....	13
2.2.1. Pruebas .....	14
2.2.1.1. Priorización de escenarios .....	14

2.2.1.2. Analizar decisiones arquitecturales .....	15
REFERENCIAS.....	17

**SOLVO**

## LISTA DE ILUSTRACIONES

**Ilustración 1.** Árbol de atributos de calidad. Elaboración Propia..... 12

# SOLVO

## LISTA DE TABLAS

<b>Tabla 1.</b> Tabla de abreviaciones y términos. Elaboración Propia. ....	6
<b>Tabla 2.</b> Tabla de priorización de escenarios. Elaboración propia. ....	15

# SOLVO

# 1. Introducción

## 1.1. Descripción de la herramienta

### 1.1.1. Propósito

El propósito de este documento es el de registrar el proceso realizado para validar la arquitectura realizada. Este documento abarca las decisiones tomadas para realizar la validación de la arquitectura, justificando la metodología de la validación, los pasos que la metodología seleccionada requiere y la retroalimentación del proceso de validación.

### 1.1.2. Abreviaciones y términos

Abreviación	Término
ATAM	Architecture Tradeoff Analysis Method
SRS	Software Requirement Specification
AWS	Amazon Web Services

*Tabla 1. Tabla de abreviaciones y términos. Elaboración Propia.*

## 2. Validación ATAM

### 2.1. Primera Fase

La primera fase de ATAM se encarga de presentar tanto la herramienta como la arquitectura y las condiciones de negocio en las cuales se basa. Además, se presentan las decisiones arquitecturales tomadas y cómo estas se relacionan con los atributos de calidad con el fin de mostrar la arquitectura de la manera más completa posible al evaluador.

#### 2.1.1. Presentación

##### 2.1.1.1. Presentación ATAM

ATAM es un proceso de mitigación realizado en etapas tempranas del desarrollo de software. Esta metodología permite entender las consecuencias de las decisiones arquitecturales

tomadas y cómo estas decisiones se relacionan con requerimientos de atributos de calidad del sistema (*Kazman, Klein, Clemens; 2000*).

Este método facilita la detección de áreas en la arquitectura en la que existan riesgos potenciales. Además, permite hacer un análisis completo y específico de los atributos de calidad del sistema a través de etapas, mirando cómo los atributos de calidad son afectados por las decisiones arquitecturales tomadas.

Para la realización correcta de esta validación, es necesario que la arquitectura realizada sea evaluada por un comité evaluador, compuesto por personas externas que tengan conocimientos relacionados a arquitectura de software y por personas que tengan conocimiento de la arquitectura desarrollada.

Este comité está compuesto por Alejandro Sierra Múnera, profesor instructor de la asignatura Arquitectura de Software y coordinador de trabajos de grado del Departamento de Ingeniería de Sistemas de la Pontificia Universidad Javeriana; y por los estudiantes de la Pontificia Universidad Javeriana Juan Sebastián Sánchez López, programador principal de SOLVO y Sergio Enrique Plazas Merino, diseñador de la arquitectura de SOLVO.

#### **2.1.1.2. Presentación condiciones de negocio**

##### **Ambiente de Negocio**

Los conductores de carga terrestre no poseen información acerca de los servicios que usan mientras desempeñan su labor; generalmente, los conductores suplen sus necesidades básicas de manera inmediata, sin medir los gastos ni consecuencias que esto acarrea.

Debido al oficio que desarrollan, no tienen la opción de disfrutar de comodidades mientras viajan (duermen dentro del camión/debajo del camión, comen lo primero que encuentren, etc.) No existen aplicaciones que provean información a los conductores de carga terrestre en Colombia.

La solución propuesta para proveer confort a los conductores de carga terrestre es SOLVO, SOLVO es una aplicación en la cual el usuario podrá obtener información relevante acerca de servicios que puedan necesitar en sus trayectos. SOLVO tiene como funcionalidades:

- Crear una ruta para el usuario.
- Filtrar los servicios que el usuario puede necesitar basado en los servicios que el sistema ofrece información.



