

Práctica 2

Aura Gourmet System

Programación Orientada a Objetos(3007744)

Grupo 1

Equipo 3

Kevin Andrés Rubiano Gómez

Andrés Felipe Chica Ospina

Mateo Pérez Gutiérrez

María de los ángeles Vargas Reales

Diana Alejandra Jacanamejoy Getial

Profesor: Jaime Alberto Guzmán Luna

Universidad Nacional de Colombia - Sede Medellín

Facultad de Minas

2025

1.Descripción General de la Solución

Identificación de los Problemas y Necesidades:

El sistema realizado está diseñado para llevar a cabo las tareas de un administrador de reservas y pedidos en un restaurante, para ello se examinan los **problemas que se planean resolver** tales como:

- Confusiones y duplicaciones al registrar reservas manualmente.
- Errores en la toma de pedidos (por ejemplo, omisiones o interpretaciones incorrectas).
- Falta de retroalimentación estructurada sobre la calidad de la comida y el servicio.
- Desorganización en los pedidos a domicilio, lo que genera entregas tardías o a direcciones incorrectas.
- Necesidad de emplear más personal para tareas administrativas o manuales.
- Lentitud en el proceso de generación de facturas y errores en cálculos manuales.
- Falta de información para evaluar el rendimiento del restaurante.
- Largas esperas, falta de información sobre disponibilidad y mala comunicación.

Además se tienen en cuenta las **necesidades específicas de los usuarios:**

- Poder realizar una reserva con base a un horario específico
- Realizar el pedido para consumir los alimentos que se quieren
- Calificar el servicio recibido por parte del restaurante, meseros, atención en general
- La capacidad de solicitar domicilios en caso tal que se requiera
- Realizar el pago correspondiente por el servicio prestado.
- Llevar un registro de los clientes, para implementar descuentos o puntos, haciendo más atractivo el restaurante a los clientes.
- Brindar una experiencia personalizada al cliente.

Para dar solución a las diferentes necesidades del usuario, se deberán solucionar las **necesidades o problemas del sistema:**

- Manejar reservas en horarios específicos y evitar conflictos en la asignación de mesas.
- Enviar confirmaciones automáticas de reservas a los clientes.
- Permitir a los usuarios realizar pedidos detallados (en el restaurante o para domicilio).
- Recoger y registrar la retroalimentación de los clientes sobre el servicio, comida y atención.
- Registrar pedidos a domicilio
- Generar facturas automáticamente y sin errores, con detalles como impuestos, descuentos y propinas.
- Llevar un registro de clientes frecuentes, con información para ofrecer promociones, puntos de fidelidad o descuentos.
- Ofrecer recomendaciones personalizadas basadas en las preferencias del cliente.
- Analizar en general el servicio ofrecido por el restaurante.

Requisitos Funcionales:

1. Gestión de Reservas:

- a)Registrar y gestionar reservas en horarios específicos.

b)Asignar automáticamente mesas disponibles según el tamaño del grupo.

2. Gestión de Pedidos:

a)Permitir a los usuarios realizar pedidos detallados

3. Gestión de Recompensas:

a)Generar recompensas especiales por fidelidad al restaurante

b)Permitir al usuario hacer uso de las recompensas asignadas

4. Retroalimentación de Clientes:

a)Permitir a los clientes calificar el servicio, comida y atención general.

5. Gestión de Pedidos a domicilio:

a)Registrar pedidos a domicilio con detalles como dirección, prioridad del domicilio y el domiciliario encargado

6. Gestión de Clientes:

a)Llevar un registro de clientes frecuentes con información como historial de visitas y preferencias.

b)Ofrecer promociones, puntos de fidelidad y descuentos personalizados.

7. Personalización de la experiencia:

a)Generar recomendaciones basadas en las preferencias registradas de los clientes.

