

## ***Documento de arquitectura de software***

### ***Control del documento***

---

#### ***Proyecto***

*Sistema de información para gestionar procesos de solicitudes en la Outlander.*

#### ***Título***

*Arquitectura del sistema – [V0.0.1 al 08 de octubre de 2022]*

#### ***Generado por***

*Estudiantes de ingeniería de sistemas– [Joseph Valbuena & Johan Carrillo]*

#### ***Aprobado por***

---

## **1. Introducción**

### **1.1. Propósito**

*El siguiente documento proporciona una descripción de la arquitectura del sistema que se pretende implementar para gestionar el proceso de solicitudes para las diferentes sedes del bar "Outlander". Utilizando diferentes tipos de vistas para representar un mismo sistema y las funcionalidades que hay dentro del mismo.*

### **1.2. Alcance**

*El documento contiene la representación del proyecto (Outlander), mediante el diseño de diagramas dado el análisis previo de los requisitos del sistema para darle solución a una problemática presentada por el cliente.*

*El documento está organizado mediante tres ideas principales:*

- *Características generales del diseño.*
- *Los requisitos funcionales y no funcionales para el diseño.*
- *Los modelos y vistas que lo representan.*

*Los modelos implementados para la representación del sistema, permiten analizar el funcionamiento del sistema desde distintos puntos de vista, de forma que podemos implementar el modelo de arquitectura 4+1" de Kruchten. Gracias a esto, se puede llegar a implementar el análisis de los requisitos, la implementación del diseño de solución, así como para la especificación, construcción y despliegue del sistema en su ambiente de explotación.*

### **1.3. Usuarios interesados (StakeHolders)**

*El Documento de Arquitectura de Software (DAS), contribuye a la fácil comprensión para todos aquellos usuarios que deseen comprender el diseño y la arquitectura correspondiente al proyecto.*

## **2. Referencias**

*Para llevar a cabo la realización de este documento, se basó la información en las siguientes referencias:*

- *Scrum*
- *The 4+1 View. Krutchen.*

### **3. Definiciones, acrónimos y abreviaciones**

**DAS:** Documento de arquitectura de software.

**SIGFO:** Sistema de información para gestionar procesos de solicitudes en la empresa Outlander.

**HTTP:** Protocolo para la transferencia de hipertextos.

**TCP:** Protocolo para el control de transmisión.

**ARQUITECTURA DE SOFTWARE:** Conjunto de elementos estáticos, propios del diseño intelectual del sistema, que definen y dan forma tanto al código fuente, como al comportamiento del software en tiempo de ejecución. Naturalmente este diseño arquitectónico ha de ajustarse a las necesidades y requisitos del proyecto.

**DESCRIPCIÓN DE ARQUITECTURA:** Colección de productos de documentación.

**VISTAS:** Es una representación de un área de interés o perspectiva del sistema en alto nivel.

**TIPOS DE VISTAS:** Especificación de una convención de cómo construir y usar una vista. Deben satisfacer la capacidad de creación y análisis de una vista.

**STAKEHOLDER:** Individuo, equipo u organización con intereses relativos al sistema.

**ESCENARIO:** Especifica el comportamiento y limita el interés de un área específica del sistema para uno o varios stakeholders.

**MODULO O COMPONENTE:** Cualquier elemento estructural abstracto, visible, externo, de alto nivel, analizable, que pueda constituir una funcionalidad de la solución del sistema.

**ATRIBUTOS DE CALIDAD:** Un atributo de calidad, es una cualidad deseable de la solución, que pueda manifestarse en forma de requerimiento no funcional, que pueda ser medible, testeable y finalmente evaluable.

## **4. Framework conceptual**

### **4.1. Descripción de la arquitectura en contexto**

*El documento representa la arquitectura del software mediante la implementación de diferentes vistas relacionadas al “Modelo de vistas 4+1” de Kruchten. Cada una de estas vistas, representan el sistema desde distintos puntos de vista para los interesados dentro del proyecto. Las vistas son las siguientes:*

- *Vista lógica.*

*El bar Outlander posee 3 sedes localizadas en varias partes de la ciudad de Bogotá, esta franquicia requiere un software el cual le permita el manejo de inventario de los productos, un muy detallado control de venta y un eficiente proceso de pago de las órdenes.*