- Mostrar información al usuario sobre servicios a los que puede acceder en una ruta.
- Permitir al usuario calificar y opinar sobre el servicio del cual obtiene información.

La información obtenida será categorizada en informes, los cuales mostrarán estadísticas acerca de la frecuencia de uso de los servicios que el cliente fue informado.

Para que esta solución funcione, tiene que interactuar con varios actores, los cuales proveerán retroalimentación a la solución, serán quienes lo usarán y serán quienes estarán interesados con el desarrollo y uso de la aplicación. Estos están clasificados de la siguiente manera:

- **Stakeholders**

- Proveedores de servicio: Los encargados de proveer información al sistema. Proveen información relevante al cliente (nombre del proveedor, horarios de atención, servicios ofrecidos, precios)

- Empresas interesadas en la información obtenida: Empresas o proveedores de servicios que deseen obtener estadísticas de los hábitos de consumo de los clientes.

- **Usuarios**

- Conductores de carga terrestre: Usuarios principales de la aplicación. Interactúan con la aplicación dando retroalimentación y usando la aplicación.

### **Limitantes técnicas**

A continuación, se presenta la lista de limitantes técnicas principales que se tuvieron en cuenta para el desarrollo de la arquitectura de SOLVO. Estas están definidas en el documento de especificación de requerimientos (SRS):

- Usuario necesita conexión a internet por datos para realizar la sincronización con el servidor.
- SOLVO debe poder ser accedido desde cualquier celular con dispositivo Android versión 4.2 Jelly Bean en adelante.
- SOLVO debe ejecutarse en dispositivos móviles similares o superiores al Motorola ® Moto G v 1.0, ya que es un dispositivo de gama media baja.
- El usuario debe activar la ubicación en su dispositivo móvil.

## Limitantes de negocio

A continuación, se presenta la lista de limitantes de negocio que se tuvieron en cuenta para el desarrollo de la arquitectura de SOLVO:

- El administrador del sistema y de las bases de datos debe estar activo 24/7.
- Debe existir un acuerdo previo entre un proveedor y SOLVO para acceder a la información que los proveedores de servicio decidan compartir.
- Debe existir un acuerdo previo entre un proveedor y SOLVO que asegure que la información dada por los proveedores es correcta.
- Los proveedores que decidan dar información a SOLVO deben tener un computador con conexión a internet para mantener comunicación con el administrador.

## Atributos a evaluar

Los atributos a evaluar son atributos que están definidos en el estándar ISO 25010 de 2011, el cual establece que atributos de calidad se tienen en cuenta para evaluar un producto de software (*ISO 25010, 2018*). Los atributos seleccionados para ser evaluados en la arquitectura, los cuales serán definidos a continuación y especificados más adelante son:

- **Disponibilidad**
  - *Tolerancia a fallos*: La capacidad de recuperación del sistema en el momento en el que se presente un fallo y que el producto continúe funcionando independientemente de los fallos que ocurran.
  - *Recuperabilidad*: La capacidad de recuperar datos que fueron afectados en un fallo de la aplicación.
- **Mantenibilidad**
  - Siendo la característica representa la capacidad de que SOLVO pueda ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas temporales. (*ISO 25010, 2018*)

## Presentación de arquitectura

La arquitectura desarrollada fue basada en requerimientos funcionales declarados previamente en el SRS; en este documento se enuncian todos los requerimientos funcionales y no

funcionales. A continuación, se mostrará la lista de los requerimientos funcionales claves para SOLVO, categorizados por actores que interactúan con la aplicación:

- **Usuario**

- El sistema debe permitir al usuario consultar información acerca de peajes.
- El sistema debe permitir al usuario consultar información acerca de puntos de alimentación.
- El sistema debe permitir al usuario consultar información acerca de establecimientos de servicio.
- El sistema debe permitir al usuario consultar información acerca de talleres.
- El sistema debe permitir al usuario consultar información acerca de alojamientos.
- El sistema debe permitir al usuario consultar información acerca de parqueaderos.
- El sistema debe permitir al usuario realizar comentarios a servicios ejecutados
- El sistema debe permitir al usuario realizar calificaciones a servicios ejecutados
- El sistema debe desplegar información acerca de alojamientos/establecimientos de servicio/peajes/puntos de alimentación/ talleres/parqueaderos.

