

## REQUISITOS FUNCIONALES

1. El administrador de cada restaurante registra los diferentes platos del menú de la semana.
2. El usuario ingresa a la aplicación y visualiza los distintos restaurantes de la ciudad, su ubicación, dirección y tipos de almuerzos.

## HISTORIAS DE USUARIO

Frente de Historia de Usuario	
Número: 1	Usuario: Administrador Restaurante
Nombre Historia: Registrar menú semanal	
Descripción: Como administrador de un restaurante puedo registrar platos al menú semanal para que mis clientes puedan estar al tanto de nuestra oferta.	
Criterios de Aceptación: Dado a que quiero registrar el menú semanal de mi restaurante, cuando yo ingrese cada uno de los platos que conforman mi menú, entonces podré observar de que el menú semanal ya se ha ingresado.	

Reverso de Historia de Usuario	
Número: 1	Usuario: Administrador Restaurante
Nombre Historia: Registrar menú semanal	
Detalles: El registro de un plato para el menú semanal deberá solicitar Nombre del plato, Precio del Plato y descripción del mismo. A medida que se ingrese un plato al menú, esta acción debe verse reflejada de manera casi inmediata en la vista del menú. Previo al registro de un plato, se deben validar de que no se ingresen datos no coherentes con lo que se está solicitando.	

Historia de Usuario	
<b>Número: 2</b>	<b>Usuario:</b> Cliente
<b>Nombre Historia:</b> Visualizar restaurantes	
<b>Descripción:</b> Como comprador puedo visualizar los diferentes restaurantes de mi ciudad con su dirección y teléfono para poder ver sus platos y así compararlos con otros restaurantes.	
<b>Criterios de Aceptación:</b> Dado a que quiero ver los restaurantes de mi ciudad, cuando ingrese a la aplicación de domicilios, entonces podré observar el listado de los restaurantes disponibles.	

Reverso de Historia de Usuario	
<b>Número: 2</b>	<b>Usuario:</b> Comprador
<b>Nombre Historia:</b> Visualizar restaurantes	
<b>Detalles:</b> La visualización de los restaurantes, se hará de forma vertical obteniendo su nombre, dirección y teléfono. Se crean unos vínculos encima de la información de cada restaurante, que llevará al comprador a otra ventana para que vea su menú.	

Historia de Usuario	
<b>Número: 3</b>	<b>Usuario:</b> Comprador
<b>Nombre Historia:</b> Visualizar menú semanal	
<b>Descripción:</b> Como comprador puedo visualizar el menú semanal del restaurante que seleccione para saber la información de los platos.	
<b>Criterios de Aceptación:</b> Dado a que quiero ver el menú semanal del restaurante que seleccione, cuando seleccione el restaurante, entonces podré observar el menú semanal con cada uno de sus platos.	

Reverso de Historia de Usuario	
<b>Número: 3</b>	<b>Usuario:</b> Comprador
<b>Nombre Historia:</b> Visualizar menú semanal	
<b>Detalles:</b> La visualización del menú semanal, se hará de forma vertical obteniendo el nombre del plato, el precio y su descripción.	

## REQUISITOS NO FUNCIONALES

### ESCENARIOS DE ATRIBUTOS DE CALIDAD

<b>Descripción</b> Un administrador de restaurante registra sus platos para su menú semanal, pero al momento de ingresar los datos del plato, comete un error al ingresar un valor o valores que no corresponden a los tipos de datos solicitados, entonces el sistema no dejará que se registre ese plato.	
Atributo: Usabilidad/Protección ante errores que cometa el usuario	
<b>Validación del Escenario</b>	
<b>Origen del estímulo</b>	Administrador de Restaurante
<b>Estímulo</b>	Ingresar platos
<b>Entorno</b>	Registrar menú semanal
<b>Artefacto</b>	BDD
<b>Respuesta</b>	Mostrar una pantalla de error al ingresar valores.
<b>Medida de respuesta</b>	Menos de 2 segundos

<b>Descripción</b> Un administrador de restaurante registra sus platos para su menú semanal, una vez suministrado los datos del plato, entonces cuando presione agregar se debe reflejar en la ventana del menú de sus restaurante.	
Atributo: Desempeño/Tiempo de Respuesta	
<b>Validación del Escenario</b>	
<b>Origen del estímulo</b>	Administrador de Restaurante
<b>Estímulo</b>	Ingresar platos
<b>Entorno</b>	Registrar menú semanal
<b>Artefacto</b>	Formulario Menú restaurante
<b>Respuesta</b>	Mostrar plato agregado en el formulario menú restaurante
<b>Medida de respuesta</b>	Menos de 2 segundos

<b>Descripción</b>	
Se desea añadir un nuevo requisito que satisface una funcionalidad de un comprador o administrador, al momento de realizar la implementación no debería afectar la implementación de los otros requisitos.	
Atributo: Mantenibilidad/Capacidad de ser cambiado	
<b>Validación del Escenario</b>	
<b>Origen del estímulo</b>	Desarrollador
<b>Estímulo</b>	Implementación nuevo requisito
<b>Entorno</b>	Depende de la implementación
<b>Artefacto</b>	Depende de la implementación
<b>Respuesta</b>	Se añade la nueva implementación sin afectar a otras.
<b>Medida de respuesta</b>	Se estima en base a la complejidad del requisito que se desea implementar

## TIPO DE APLICACIÓN

El tipo de aplicación óptima para esta idea de negocio es una aplicación de tipo Móvil híbrida porque brinda una usabilidad y portabilidad mayor al usuario.

Sin embargo, el tipo de aplicación implementada en esta iteración fue de escritorio. Lo anterior porque aún no tenemos los suficientes conocimientos para desarrollar aplicaciones Móviles y se espera que en las próximas iteraciones se pueda implementar.

## ESTILO ARQUITECTÓNICO

La arquitectura elegida fue de Cliente-Servidor, con acompañamiento del estilo arquitectónico en capas. El estilo arquitectónico en capas nos brinda un desacoplamiento entre las partes que componen nuestro sistema de domicilios, por esta razón al momento de agregar o modificar algún componente se facilitará dicho trabajo gracias a la modularidad del proyecto, evitando errores de funcionalidad en otras partes del sistema. Además se hizo una separación de roles (Administrador Restaurante - Cliente) permitiéndonos dividir la lógica entre estos dos actores, también se hizo uso del patrón MVC (ControladorCliente - ControladorAdministrador), el cual permite tener vistas de usuario desacopladas de la lógica de acceso para maximizar su mantenibilidad y seguridad.

## TECNOLOGÍAS

**Tecnología Multi-Thread y Sockets:** Esto nos permite implementar la arquitectura cliente-servidor, además permite recibir múltiples peticiones sin que el servidor colapse.

**API Swing Java:** Esta tecnología nos permite crear las interfaces gráficas de una app de escritorio para mostrarle el funcionamiento de la aplicación al usuario. Además satisface temas de Usabilidad del usuario, permitiendo la interacción con el sistema de manera sencilla.

**MySQL SGBD:** La implementación de esta tecnología permite a la aplicación, mantener un sistema de información en el tiempo, permitiéndonos realizar consultas y registrar datos requeridos por la aplicación.