- *Vista física.*

*Este al ser un software que se implementara vía web tiene la facilidad de ser ejecutado desde cualquier equipo, ya sea un computador, un portátil, una tablet o un celular, ahora bien Outlander dispone de computadores para sus cajeros y celulares o tablets para sus meseros*

- *Vista de desarrollo.*

*Después de realizar un análisis a lo requerido por el cliente, se concluye que para el presente proyecto lo ideal a implementar por parte del Backend del aplicativo es por medio de Java, junto con SpringBoot el cual es un framework de desarrollo, por otro lado la mejor opción par Frontend es por medio de Angular y JavaScript por su balance en carga y ejecución*

- *Vista de procesos*

*Los principales procesos que requiere cubrir el aplicativo para Outlander son:*

- *Ingreso de productos al inventario*
- *Descuento de productos del inventario*
- *Selección de mesa para toma de pedido*
- *Ingreso a cuenta de la mesa para realizar el cobro de cuenta*
- *Creación de usuarios*
- *Restablecimiento de contraseña*

*Aparte existe la vista adicional que reúne los atributos de cada una de las vistas anteriores, y las plasma en una vista de escenarios. Cada una de las vistas pertenecientes a este modelo posee su representación a través de modelos realizados en la herramienta LucidChart.*

#### **4.2. Stakeholders y sus roles**





*El presente documento identifica las funciones de los interesados o mejor conocidos como stakeholders a partir de la interpretación de los casos de uso del sistema.*

- *Meseros: Descuento de inventario, registro de productos en las cuentas de las mesas y acceso a otras mesas para tomar orden*
- *Cajeros: Dar ingreso a productos al inventario, pago de productos e ingresar a la cuenta de la mesa para realizar el debido cobro*
- *Administrador: Creación de usuarios, restablecimiento de contraseña, agregar valor a los productos a vender, modificar mesas y este posee las mismas actividades que meseros y cajeros*
- *Cliente: Informar pedido y realizar pago de pedido*

#### **4.3 Usos de las descripciones de arquitectura**

*Las descripciones de arquitectura de este documento se usarán para referenciar el diseño del sistema de software de SIGFO.*

### 5. Identificación de los Stakeholders y sus responsabilidades

<b>Stakeholder</b>	<b>Descripción</b>	<b>Escenario</b>
 <i>Cliente</i>	<i>Persona que interactúa con el mesero para solicitar productos e interactúa con el cajero para realizar el pago correspondiente.</i>	<ul style="list-style-type: none"><li>● <i>Escenario de negocio.</i></li></ul>
 <i>Administrador</i>	<i>Persona encargada de gestionar cada uno de los procesos ejecutados dentro del sistema.</i>	<ul style="list-style-type: none"><li>● <i>Escenario de negocio.</i></li><li>● <i>Escenario de diseño.</i></li></ul>
 <i>Cajero</i>	<i>Persona encargada de efectuar y cobrar los pagos correspondientes de la cuenta de cada mesa.</i>	<ul style="list-style-type: none"><li>● <i>Escenario de negocio.</i></li></ul>
 <i>Mesero</i>	<i>Persona encargada de realizar la función de tomar la orden de cada cliente, entregar los productos y descontar del inventario.</i>	<ul style="list-style-type: none"><li>● <i>Escenario de negocio.</i></li></ul>