- **Administrador**

- El sistema debe permitir al administrador iniciar sesión
- El sistema debe otorgar al Administrador un nivel de acceso mayor para ver un listado de las cuentas que están registradas en el sistema
- El sistema debe permitir al administrador monitorear contenido en cuanto a las cuentas de los usuarios registrados, y el contenido de los establecimientos
- El sistema debe permitir al administrador eliminar cuentas de los usuarios registrados

### **2.1.1.3. Investigación y análisis**

Una vez presentada la arquitectura, podemos continuar con la especificación de la arquitectura, concretamente, cuáles y por qué decisiones arquitecturales fueron tomadas y bajo qué atributos de calidad la arquitectura va a ser evaluada.

## Identificación de decisiones arquitecturales

El desarrollo de esta arquitectura está planteado por las decisiones arquitecturales tomadas para facilitar el desarrollo de la arquitectura. La arquitectura empleada es de tipo multi-tier, donde en cada *Tier* existen capas. Esta arquitectura se representa de la siguiente manera:

- **Tier 1: Dispositivo móvil y browser**

- Capa 1 (Presentación)

- El administrador accede al servidor a través del browser de un computador.
- El usuario accede al servidor a través de un dispositivo móvil.

- **Tier 2: Servidor**

- Capa 2 (Lógica e integración)

- De ser necesario acceder a los datos del usuario, el administrador a través del servidor realiza los cambios necesarios en las base de datos.
- El cliente (ya sea el administrador o el usuario) se conectará al servidor que posee la lógica y dependiendo de las acciones que se quieran realizar conecta a la base de datos.

- **Tier 3: Bases de datos**

- Capa 3 (Bases de datos)

- Las bases de datos de proveedores y clientes contienen información relevante que será modificada de acuerdo a las necesidades que el cliente o administrador desee.

## Árbol de atributos de calidad

El árbol de atributos de calidad no solo contiene los atributos con los que se evalúa la arquitectura, sino que además este árbol posee las debilidades que fueron identificadas en el sistema que serán validadas con estos atributos.

Este árbol está representado a continuación:



**Ilustración 1.** *Árbol de atributos de calidad. Elaboración Propia.*

## **Analizar decisiones arquitecturales**

Las decisiones arquitecturales tomadas tienen implicaciones, tanto positivas como negativas en el desarrollo de SOLVO. En el diseño de la arquitectura se pudieron encontrar las siguientes características que pueden afectar la arquitectura de SOLVO.

- **Debilidades**

- Aplicación puede tardar en proveer la información.
- Dependencia de tercero para actualización de información.
- Pérdida de datos en caso de que internet o la aplicación falle.
- Consistencia eventual de la información cuando no haya conexión a internet.

- **Puntos sensibles**

- Recuperación información
- Tiempos de respuesta

Estas debilidades surgen ya que toda la lógica de SOLVO está ubicada en un servidor externo provisto por AWS. Estos servicios son accedidos por un dispositivo móvil, el cual posee las funcionalidades del cliente; mientras que las funcionalidades del administrador son accedidas por un browser de un computador.

Ya que el servidor de AWS está ubicado en otra parte del mundo, la falta de conexión al servicio de internet puede ocasionar que los datos que el usuario ingrese para realizar una búsqueda se pierdan o no se realice la búsqueda deseada.

Independientemente de que haya conexión o no, la aplicación puede cerrarse inesperadamente por fallos en el dispositivo móvil, perdiendo los datos que el cliente haya ingresado para realizar búsquedas, o calificar y comentar un servicio.

Además, la información mostrada al cliente puede estar desactualizada debido a la falta de conexión a internet, la cual permite acceso al API de *Google Maps* y a las bases de datos de SOLVO, ubicadas en el mismo servidor de AWS.

## **2.2. Segunda Fase**

Una vez realizada la primera fase, esta y la arquitectura se exponen frente al comité evaluador. La segunda fase consiste en integrar la retroalimentación recibida a la arquitectura

diseñada y realizar una segunda presentación de la arquitectura al comité con las correcciones necesarias.