Requisitos No Funcionales:

1. Rendimiento:El sistema debe procesar solicitudes sin demoras extensas.

2. Seguridad:Debe proteger los datos sensibles de los clientes que solo serán accesados por quienes lo necesiten.

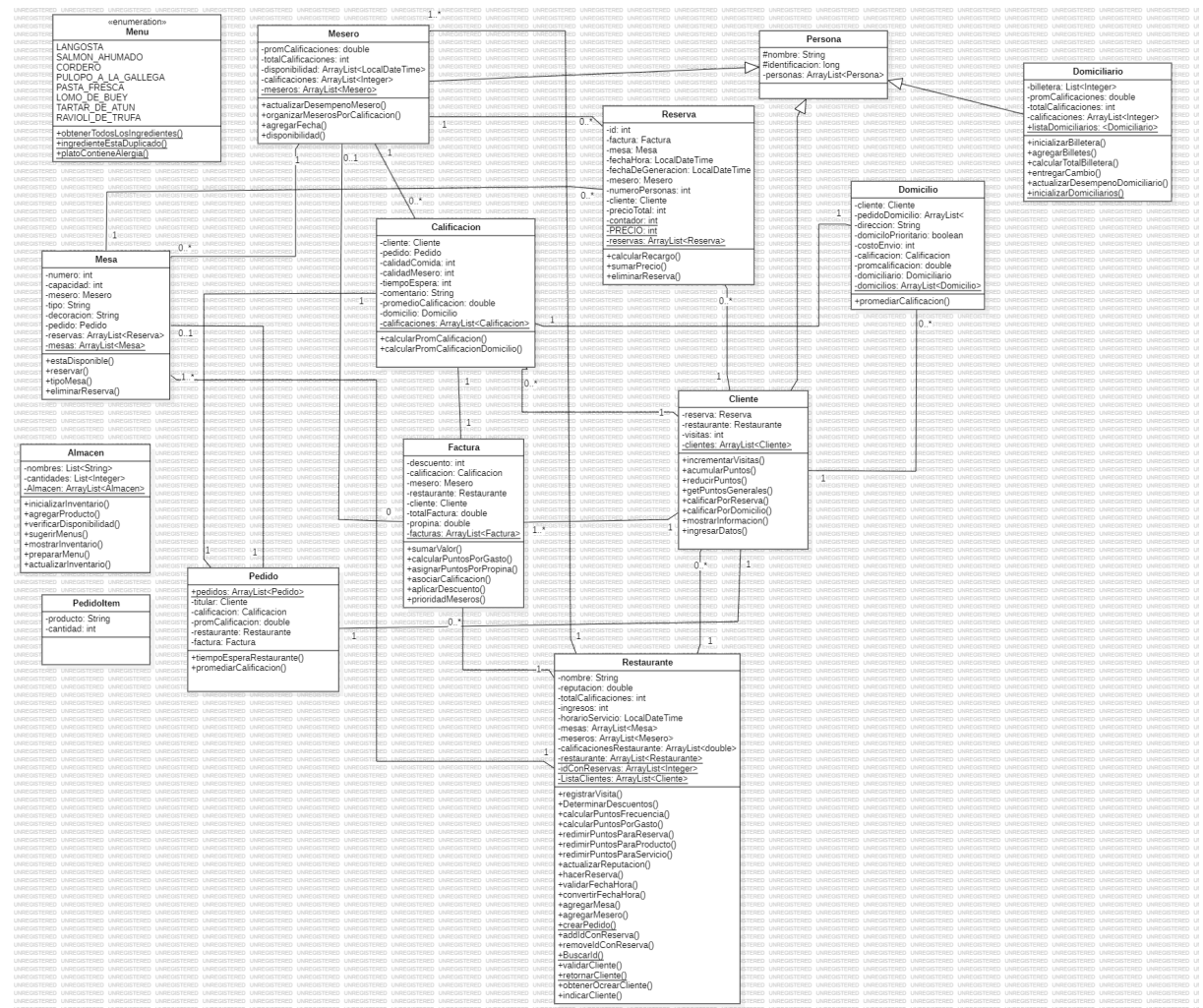
3. Disponibilidad:El sistema debe estar siempre disponible para uso de los clientes y el restaurante

4. Escalabilidad:El diseño debe permitir añadir nuevas funcionalidades sin afectar el código previo

5. Usabilidad:Interfaz intuitiva y fácil de usar, que requiera un mínimo de instrucciones

2.Descripción del diseño estático del sistema en la especificación UML

UML:



Link PDF: [PDF](#) Untitled.pdf

Descripción de las Clases:

Restaurante: Su función es encargarse de validar fechas y horas, hacer reservas, redimir puntos, almacenar las mesas, meseros y listas importantes como la lista de clientes y la lista de reservas. Posee cardinalidades con Cliente, Factura, Mesero, Mesa. Le asigna al cliente la factura, solicita información a las mesas, y gestiona a los meseros.

