

DETALLA LOS REQUISITOS SEGÚN LOS OBJETIVOS QUE SE INTENTAN ALCANZAR.

1. IDENTIFICACIÓN

Nombre de Proyecto: LaSalle FOODS	
Fecha en que se solicita: 08 de agosto del 2013	
Nombre y puesto del solicitante:	Propietarios de las tiendas de comida o servicios dentro de la Universidad.
Integrantes del equipo: Marcos Ariciaga Valdez, David Fonseca Marín, Edgar Leonel castro Méndez, Cesar Alberto carrillo Páez, Claudio castro Domínguez	Fecha acordada de término: 08 de diciembre del 2023

2. ANALISIS

2.1 ESTUDIO DE FACTIBILIDAD

2.1.1 ECONOMICA:

COSTOS INICIALES DE DESARROLLO:

El software al inicio no va a tener algún costo ya que va a ser desarrollado por nosotros, tal vez lleguemos a tener que, si nos ayuda alguien para el diseño y su mejora, pero en este inicio de la aplicación no.

ANALISIS DEL MERCADO: Por suerte no hay otra aplicación en la universidad que cuente con esta aplicación, por lo que sería la primera aplicación implementada, entonces no hay competencia de mercado que nos preocupe por el momento.

- El único costo que podríamos tener sería el sí contratamos para la entrega de los productos.

2.1.2 TÉCNICA:

DESARROLLO DE LA APLICACIÓN: Este software va a ser básico pero intuitivo para la facilitación de los usuarios y de los vendedores y va a cumplir su funcionamiento específico que es el ayudar a los alumnos a ahorrar su tiempo en la universidad.

INTEGRACIÓN DE PAGOS Y PROCESAMIENTOS

DE PEDIDOS: Esta parte va a ser sencilla esperamos que las tiendas vayan a proporcionar alguna cuenta de pago, donde ellos puedan verificar si la persona ha realizado el pago y proceder con el envío, también vamos a tener la opción de efectivo donde se le pagara al repartidos cuando haya llegado a dicho edificio.

PRUEBAS Y CONTROL DE CALIDAD: La forma de ver si nuestro software tiene fallas es agregando un apartado

donde los usuarios puedan dar sus quejas y recomendaciones, en si nuestros “Testers” van a ser los usuarios, obviamente primero haciendo pruebas internas para que no haya muchos errores y ya después con el apartado de quejas ir solucionando más problemas.

2.1.3 OPERACIONAL

PROCESOS DE OPERACIÓN: El cómo funcionara será que las tiendas que estén en el campus, la que quiera ser incluida va a tener la parte de venta en la cual van a agregar su logo, nombre y además todo lo que tiene a la venta para que la otra parte de lo app sea vista por el usuario y así escoger lo que quiere comprar, en la parte de la entrega esperamos que los que realicen la entrega sean las mismas tiendas para así hacerlo más eficiente.

PROGRAMACIÓN DE OPERACIONES:

La aplicación estará abierta durante todo el horario de la universidad tanto matutino como verpertino y también el horario que tenga establecido dichas tiendas donde se colocará si está disponible o no.

ATENCION AL CLIENTE: La atención que le estaremos brindando como ya lo dijimos anteriormente es que va a ver un apartado de quejas y recomendaciones en las cuales vamos a estar muy al pendiente para ver que podemos corregir e implementar.

2.2 ANTECEDENTES *El tiempo es un recurso fundamental para todos, pero no todos lo gestionan adecuadamente. Esto se vuelve especialmente desafiante para los estudiantes debido a sus horarios de clases, entregas de trabajos y proyectos, lo que puede dificultar tareas simples.*

La iniciativa surgió porque uno de los miembros del equipo siente que no dispone de suficiente tiempo. Por las mañanas trabaja y por las tardes asiste a clases.

2.3 OBJETIVO *Este proyecto tiene como objetivo principal ser una plataforma eficaz y práctica tanto para proveedores de alimentos como para personal universitario y estudiantes, este software no es una simple app de delivery, ahorra tiempo para los estudiantes y profesores entre clases e incluso fomenta una mayor demanda económica para la Universidad De La Salle Bajío.*

- Rentabilidad > Eficiencia operativa

- Nuevos productos y servicio modelos de negocio (produce, entrega y vende).
- Buenas relaciones con clientes y proveedores
- Mejores tomas de decisiones

Los sistemas de información (SI) tienen un impacto significativo en la forma en que operan y prosperan los negocios. Algunos de estos beneficios pueden ser:

- **Automatización de Procesos:** Los SI automatizan tareas y procesos rutinarios. Esto reduce la carga de trabajo manual y permite a los empleados centrarse en actividades más estratégicas y creativas.
- **Toma de Decisiones Informada:** Proporcionan acceso a información actualizada y relevante, permitiendo a los líderes y gerentes tomar decisiones basadas en datos concretos en lugar de depender de conjeturas o intuiciones.
- **Acceso Remoto y Flexibilidad Laboral:** Los SI permiten el acceso a la información y herramientas de trabajo desde cualquier lugar con conexión a Internet, lo que facilita el trabajo remoto y horarios flexibles.
- **Optimización de la Experiencia del Cliente:** Los SI pueden recopilar y analizar datos sobre el comportamiento de los clientes, lo que permite a las empresas ofrecer experiencias personalizadas y adaptar sus productos o servicios para satisfacer las necesidades y preferencias de los clientes.
- **Innovación y Desarrollo de Nuevos Productos o Servicios:** Los SI pueden impulsar la innovación al proporcionar herramientas para la investigación y desarrollo, así como para el diseño y pruebas de productos o servicios.
- **Creación de Ventajas Competitivas:** Un uso efectivo de los SI puede diferenciar a una empresa de sus competidores al ofrecer un servicio más eficiente, personalizado y centrado en el cliente.

2.4 ALCANCE: *El alcance del proyecto es en un área centralizada y reducida, pues esto solo estará disponible para uso exclusivo de los alumnos de la universidad de La Salle Bajío, Campus Campestre.*

2.5 SUPOSICIONES: *Cada comercio quiera un diseño específico, dar tickets en físico, dar facturas en físico, solo aceptar pagos en efectivo o de*

forma electrónica, asumir que todos los platillos y comercios serán calificados.

2.6 RESTRICCIONES: *En cuanto a restricciones pueden ser: Limitaciones en dar de alta los comercios por políticas de la Universidad de La Salle Bajío, el no poder agregar todos los productos, el que los comercios no quieran adaptarse o migrar al uso de una aplicación.*

2.7 REQUERIMIENTOS FUNCIONALES

Núm. Requerimientos	Descripción
RF001: Registro de Usuarios	Los usuarios tendrán que crear cuentas en la aplicación, proporcionando la siguiente información: <ul style="list-style-type: none"> • Nombre • Correo electrónico (@lasallebajio.edu.mx) • Contraseña • Matrícula
RF002: Inicio de Sesión	Los usuarios registrados deberán iniciar sesión en la aplicación utilizando su matrícula y contraseña previamente registrado
RF003: Explorar Menú	Los usuarios podrán navegar y explorar el menú del establecimiento el cual va a estar organizado en las siguientes categorías: <ul style="list-style-type: none"> • Entradas • Platos principales • Postres
RF004: Agregar y Eliminar Ítems	Los usuarios tendrán un carrito de compras en el cual podrán agregar o eliminar platillos para su compra
RF005: Personalización de ítems	Para algunos productos, los usuarios tienen que realizar personalizaciones, como elegir ingredientes.
RF006: Gestión de Pedidos	Los usuarios tienen que revisar y confirmar los pedidos antes de finalizarlos, así como poder ver un resumen detallado del pedido, incluido el costo total.
RF007: Selección de Establecimiento	Los establecimientos pueden elegir si reciben o no pedidos, los usuarios tienen que elegir entre los establecimientos que están disponibles para realizar sus pedidos
RF008: Pago y Confirmación	Los usuarios realizarán el pago a través de la aplicación utilizando métodos de pago seguros, en efectivo, tarjetas de crédito o débito.
RF009: Notificaciones	La aplicación debe proporcionar notificaciones en tiempo real para informar sobre el estado de los pedidos, la confirmación del pedido, tiempo estimado de preparación y cuando se encuentre listo para su recolección.
RF010: Historial de Pedidos	Los usuarios tendrán acceso a un historial de sus pedidos anteriores, donde podrán revisar lo que han ordenado en el pasado.

2.8 REQUERIMIENTOS NO FUNCIONALES

2.8.1 Producto

2.8.1.1 Eficiencia

Núm. Requerimientos	Descripción
Rnf001: Seguridad de Datos	La aplicación implementa medidas de seguridad (AES) con información confidencial de los usuarios, como contraseñas y detalles de pago.
Rnf002: Tiempo de Respuesta	La aplicación tiene tiempos de respuesta menores a 0.5s para garantizar una experiencia de usuario fluida y eficiente.
Rnf003: Disponibilidad y Confiabilidad	La aplicación está disponible y funcionando correctamente todo tiempo.
Rnf004: Compatibilidad y Usabilidad	La aplicación es compatible con una variedad de dispositivos móviles sistemas operativos móviles de Android 7-13 También debe tener interfaces reconocibles y no supere más de 10 opciones en pantalla para que el usuario no se abraze
Rnf005: Escalabilidad	La aplicación es capaz de comunicarse con una base de datos de usuarios en la nube.
Rnf006: Cumplimiento Legal y Normativo	La aplicación es capaz cumplir con las regulaciones relacionadas con la venta de alimentos, protección de datos y transacciones en línea.
Rnf007: Integración con Medios de Pago	La aplicación es compatible con múltiples métodos de pago populares y seguros.
Rnf008: Gestión de Tráfico y Carga	La aplicación es capaz de gestionar de manera eficiente el tráfico de red es decir hasta 8,000 usuarios al mismo tiempo y la carga de servidores durante los períodos de alta demanda.
Rnf009: Experiencia de usuario Responsiva	La aplicación ofrece una experiencia de usuario óptima en smartphones y tabletas con al menos 0.5gb de Ram y Android 7 en adelante
Rnf010:	