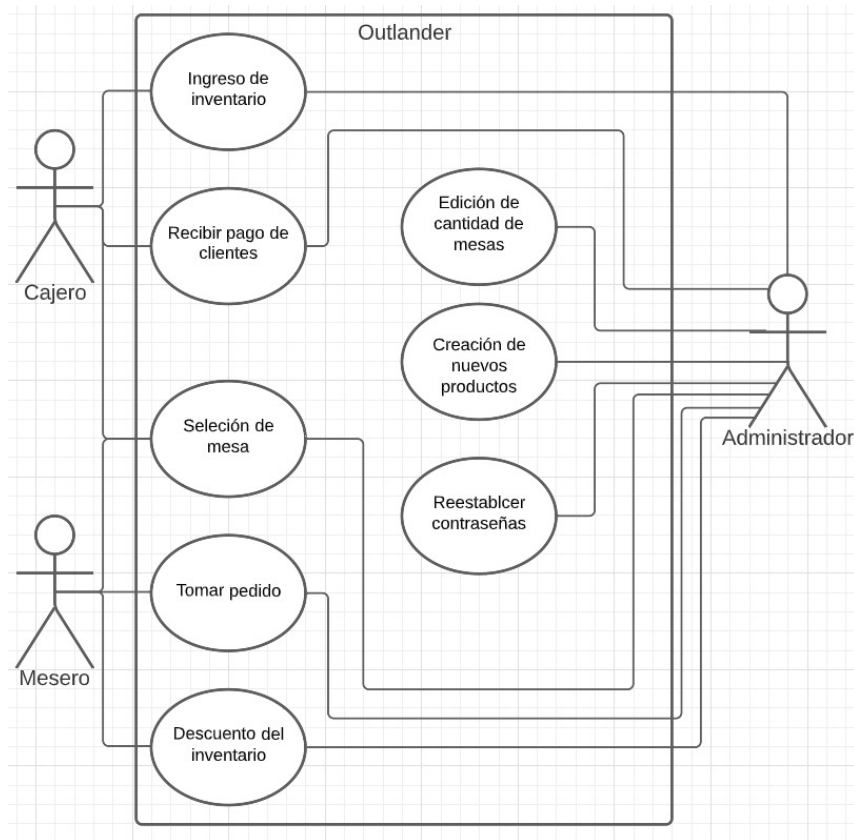
### 5.1. Selección de puntos de vista de la arquitectura

<i>Vistas</i>	<i>UML</i>
<i>Escenarios</i>	<i>Casos de uso</i>
<i>Lógica</i>	<i>Clases</i>
<i>Física</i>	<i>Paquetes</i>
<i>Procesos</i>	<i>Secuencia</i>

### 5.2. Vistas de arquitectura

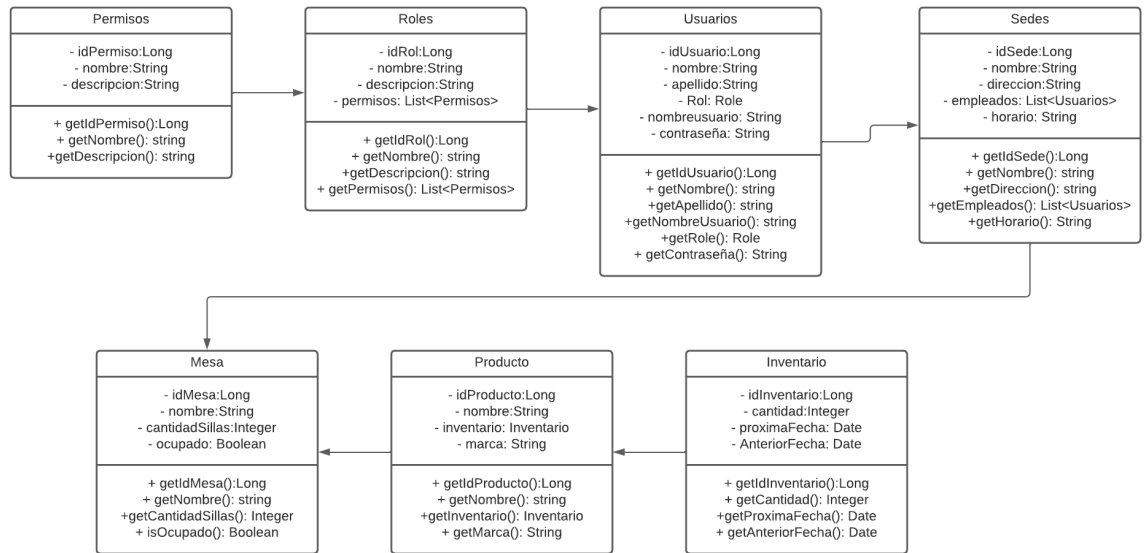
#### Vistas - Escenarios (+1)