Reserva: Su función es guardar y gestionar información relevante de la Reserva, Posee cardinalidades con Mesero, Mesa y Cliente. A la reserva se le asigna un Mesero que la atenderá, una Mesa asignada para el Cliente, y por supuesto un Cliente asignado a la reserva.

Persona: Clase padre de Mesero, Domiciliario y Cliente, encargada de asignar información básica, como nombre e identificación.

Calificación : Su función es administrar las calificaciones obtenidas y asociarlas a su objeto de tipo Cliente correspondiente, también a su vez se encarga de mantener actualizadas la reputación del restaurante y de todos sus meseros cada vez que se les califica.

Cliente: Su función es mantener un control de las personas que ingresan al restaurante. Posee cardinalidades con Reserva, Calificación, Factura, Restaurante, Pedido, Domicilio y Persona. El cliente hereda de la clase Persona, se le asigna una Factura que deberá pagar, un Restaurante donde asistirá, un Pedido que contiene los alimentos solicitados, una Reserva para asistir en el momento adecuado, una Calificación y Domicilio opcionales.

Mesero: Su función es encargarse de atender las mesas, posee cardinalidad con la clase Mesa, Calificación, Reserva y Restaurante. El mesero es asignado a una mesa, una reserva y pertenece a un restaurante, además puede recibir calificaciones.

Domiciliario: Su función es llevar los domicilios ordenados por la clase Domicilio y devolver dinero. Posee cardinalidades con Persona, que es su clase padre.

Domicilio: Su función es crear los domicilios en base a la información suministrada por el usuario. Posee cardinalidades con Calificación y Cliente

Factura: Su función es cobrar al cliente todo lo consumido. Posee cardinalidades con Calificación, Cliente, Mesero, Restaurante.

Pedido: Se encarga de almacenar los métodos necesarios para concluir de manera exitosa la creación de la lista de platos que el cliente desea consumir y posteriormente calcular el valor a pagar. Posee cardinalidades con Cliente y Calificación.

Menu: Es un enum que almacena los platos del menú principal, sus ingredientes, y precios. En esta clase también se almacena el enum MenuCortesias que contiene los platos que se le ofrece a los clientes que reservan una mesa deluxe. No tiene cardinalidades por ser un enum

Almacen: Guarda los ingredientes necesarios para realizar los platos y los métodos necesarios para ir haciendo actualización de inventario a medida que se crean pedidos.

Excepciones: Esta clase es encargada de manejar las diferentes excepciones que puedan ser lanzadas en el sistema, se encuentra muy relacionada con la InterfazUsuario para funcionar correctamente

FieldFrame: Esta clase se encarga de solicitar información al usuario de una manera predeterminada, es decir, permite a cada funcionalidad, independientemente de los tipos de datos que pida, usar esta clase para solicitar información.

InterfazUsuario: Se encarga de manejar todo lo relacionado con la GUI, además funciona como método main para generar una ejecución de código más limpia

3.Descripción de la implementación de características de programación orientada a objetos.

Como parte de lo solicitado en los comicios de la elaboración del proyecto, el programa se ha diseñado contractualmente con estas instrucciones basadas en las herramientas propias del paradigma que acoge el lenguaje de programación Java: Programación Orientada a Objetos (POO). Las características implementadas, junto a su uso y ubicación, son:

Herencia (1 caso):

La clase Cliente, Domiciliario y Mesero heredan de la clase Persona, a continuación se puede ver cómo se implementa en la clase Cliente, esto con el objetivo de que Cliente Domiciliario y Mesero puedan tener los atributos nombre e identificación

```
1  class Persona:
2      _personas = [] # Lista estática para almacenar personas
3
4      def __init__(self, nombre: str = None, identificacion: int = None):
5          self._nombre = nombre
6          self._identificacion = identificacion
7          Persona._personas.append(self)
8
```

```
5  class Cliente(Persona):
6      _clientes = []
7
8      def __init__(self, nombre=None, identificacion=None, restaurante=None, reserva=None):
9          super().__init__(nombre, identificacion)
10         if restaurante is None:
11             restaurante = self.crear_o_asignar_restaurante("Aura Gourmet")
12         self._reserva = reserva
13         self._pedido = None
14         self._restaurante = restaurante
15         self._visitas = 0
16         self._visitas_para_descuentos = 0
17         self._descuento_por_visitas = 0
18         restaurante._lista_clientes.append(self)
19         Cliente._clientes.append(self)
```