2.9 Matriz de validación de requerimientos

	Completo 6	Consistente 5	Comprensible 4	Ambiguo 3	Estructurado 2	Trazable 1	Relevante	Gestionable	Viable
RF001				X					
RF002				X			X		X
RF003				X					
RF004									
RF005				X	X	X			
RF006									
RF007	X			X	X	X			
RF008									
RF009									
RF010			X		X				

2.10. Diagramas UML

CASO DE USO 1: CLIENTE-SOFTWARE

Título: La Salle Foods.

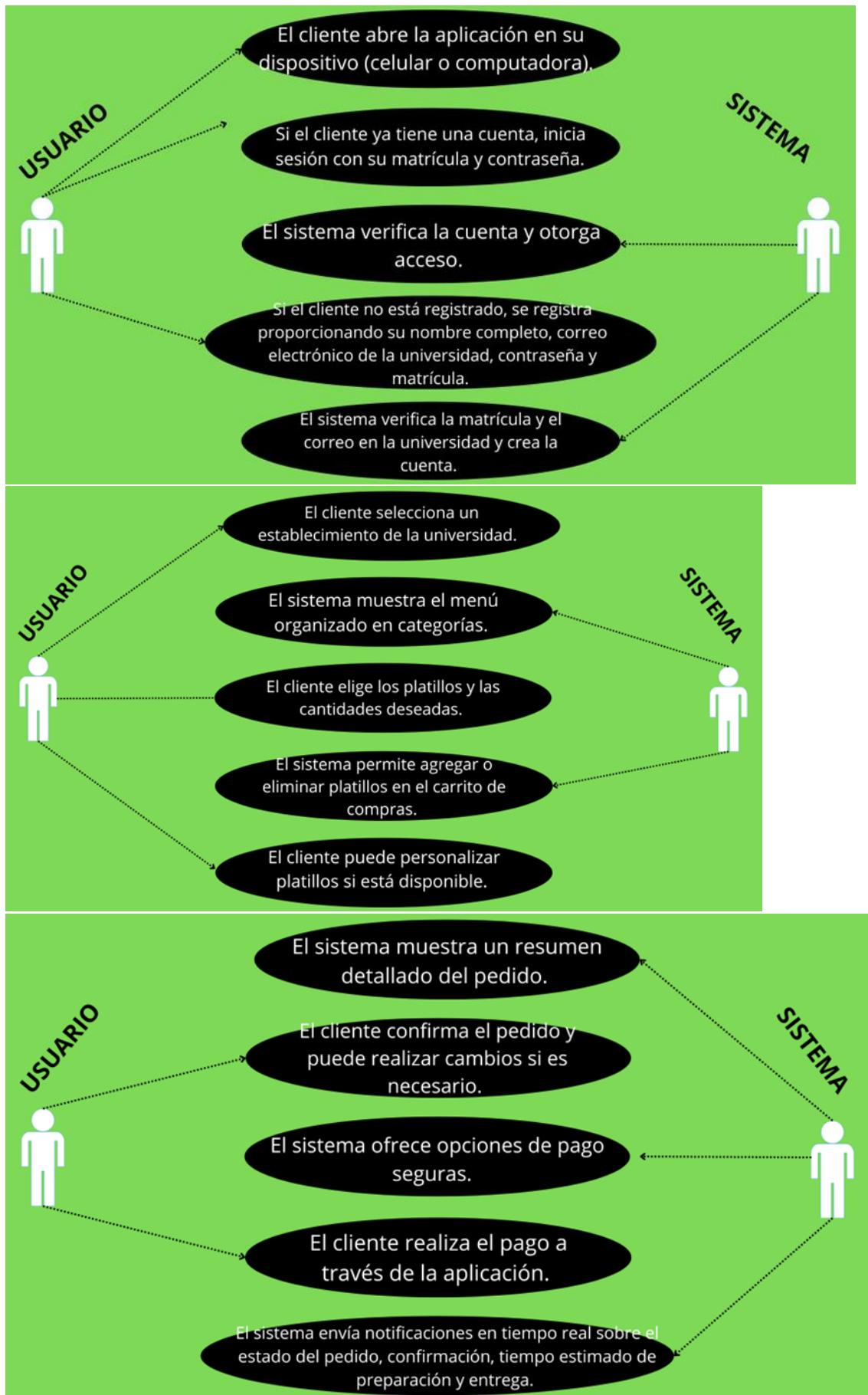
Actores: Cliente y Sistema.