#### Diagrama de casos de uso



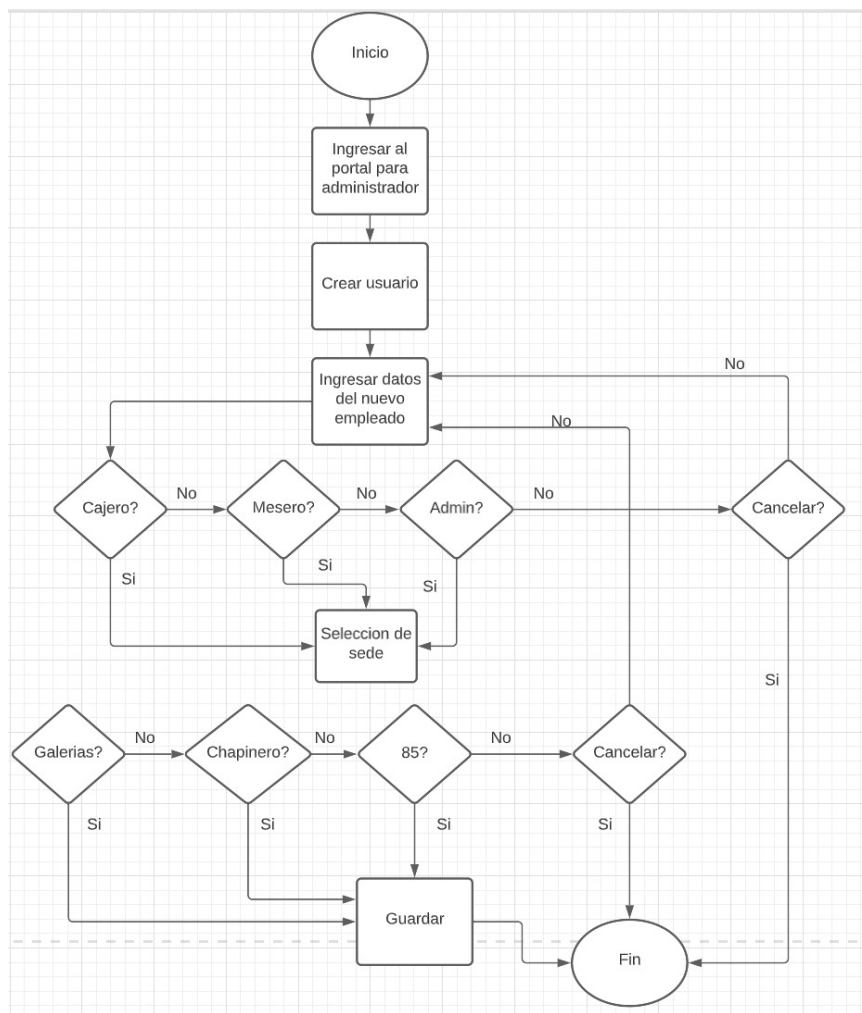
#### Vistas – Vista lógica

## Diagrama de clases

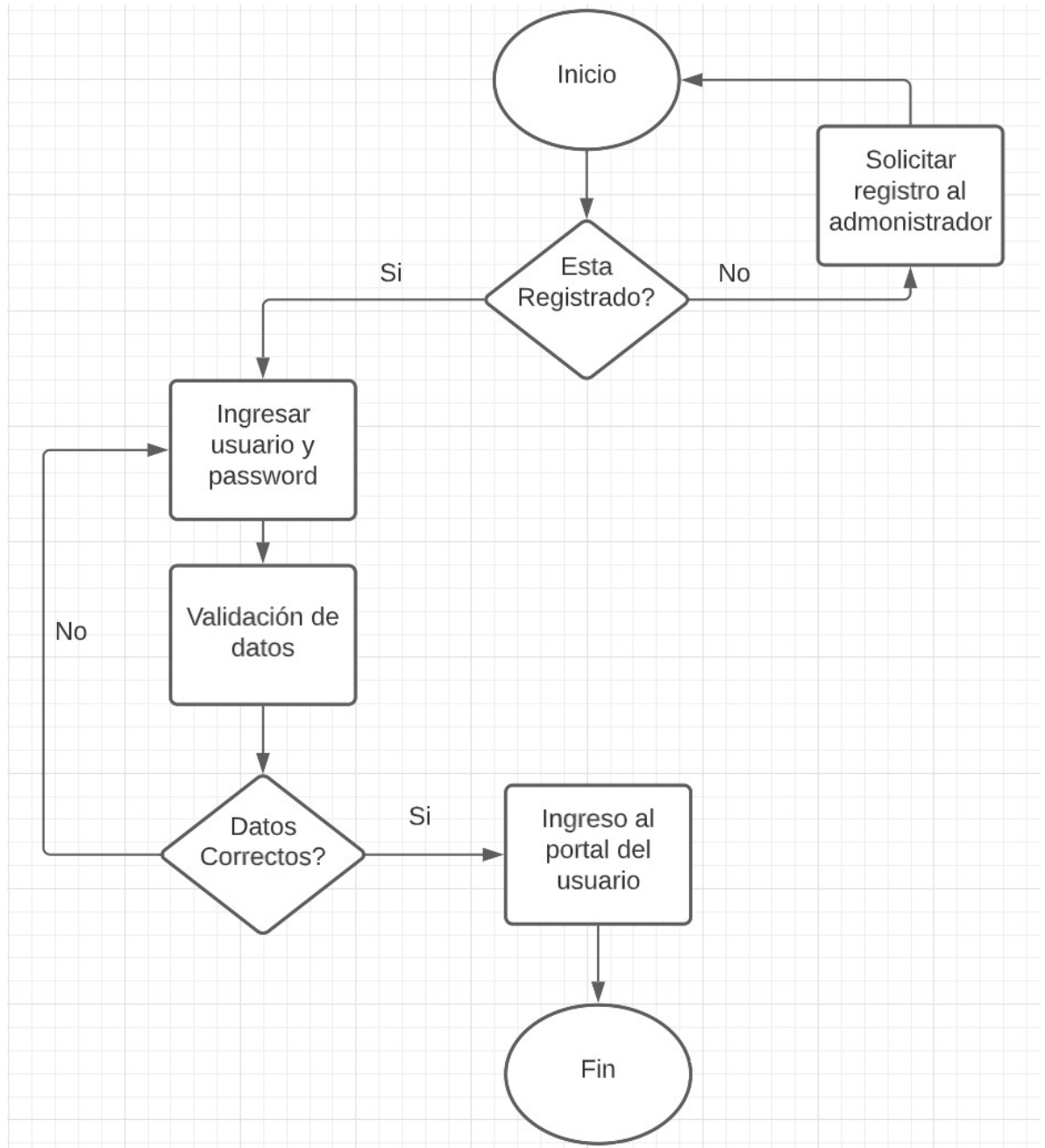




### Diagrama de secuencias para la creación de usuario

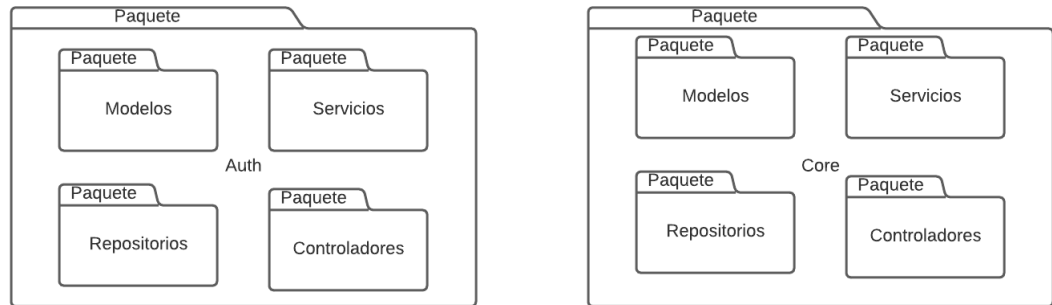


### Diagrama de secuencias para el inicio de sesión



## ***Vistas – Vista de desarrollo***

### ***Diagrama de paquetes***



## ***DESCRIPCION DE CONECTORES***

### ***5.3 Arquitectura lógica.***

#### ***Performances***

*La arquitectura de software escogida apoya a los requerimientos no funcionales y requerimientos de arquitectura de sistemas descritos a continuación:*

- 1. El sistema apoyará hasta 100 usuarios simultáneos contra la base de datos central en cualquier tiempo dado, y hasta 50 usuarios simultáneos contra los servidores locales en un momento dado.*
- 2. El sistema proporcionará el acceso a la base de datos sin más que una 10 segunda latencia.*
- 3. El sistema debe ser capaz de completar el 80 % de todas las transacciones dentro de 2 minutos.*
- 4. La parte de cliente requerirá el espacio de disco de menos de 20 MB y la RAM de 32 MB.*

## ***Calidad***

*La arquitectura de software apoya las exigencias de calidad, como estipulado a continuación.*

- 1. El interfaz de usuario será WEB.*
- 2. El interfaz de usuario del Sistema SIGFO será diseñado para la facilidad de uso y será apropiado para asegurar las normas de usabilidad universal establecidas.*
- 3. Cada despliegue de opciones de pantalla, tendrá el manual de usuario que acompañará y tendrá la información de cómo utilizar el sistema*