Ligadura Dinámica

```
pr-ctica-2-g1-e3 > gestorAplicacion > persona.py > Persona > get_personas
1 class Persona:
24
25     def saludar(self):
26         print(f"hola {self.get_nombre()}! :)")
```

```
pr-ctica-2-g1-e3 > gestorAplicacion > cliente.py > Cliente > _str_
5 class Cliente(Persona):
69
70     def saludar(self):
71         mensaje = ""
72         if self.get_descuento_por_visitas() == 0:
73             mensaje = (f"¡Reserva encontrada!\nBienvenid@ {self.get_nombre()}, "
74                       f" esta es su visita #{self.get_visitas()} "
75                       f"a Aura Gourmet\n"
76                       f"(recuerde que tenemos descuentos por fidelidad cada 5 visitas)")
77         else:
78             mensaje = (f"¡Reserva encontrada!\nBienvenid@ {self.get_nombre()}, "
79                       f"por ser su visita #{self.get_visitas()} a Aura Gourmet,\n"
80                       f"obtendrá un descuento del {self.get_descuento_por_visitas()}%"
81                       f"sobre el total de su pedido")
82         return mensaje
```

La clase Cliente hereda de la clase Persona y Cliente tiene definido con el mismo nombre un método de la clase Persona, en este caso, saludar.

La ligadura dinámica permite que el método correcto sea llamado en tiempo de ejecución, basándose en el tipo del objeto. Por ejemplo cuando llamamos al método saludar para mostrar un mensaje de bienvenida al cliente, indicarle su número de vistas y (si aplica) informarle sobre sus descuentos en el labelv2

```
pr-ctica-2-g1-e3 > InterfazUsuario.py > funcionalidad3 > buscarReserva
1526
1529 limpiar_frame()
1530 labelv2.config(text= clienteConReserva.saludar(), fg='red') #llama al metodo saludar
1531
```

Cuando estamos realizando un pedido asociado a una reserva y después de confirmar la reserva se puede ver que se ejecuta correctamente el método asociado al tipo de objeto, en este caso el método saludar de cliente.

```
Realizar el pedido de mi reserva

¡Reserva encontrada!
Bienvenid@ mariano herrera, esta es su visita #1 a Aura Gourmet
(recuerde que tenemos descuentos por fidelidad cada 5 visitas)
```

Atributos de clase (1 caso) y métodos de clase (1 caso):

El **atributo de clase** se encuentra en la clase Restaurante y se usa para llevar un registro de todos los clientes que se crean, guardándolos en `_lista_clientes`

```
6 class Restaurante:
7     _restaurantes = []
8     _lista_clientes = []
9
```

El **método de clase** se encuentra en la clase Cliente y es usado para validar si un cliente ya existe en el listado de `_clientes`

```
107 @staticmethod
108 def validarCliente(ident):
109     for cliente in Cliente.get_clientes():
110         if cliente.get_identificacion() == ident:
111             return True
112     return False
113
```

Uso de constante (1 caso):

Se usa en la clase Reserva una constante para definir el precio base que cuesta realizar una reserva.

```
6 class Reserva:
7     _reservas = []
8     _contador = 0
9     _PRECIO = 30000
```

Encapsulamiento

```
6 class Restaurante:
7     _restaurantes = []
8     _id_con_reservas = [1990, 2004] #para probar
9     _lista_clientes = []
10
11     def __init__(self, nombre, horario_servicio=None):
12         self._nombre = nombre
13         self._reputacion = 3.8
14         self._total_calificaciones = 30
15         self._ingresos = 0
16         self._horario_servicio = horario_servicio
17         self._mesas = []
18         self._meseros = []
19         self._calificaciones_restaurante = []
20         Restaurante._restaurantes.append(self)
21
```


Se muestra como, aunque en Python no haya como tal encapsulamiento, como buena práctica, se indica que los atributos deben ser private

Manejo de referencias this para desambiguar y this() entre otras. 2 casos mínimo para cada caso:

Para **desambiguar** usamos **self** en varias ocasiones, como en todos los métodos, un ejemplo está en la clase Reserva, usando self para que se pueda añadir precios correctamente

```
26     def calcular_recargo(self):
27         años_diferencia = self._fecha_hora.year - self._fecha_de_generacion.year
28         meses_diferencia = (años_diferencia * 12) + (self._fecha_hora.month - self._fecha_de_generacion.month)
29
30         if meses_diferencia > 1:
31             self._precio_total += 50000
32             self._recargo_por_fecha = True
33
```

Y en la clase Factura para aplicar descuentos de manera adecuada

```
49     def aplicar_descuento(self, calificacion):
50         if calificacion:
51             promedio = calificacion.get_promedio_calificacion()
52
53             if promedio <= 2:
54                 self._descuento = 10 # 10% de descuento
55                 self._total_factura -= (self._total_factura * self._descuento) / 100
56                 return "10% de descuento"
57             elif promedio <= 3:
58                 self._descuento = 5 # 5% de descuento
59                 self._total_factura -= (self._total_factura * self._descuento) / 100
60                 return "5% de descuento"
61             else:
62                 self._descuento = 0 # sin descuento
63                 self._total_factura -= (self._total_factura * self._descuento) / 100
64                 return "Sin descuento"
65         return "N/A"
```

También se usa self para ordenar a las clases hacer algo con el objeto, como en la clase Reserva, que usamos self para que se genere el código único automáticamente.

```
6     class Reserva:
7         _reservas = []
8         _contador = 0
9         _PRECIO = 30000
10
11     def __init__(self, mesa, fecha_hora=None, numero_personas=None, fecha_de_generacion=None, cliente=None, mesero=None):
12         self._mesa = mesa
13         self._fecha_hora = fecha_hora
14         self._numero_personas = numero_personas
15         self._fecha_de_generacion = fecha_de_generacion
16         self._cliente = cliente
17         self._mesero = mesero
18         self._precio_total = self._PRECIO
19         self._recargo_por_fecha = False
20         self._factura = Factura()
21         self._id = self.generar_id()
22         if fecha_hora and fecha_de_generacion:
23             self.calcular_recargo()
24         Reserva._reservas.append(self)
```

Implementación de un caso de enumeración:

Usamos la enumeración en la clase Menu para guardar los platos que ofrece el restaurante con sus precios e ingredientes

```
3 class Menu(Enum):
4     LANGOSTA = ("Langosta", 120000, ["langosta", "mantequilla", "limon"])
5     SALMON_AHUMADO = ("Salmón Ahumado", 85000, ["salmon", "hierbas finas", "aceite de oliva"])
6     CORDERO = ("Cordero", 95000, ["cordero", "hierbas finas", "ajo"])
7     PULPO_A_LA_GALLEGA = ("Pulpo a la Gallega", 110000, ["pulpo", "pimenton", "aceite de oliva"])
8     PASTA_FRESCA = ("Pasta Fresca", 60000, ["pasta", "tomate", "albahaca"])
9     LOMO_DE_BUEY = ("Lomo de Buey", 135000, ["lomo de buey", "sal", "pimienta"])
10    TARTAR_DE_ATUN = ("Tartar de Atún", 98000, ["atun", "aguacate", "salsa de soya"])
11    RAVIOLI_DE_TRUFA = ("Ravioli de Trufa", 115000, ["ravioli", "trufa", "queso parmesano"])
12
13    def __init__(self, nombre, precio, ingredientes):
14        self._nombre = nombre
15        self._precio = precio
16        self._ingredientes = ingredientes
17
```

5.Descripción de cada una de las 5 funcionalidades implementadas.

Funcionalidad 1 (Realizar Reserva):

Inicio del proceso:

- Se inicializan variables como nombre, identificación, personas, tipoMesa, fecha, horaReserva, fechaReserva y cliente.
- Se configuran etiquetas (labelv1 y labelv2) para mostrar instrucciones en la interfaz gráfica.

Captura de datos del cliente

- Se muestra un formulario (FieldFrame) para que el usuario ingrese:
 - Nombre
 - Número de identificación
- Una vez que el usuario ingresa esta información, se destruye el formulario y se muestra un nuevo FieldFrame para capturar:
 - Número de personas
 - Tipo de mesa (básica o deluxe)
 - Fecha de la reserva
 - Hora de la reserva

Validación de la fecha y búsqueda de mesas disponibles:

- Se valida si la fecha y la hora ingresadas son correctas con `restaurante.validar_fecha(fecha)` y `restaurante.validar_hora(horaReserva)`.
- Si son válidas, se convierten en un formato de fecha-hora (`restaurante.convertir_fecha_hora(fecha, horaReserva)`) y se buscan mesas disponibles con `restaurante.hacer_reserva(fechaReserva, personas, tipoMesa)`.
- Si la fecha u hora son inválidas, se vuelve a mostrar el formulario para corregir los datos.