ESCENARIO PRINCIPAL:

1. El cliente abre la aplicación en su dispositivo (celular o computadora).
2. Si el cliente ya tiene una cuenta, inicia sesión con su matrícula y contraseña.
3. El sistema verifica la cuenta y otorga acceso.
4. Si el cliente no está registrado, se registra proporcionando su nombre completo, correo electrónico de la universidad, contraseña y matrícula.
5. El sistema verifica la matrícula y el correo en la universidad y crea la cuenta.
6. El cliente selecciona un establecimiento de la universidad.
7. El sistema muestra el menú organizado en categorías.
8. El cliente elige los platillos y las cantidades deseadas.
9. El sistema permite agregar o eliminar platillos en el carrito de compras.
10. El cliente puede personalizar platillos si está disponible.
11. El sistema muestra un resumen detallado del pedido.
12. El cliente confirma el pedido y puede realizar cambios si es necesario.
13. El sistema ofrece opciones de pago seguras.
14. El cliente realiza el pago a través de la aplicación.
15. El sistema envía notificaciones en tiempo real sobre el estado del pedido, confirmación, tiempo estimado de preparación y entrega.

ESCENARIO SECUNDARIO:

1. Error de inicio de sesión
2. Correo o matrícula no registrados
3. Problema de conexión a Internet
4. Producto agotado
5. Problema de pago
6. Tiempo de entrega excedido



CASO DE USO 2: CLIENTE-SOFTWARE

CASO DE USO 2: TIENDAS – SISTEMA

ACTORES:

- 1-Tienda Universitaria (Cafeterías, tiendas): El personal de las tiendas que reciben y preparan los pedidos.
- 2-Sistema de Pedidos: La plataforma de software que permite a las tiendas gestionar los pedidos de comida en línea.

Precondiciones:

- La cafetería está registrada en el sistema y ha configurado su menú y horarios de funcionamiento.

Escenario Principal:

1. El personal de la tienda inicia sesión en el sistema con sus credenciales.
2. El sistema muestra una lista de pedidos pendientes de preparación.
3. La tienda selecciona un pedido pendiente para procesar.
4. El sistema muestra los detalles del pedido, incluyendo la descripción de los alimentos solicitados, en donde recoger el pedido y la hora del pedido.
5. La tienda confirma que tiene todos los elementos necesarios para preparar el pedido.
6. La tienda prepara el pedido y marca su estado como "en preparación" en el sistema.
7. El sistema notifica al usuario que su pedido está siendo preparado.
8. Una vez que el pedido está listo, la tienda lo marca como "listo para recoger" en el sistema.
9. El sistema notifica al usuario que su pedido está listo
10. El usuario recoge el pedido
11. El sistema registra la entrega como completada y cierra el pedido.
12. El usuario recibe una notificación de que su pedido ha sido entregado con éxito.

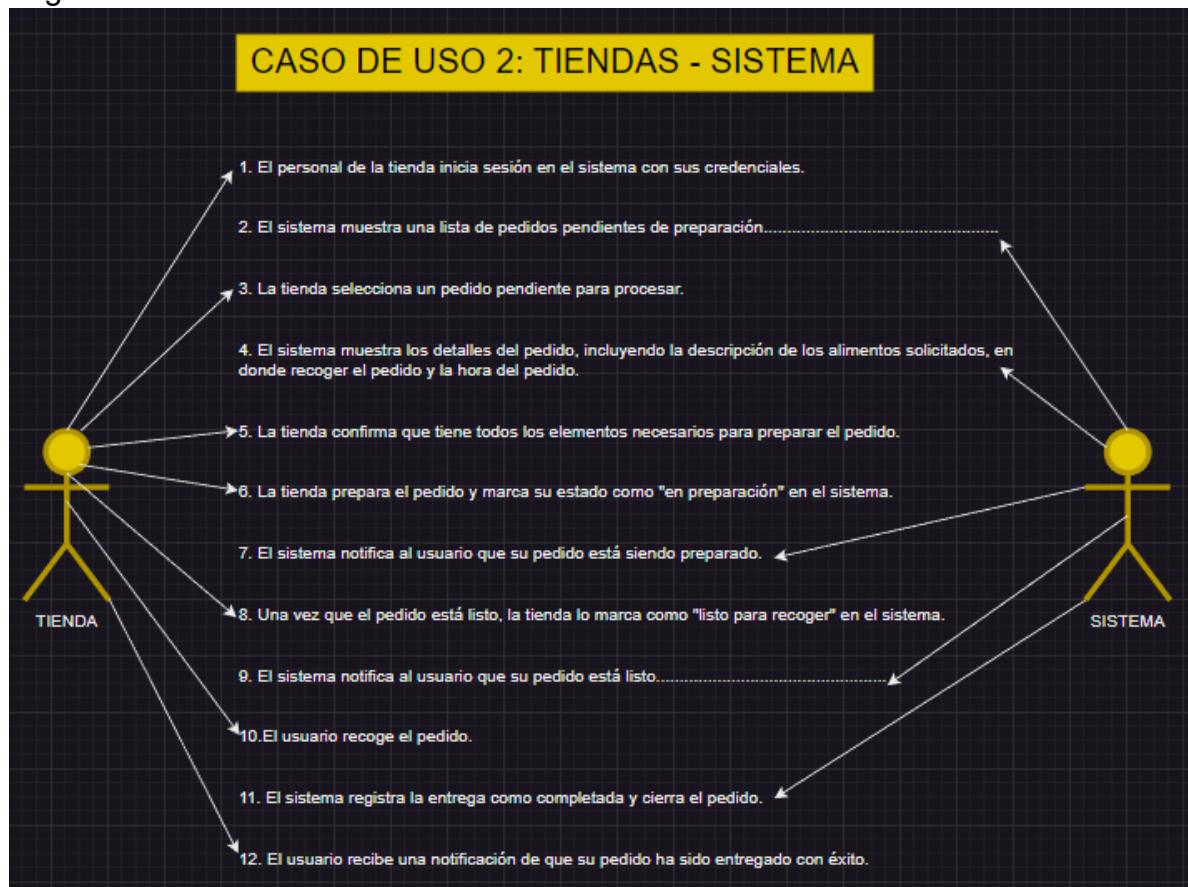
Escenario Secundario:

1-Cancelación del Pedido: Si el usuario cancela el pedido antes de la preparación (por ejemplo, debido a un cambio de planes), la cafetería puede cancelar el pedido en el sistema y notificar al usuario.

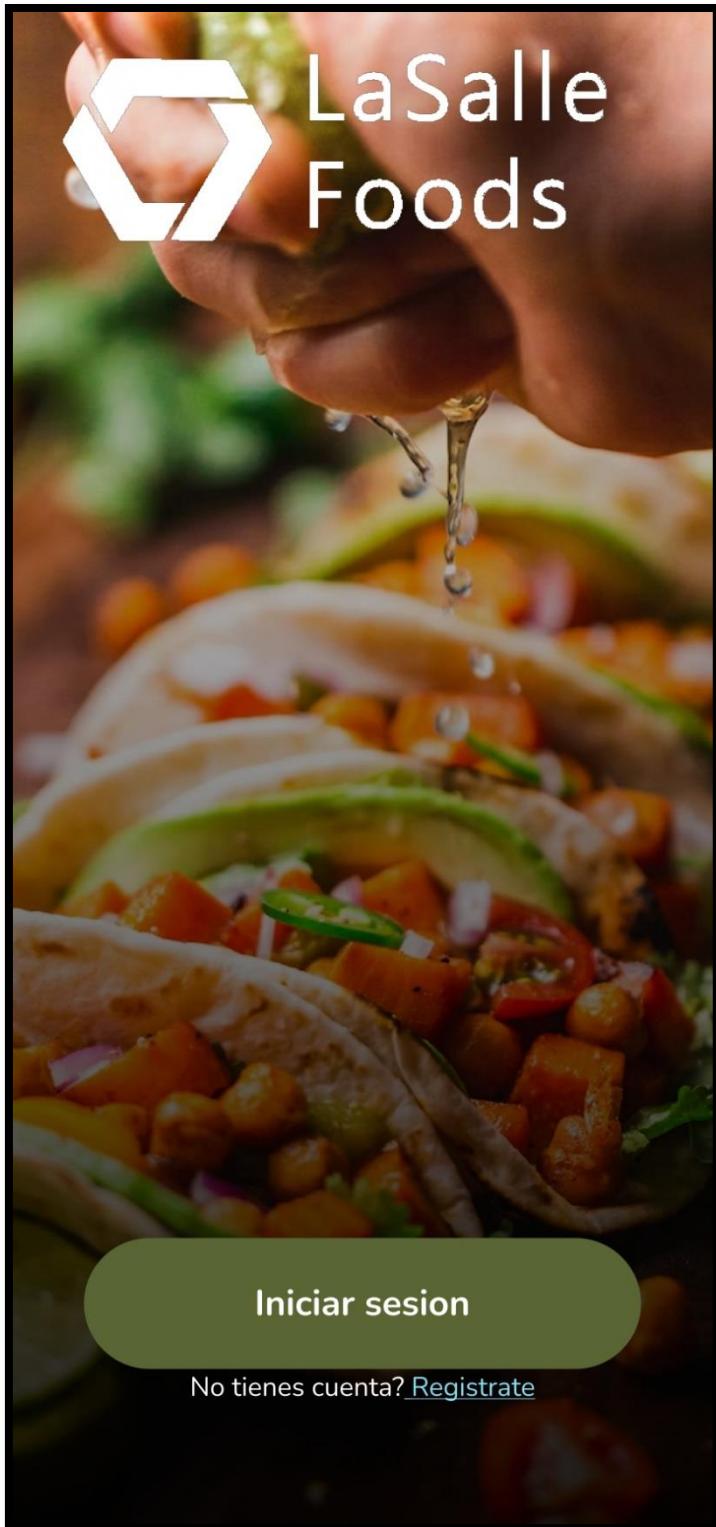
2-En el caso de que la tienda (cafetería) no tenga todos los elementos necesarios para preparar el pedido:

1. La tienda se comunica con el usuario a través del sistema para informar sobre la falta de elementos y acordar una solución, como un reembolso o una sustitución.
2. El usuario confirma la solución acordada.