Esta fase tiene dos características principales, en las cuales se presenta la priorización de las correcciones a realizar, el análisis de las correcciones realizadas y la arquitectura con las correcciones realizadas.

## 2.2.1. Pruebas

### 2.2.1.1. Priorización de escenarios

Esta priorización se realiza de acuerdo a la importancia de la aplicación de las correcciones a la arquitectura. Cada escenario con su elemento a corregir y la causa del elemento están priorizados en la siguiente tabla:

Prioridad	Escenario	Causa
1	La arquitectura no refleja los requerimientos no funcionales.	La arquitectura está diseñada para cumplir los requerimientos funcionales.
2	No existe diferenciación entre las capas de integración de servicios de cliente y funcionalidades ofrecidas por la aplicación	Causado por una confusión en los nombres de los paquetes de integración
3	Hay duda en los protocolos a usar para la comunicación entre el servidor con la lógica de negocio y el dispositivo móvil.	Los protocolos a usar aunque pueden ser usados, pueden dificultar el desarrollo.
4	No existe relación entre los atributos de calidad a evaluar y la arquitectura	Los atributos fueron declarados, pero no fueron implementados debido a que la arquitectura no refleja los requerimientos no funcionales
5	Falta especificación en la arquitectura.	Falta especificar la

	Ubicación geográfica de los nodos de la arquitectura.	interacción y ubicación de los componentes del sistema.
6	No existe una implementación clara de las bases de datos en la arquitectura	Las bases de datos estaban separadas, ignorando el hecho que pertenecen a un mismo servicio (AWS)
7	No hay limitantes de negocio claras	Limitantes de negocio se aproximan más a ser requerimientos que limitantes de negocio
8	La arquitectura desarrollada no se ha categorizado en los diferentes tipos de arquitectura	El tipo de arquitectura planteado es incorrecto (Cliente-servidor)

*Tabla 2. Tabla de priorización de escenarios. Elaboración propia.*

### **2.2.1.2. Analizar decisiones arquitecturales**

Una vez realizadas las correcciones sugeridas por el comité evaluador, se vuelve a realizar el análisis de las decisiones arquitecturales, esta vez incluyendo las posibles soluciones a los problemas identificados anteriormente:

- **Debilidades**
  - Aplicación puede tardar en proveer la información.
  - Dependencia de tercero para actualización de información.
  - Pérdida de datos en caso de que internet o la aplicación falle.
  - Consistencia eventual de la información cuando no haya conexión a internet.
- **Puntos sensibles**
  - Recuperación información
  - Tiempos de respuesta
  - Validez información



Las soluciones planteadas para combatir las debilidades y puntos sensibles de la arquitectura son:

- Adición de estructuras temporales (colas) que guarden la información al detectar no conexión.
- Logs que registren y guarden la última operación realizada antes del fallo y restaurar búsqueda al reiniciar aplicación.
- Descargar mapas de GoogleMaps y trabajar con ellos mientras se recupera la conexión a internet, aclarando que los servicios mostrados en el mapa están desactualizados.
- Tener diferentes personas responsables de la actualización de los datos dados por los proveedores.
- Mostrar la fecha y hora de la última actualización de la información cada vez que el usuario desee mirar algún servicio en el mapa.

Las soluciones planteadas mitigan la pérdida de información ya sea por fallo del dispositivo móvil o fallo de conexión al servidor. Además, garantiza que la información mostrada en caso de fallos con conexión a internet sea válida, mostrando los tiempos de actualización.

# SOLVO

## REFERENCIAS

- Kazman, Rick., Klein, Mark., & Clements, Paul. (2000). *ATAM: Method for Architecture Evaluation (CMU/SEI-2000-TR-004)*. Visitado el 3 de abril de 2018, Obtenido del sitio web “Software Engineering Institute, Carnegie Mellon University”:  
<http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=5177>
- ISO 25010. *En línea*. Visitado el 3 de abril de 2018. Disponible en:  
<http://iso25000.com/index.php/en/iso-25000-standards/iso-25010>

# SOLVO