Elección de la mesa

- Se muestra un nuevo FieldFrame con las mesas disponibles.

- El usuario elige un número de mesa, y el sistema busca en la lista de mesas (Mesa._mesas) para asignarla.

Creación de la reserva

- Se crea una instancia de Reserva, asociándola con la mesa elegida, la fecha de reserva y el número de personas.
- Se asigna un mesero (meseroAsignado = mesaEscogida.reservar(reserva)).
- Se obtiene o crea un cliente en el sistema (restaurante.obtener_o_crear_cliente(nombre, identificacion)).
- Se asocia la reserva con el cliente y su factura (reserva.set_cliente(cliente)).

Personalización adicional (solo para mesas "deluxe")

Si el cliente eligió una mesa deluxe, se solicita información adicional:

- Decoración de la mesa (Ej: Elegante, Rústico, Moderno).
- Hora adicional:
 - Si el usuario elige agregar una hora extra, se verifica si la mesa está disponible durante ese tiempo.
 - Si es posible, se suma una hora a la reserva y se cobra un recargo de 30,000.

Gestión de alergias (solo para mesas "deluxe")

- Se solicita al cliente que ingrese ingredientes a los que es alérgico.
- Se filtran los platos del menú para mostrar solo aquellos que no contienen los ingredientes restringidos.

Resumen de la reserva

- Se muestra un resumen con todos los detalles de la reserva:
 - Nombre del cliente
 - Identificación
 - Número de personas
 - Tipo de mesa
 - Fecha y hora de la reserva
 - Número de reserva (ID único)
 - Decoración (si aplica)
 - Hora adicional (si aplica)
 - Alergias (si aplica)

Finalización del proceso

- Se muestra el resumen en un FieldFrame.
- El usuario puede confirmar la reserva y salir del proceso.

Funcionalidad 2 (Domicilios):

Esta funcionalidad permite al usuario realizar un pedido a domicilio, solicitando información básica como nombre, identificación, dirección, prioridad del pedido, selección de alimentos y su cantidad, además del pago. Se divide en tres interacciones principales que cubren todo el flujo, desde la validación de los datos del cliente hasta la generación del domicilio y el procesamiento del pago.

Interaccion 1: Pedir los datos del cliente y validarlos

1. Clases utilizadas:

- **Persona:** Clase padre de cliente
- **Cliente:** Representa al cliente que realiza el pedido y hereda atributos y métodos de Persona
- **Restaurante:** Proporciona información del restaurante para asociar al cliente.
- **Serializador:** Serializa los datos del cliente para almacenarlos.

2. Métodos utilizados:

- Inicializadores de las clases utilizadas en las demás interacciones.
- Getters y setters de todas los atributos utilizados en las demás interacciones

3. Flujo:

- **Nombre:** Solicita el nombre del cliente y verifica que no esté vacío.
- **Dirección:** Valida que la dirección ingresada siga un formato específico.
- **Identificación:** Solicita un número de identificación y verifica que sea un número positivo.
- **Prioridad del pedido:** Pregunta si el cliente desea un pedido prioritario y valida la respuesta (sí o no).
- **Creación del cliente:** Con la información validada, se crea un nuevo objeto Cliente que se almacena en la base de datos.

4. Resumen: Esta interacción se encarga de asociar el objeto creado de cliente a un domicilio y verificar que los datos estén correctamente diligenciados, además de verificar que las siguientes interacciones puedan funcionar correctamente (el menú tenga alimentos suficientes o que el domiciliario tenga cambio suficiente, etc).

Interacción 2: Procesar el pedido

1. Clases utilizadas:

- **Menú:** Enumera los alimentos disponibles, junto con sus precios.
- **Almacen:** Verifica y actualiza el inventario de ingredientes.
- **PedidoItem:** Representa cada elemento del pedido (producto y cantidad).
- **Utilidad:** Ofrece métodos auxiliares como el formateo de precios.