3. La cafetería registra la solución en el sistema y procede según lo acordado.



3.-Diseño

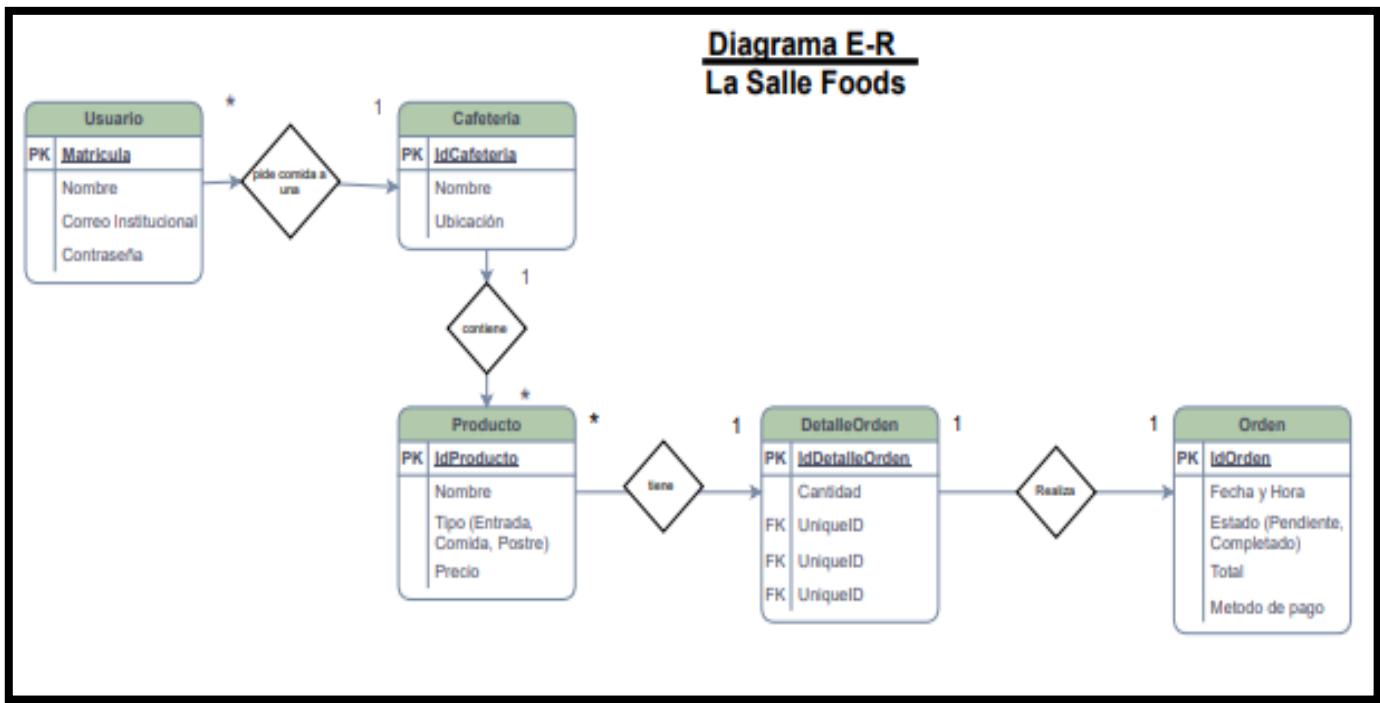








3.1- Diagrama



3.2- Algoritmo

```

    LaSalleFoods
    Archivo  Editar  Ver
    import java.util.HashMap;
    import java.util.Map;
    import java.util.Scanner;

    public class LaSalleFoods {
        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);

            // Definir precios de los platos
            Map<String, Double> menu = new HashMap<>();
            menu.put("Hamburguesa", 10.00);
            menu.put("Pizza", 12.00);
            menu.put("Ensalada", 8.00);
            menu.put("Pasta", 14.00);
            // Agrega más platos y precios aquí

            // Solicitar al usuario que inicie sesión o se registre
            boolean usuarioRegistrado = false;
            String usuario = null;

            while (!usuarioRegistrado) {
                System.out.print("{Desea iniciar sesión (I) o registrarse (R)? ");
                String opcion = scanner.nextLine().toUpperCase();

                if (opcion.equals("I")) {
                    System.out.print("Nombre de usuario: ");
                    usuario = scanner.nextLine();
                    // Aquí podrías verificar si el usuario existe en una base de datos
                    // y si la contraseña es correcta.
                    usuarioRegistrado = true;
                } else if (opcion.equals("R")) {
                    System.out.print("Nombre de usuario: ");
                    usuario = scanner.nextLine();
                    // Aquí podrías registrar al usuario en una base de datos.
                }
            }
        }
    }
    Ln 6, Col 45  100%  Windows (CRLF)  UTF-8
  
```

The image shows two side-by-side code editors, both titled "LaSalleFoods". The top editor contains the following Java code:

```
    usuarioRegistrado = true;
} else {
    System.out.println("Opción no válida. Por favor, elija 'I' para iniciar sesión o 'R' para registrarse.");
}
}

// Mostrar el menú al usuario
System.out.println("Menú:");
for (Map.Entry<String, Double> plato : menu.entrySet()) {
    System.out.println(plato.getKey() + ": $" + plato.getValue());
}

// Solicitar al usuario que seleccione platos y cantidades
Map<String, Integer> orden = new HashMap<>();
boolean continuarOrdenando = true;

while (continuarOrdenando) {
    System.out.print("Elige un plato del menú o ingrese 'fin' para terminar la orden: ");
    String plato = scanner.next();

    if (menu.containsKey(plato)) {
        System.out.print("Cantidad de " + plato + ": ");
        int cantidad = scanner.nextInt();
        orden.put(plato, cantidad);
    } else if (plato.equals("fin")) {
        continuarOrdenando = false;
    } else {
        System.out.println("Plato no válido. Por favor, elija un plato del menú o ingrese 'fin'.");
    }
}

// Calcular el costo total de la orden
double total = 0;

for (Map.Entry<String, Integer> entrada : orden.entrySet()) {
    System.out.println("Entrada: " + entrada.getKey() + " cantidad: " + entrada.getValue());
    total += menu.get(entrada.getKey()) * entrada.getValue();
}

System.out.println("Costo total: $" + total);
}

// Mostrar la orden al usuario
System.out.println("Su orden:");
for (Map.Entry<String, Integer> entrada : orden.entrySet()) {
    String plato = entrada.getKey();
    int cantidad = entrada.getValue();
    System.out.println(plato + ": " + cantidad);
}

System.out.println("Costo total: $" + total);

// Solicitar al usuario el método de pago
System.out.print("Elige un método de pago (tarjeta de crédito, efectivo, PayPal, etc.): ");
String metodoPago = scanner.next();

// Solicitar al usuario el tiempo de entrega
System.out.print("Ingrese el tiempo de entrega (en minutos): ");
int tiempoEntrega = scanner.nextInt();

// Procesar el pago y confirmar la orden
System.out.println("Pagado $" + total + " con " + metodoPago + ". ¡Gracias por su pedido!");

// Aquí podrías implementar la entrega o envío de la orden.
}
```

The bottom editor contains the following Java code:

```
// Calcular el costo total de la orden
double total = 0;

for (Map.Entry<String, Integer> entrada : orden.entrySet()) {
    String plato = entrada.getKey();
    int cantidad = entrada.getValue();
    total += menu.get(plato) * cantidad;
}

// Mostrar la orden al usuario
System.out.println("Su orden:");
for (Map.Entry<String, Integer> entrada : orden.entrySet()) {
    String plato = entrada.getKey();
    int cantidad = entrada.getValue();
    System.out.println(plato + ": " + cantidad);
}

System.out.println("Costo total: $" + total);

// Solicitar al usuario el método de pago
System.out.print("Elige un método de pago (tarjeta de crédito, efectivo, PayPal, etc.): ");
String metodoPago = scanner.next();

// Solicitar al usuario el tiempo de entrega
System.out.print("Ingrese el tiempo de entrega (en minutos): ");
int tiempoEntrega = scanner.nextInt();

// Procesar el pago y confirmar la orden
System.out.println("Pagado $" + total + " con " + metodoPago + ". ¡Gracias por su pedido!");

// Aquí podrías implementar la entrega o envío de la orden.
```

3.3- Entradas

Entradas	Clases validas	Clases invalidas
E-mail Institucional	(5-30)Caracteres alfanumericos,Simbolos,@.	NULL
Matricula	(4-8) Caracteres Numericos	NULL,Simbolos,@,Caracteres Alfabeticos, Espacios
Nombre	(3-30) Caracteres Alfabeticos, Espacios.	NULL Simbolos,@,Caracteres Numericos
Contraseña	(1-50) Caracteres alfanumericos,Simbolos,Espacios	NULL
Confirmar Contraseña	(1-50) Caracteres alfanumericos,Simbolos,Espacios	NULL, Debe ser exactamente igual al "Campo de Contraseña"
Crear Cuenta	Manda la informacion a la base de datos	No Guardar



Entradas	Clases validas	Clases invalidas
Matricula	(4-8) Caracteres Numericos	NULL,Simbolos,@,Caracteres Alfabeticos, Espacios
Contraseña	(1-50) Caracteres alfanumericos,Simbolos,Espacios	NULL



3.4- Algoritmo Entradas

```
# Definir precios de los platos
menu = {
    "Hamburguesa": 10.00,
    "Pizza": 12.00,
    "Ensalada": 8.00,
    "Pasta": 14.00,
    # Agrega más platos y precios aquí
}

# Mostrar el menú al usuario
Mostrar "Menú:"
Para cada plato, precio en menu:
    Mostrar plato + ": $" + precio

# Solicitar al usuario que seleccione platos y cantidades
orden = {}
Repetir:
    Solicitar al usuario que elija un plato del menú o ingrese
    'fin' para terminar la orden
    plato = Entrada del usuario
    Si plato en menu:
        Solicitar al usuario la cantidad de plato
        cantidad = Convertir a entero la entrada del usuario
        orden[plato] = cantidad
    Sino:
        Mostrar "Plato no válido. Por favor, elija un plato del
        menú."

# Calcular el costo total de la orden
total = 0
Para cada plato, cantidad en orden:
    total = total + menu[plato] * cantidad

# Mostrar la orden al usuario
Mostrar "Su orden:"
Para cada plato, cantidad en orden:
    Mostrar plato + ": " + cantidad
Mostrar "Costo total: $" + total

# Solicitar al usuario el método de pago
Solicitar al usuario que elija un método de pago (tarjeta de
crédito, efectivo, PayPal, etc.)
metodo_pago = Entrada del usuario

# Procesar el pago y confirmar la orden
Mostrar "Pagado $" + total + " con " + metodo_pago + ".
¡Gracias por su pedido!"
```

7. Anexo

Entrevista en base a requerimientos la Salle food.

Entrevista 1

Entrevistador: Gracias por participar en esta entrevista. Estamos en la fase de planificación de una nueva aplicación. Nos gustaría conocer tus opiniones y necesidades. Mantendremos tu nombre en anónimo para la protección de datos.

Pregunta 1: ¿Qué te gustaría que una aplicación de entrega de alimentos o pedidos en un restaurante haga por ti?

Respuesta: Me gustaría poder ordenar comida de manera conveniente y personalizada desde diferentes restaurantes y hacer un seguimiento de mis pedidos en tiempo real.

Pregunta 2: ¿Qué información crees que sería importante proporcionar al registrarte en la aplicación?

Respuesta: Nombre, matrícula y correo electrónico (con el dominio "@lasallebajio.edu.mx") serían esenciales. Además, una contraseña segura

Pregunta 3: ¿Qué características de inicio de sesión te resultarían más útiles?

Respuesta: Un inicio de sesión rápido y seguro utilizando mi matrícula y contraseña previamente registrada.

Pregunta 4: ¿Cómo te gustaría que se organice el menú de los restaurantes en la aplicación?

Respuesta: Sería útil si el menú se organiza en categorías claras, como entradas, platos principales y postres. También, la capacidad de buscar por nombre de plato.

Pregunta 5: ¿Qué funciones te gustaría ver en el carrito de compras? ¿Algún otro detalle importante?

Respuesta: Poder agregar y eliminar elementos fácilmente. Además, ver el costo total de mi pedido antes de confirmarlo.

Pregunta 6: ¿Te gustaría la opción de personalizar algunos platillos, como elegir ingredientes?

Respuesta: Sí, eso sería genial. A veces tengo preferencias específicas y me gustaría poder personalizar mi pedido

Pregunta 7: ¿Qué información te gustaría ver en la gestión de pedidos y en la confirmación?

Respuesta: Un resumen detallado del pedido, incluyendo la lista de platillos, el costo total y el tiempo estimado de entrega.

Pregunta 8: ¿De qué manera te gustaría poder seleccionar diferentes establecimientos para realizar pedidos?

Respuesta: Debería ser fácil elegir entre diferentes restaurantes disponibles para pedir. Una lista clara sería útil.

Pregunta 9: ¿Qué métodos de pago te sentirías más cómodo utilizando?

Respuesta: Tarjeta de crédito y efectivo serían opciones ideales para mí.

Pregunta 10: ¿Qué tipo de notificaciones en tiempo real considerarías útiles?

Respuesta: Notificaciones que me informen sobre el estado de mi pedido, la confirmación de pedido, el tiempo estimado de preparación y cuando esté listo para recogerlo.

Pregunta 11: ¿Te gustaría tener acceso a un historial de tus pedidos anteriores en la aplicación?

Respuesta: Sí, sería útil para revisar lo que he ordenado en el pasado y repetir pedidos anteriores si lo deseo.

Pregunta 12: ¿Tienes alguna otra sugerencia o comentario que quieras compartir sobre esta futura aplicación?

Respuesta: Me gustaría que la aplicación sea fácil de usar y que tenga una interfaz intuitiva. También, sería genial si se pudiera agregar una función de seguimiento del repartidor en tiempo real.