2. Métodos utilizados:

- Mostrar menú (Menu.values()).
- Verificar disponibilidad (Almacen.verificarDisponibilidad()).
- Actualizar inventario (Almacen.actualizarInventario()).
- Calcular costo total y aplicar recargo (PedidoItem.getCantidad())
- Definir prioridad del domicilio (isDomicilioPrioritario())

3. Flujo:

- **Selección de alimentos:** Muestra el menú disponible y permite al cliente seleccionar alimentos.
- **Verificación de disponibilidad:** Comprueba si hay suficientes ingredientes en el almacén para el alimento seleccionado.
- **Cantidad:** Solicita la cantidad deseada de cada alimento, asegurándose de que sea un número válido.
- **Actualizar inventario:** Reduce la cantidad de ingredientes disponibles en el almacén según el pedido.

- **Cálculo del costo total:** Calcula el costo del pedido considerando los precios y las cantidades seleccionadas.
- **Recargo por prioridad:** Si el pedido es prioritario, añade un recargo del 5%.

4. Resumen: Esta interacción se encarga de seguir el flujo con la funcionalidad anterior y darle la posibilidad al cliente de crear su domicilio como sea de su preferencia además de permitirle solicitar que el que el domicilio sea prioritario, el cual cambia el funcionamiento de la siguiente interacción.

Interacción 3: Procesar el pago y el cambio

1. Clases utilizadas:

- **Domiciliario:** Representa a los encargados de entregar los pedidos y calcular el cambio.
- **Domicilio:** Representa el pedido final, incluyendo cliente, alimentos, dirección, costo y domiciliario.
- **Utilidad:** Se utiliza para formatear precios y mostrar datos al cliente.

2. Métodos utilizados:

- Selección de domiciliario (`Domiciliario.getListaDomiciliarios()`).
- Creación de domicilio (`new Domicilio()`).
- Calcular cambio (`Domiciliario.entregarCambio()`).

2. Flujo:

- **Crear el pedido:** Genera un objeto Domicilio que incluye los datos del cliente, los alimentos seleccionados, la dirección, la prioridad y el costo total.
- **Validar el pago:** Solicita al cliente el monto con el que desea pagar, verificando que sea suficiente para cubrir el costo total.
- **Calcular y entregar el cambio:** Calcula el cambio y utiliza el método del domiciliario para determinar los billetes a entregar al cliente.
- **Confirmar la entrega:** Muestra un resumen del pedido, incluyendo la información del cliente, el pedido, el costo total y los datos del domiciliario que hará la entrega.

3. Resumen: Esta funcionalidad recopila toda la información obtenida de las dos primeras interacciones y genera el domicilio del cliente, se le solicita al cliente cual es la cantidad que piensa pagar para comprobar la disponibilidad que tiene el domiciliario de devolver el dinero necesario y así terminar la funcionalidad.

Funcionalidad 3 (Creación de Pedido):

Objetivos:

- Permitir que un cliente con una reserva existente seleccione los platos del menú que desea ordenar
- Hacer la recolección de los datos necesarios para el posterior cálculo del total del pedido, incluyendo la aplicación de descuentos personalizados.

(**requisitos previos:** en la ventana de inicio haber dado click en el botón “Ingresar”, que permite abrir la ventana principal del usuario, y luego en esta ventana seleccionar en la zona de menús la opción “Procesos y Consultas” y luego seleccionar “Realizar el pedido de mi reserva”)

interacción 1:

- Primero se configuran las etiquetas (labelv1 y labelv2) para mostrar instrucciones en la interfaz gráfica. En este caso particular en labelv1 se muestra: "Realizar el pedido de mi reserva" y en labelv2: "Desde esta ventana podrá seleccionar los platos y las cantidades que desee ordenar para su mesa, por favor ingrese todos los datos que se le solicitan continuación"
- En la zona de interacción del usuario se utiliza fieldFrameParaIdentificacion, que es una instancia de la clase **FieldFrame**, para crear un nuevo marco con configuraciones específicas, en este caso para mostrar un formulario donde se le pide al usuario que “ingrese el número de Identificación asociado a su reserva", esto con el fin de **confirmar la existencia de la reserva** antes de continuar, el botón limpiar campos y un botón “Verificar”
 - Cuando el usuario le da clic al botón “Verificar” se llama a la función buscarReserva(fieldFrameParaIdentificacion), aquí lo primero que se procede a hacer es verificar que el campo no este vacío, y en caso tal se interactúa con la clase **EntryVacio** para mostrar una ventana emergente con el error.
 - Si el usuario da clic en “Verificar” y el campo no está vacío se verifica que la entrada sea válida, en este caso que sea un número, en caso de que no se interactúa con la clase **IdEntryNoNumerico** para mostrar una ventana emergente con el error.
 - Si el usuario da clic en “Verificar” y la entrada cumple los requisitos se procede a hacer la **verificación de la existencia de la reserva**. Se utiliza el id ingresado como argumento para llamar el método buscar_en_lista_reservas(id) de la clase **restaurante** que busca si el id existe como atributo de alguno de los objetos de una lista de la clase **Cliente** donde se guardan los clientes que crean reservas.
 - Si no se encuentra el id, el método retorna False y se procede a interactuar con la clase **IdNoEncontrado(id)** para mostrar una ventana emergente con el mensaje de error
 - Si se encuentra el id (el método retornó True) el id ingresado se pasa como argumento al método **retornar_cliente(id) de la clase**

restaurante que busca y retorna el **objeto de tipo cliente** que tenga el id ingresado como atributo, y este objeto cliente se utiliza posteriormente para llevar varios procesos a cabo como:

- Analizar el historial de reservas del cliente para determinar su número de visitas en el restaurante y sumarle la visita actual. Esto se hace llamando al método `determinar_descuentos(cliente)` de la clase **restaurante**.

Basado en el comportamiento de visitas del cliente, El restaurante asigna descuentos por fidelidad, que corresponde a un descuento del 10% sobre el total de los platos pedidos por el cliente cuando se realicen 5 visitas y uno del 15% cuando se realicen 10 visitas al restaurante (cuando se obtiene este descuento, se reinicia la cuenta de las visitas).

- Este cliente también interactúa con la clase **pedido** ya que se utiliza para crear una instancia de esa clase y ser asignado como el atributo titular del pedido, y también asignarle el atributo pedido a la mesa asociada.

luego de que la reserva sea verificada se llama al método `saludar del cliente` para mostrar en `labelv2` un mensaje de bienvenida al cliente, informarle su número de visitas y si se le aplica algún descuento y mostrar en la zona de interacción el botón “Continuar con la creación de mi pedido” que al darle clic llama al método `FieldFrameParaCrearPedido`

interacción 2:

En `FieldFrameParaCrearPedido` proceden a interactuar la clase **Menu** (la cual contiene la información correspondiente de todos los platos: nombre, precio, lista de ingredientes y alérgenos) y **Almacen** (esta se encarga de aplicar los métodos correspondientes para determinar si hay inventario disponible para preparar determinado plato del menú) para crear las listas `listaDePlatos` y `unidadesDisponibles`, donde se guardan respectivamente los nombres de los platos disponibles para ser preparados y la cantidad máxima de unidades que se pueden preparar de dicho plato.

En la zona de interacción, con ayuda de la clase **FieldFrame**, se va a mostrar el mensaje "A continuación seleccione el nombre del plato y la cantidad de unidades que desea pedir" junto con dos listas desplegables (plato, unidades a pedir) que contienen los platos y las respectivas cantidades que el usuario puede pedir. También se muestra el botón “Agregar plato” que al darle clic llama al método `agregarPlatoAlResumen`, el cual primeramente verifica que se haya seleccionado una opción válida de las listas desplegables (que no se haya no seleccionado nada), en caso de que no, se interactúa con la clase **BoxVacio** para mostrar el mensaje de error. Si la selección fue válida se interactúa con la clase **Almacen** para actualizar las unidades del plato (restar las unidades pedidas) y en la ventana de interacción, con ayuda de la clase `FieldFrame`, se muestra el mensaje "{unidades} unidades de {nombrePlato} añadidas con éxito, ¿Qué desea hacer ahora?" y los botones "Continuar Pidiendo" y "Finalizar Pedido"

- Si se selecciona "Continuar Pidiendo" se vuelve a llamar el método `FieldFrameParaCrearPedido` y se repite todo el proceso descrito

- Si se selecciona “finalizar Pedido”:
 - **opción 1:** si la mesa asociada a la reserva **NO** es de tipo deluxe:

Se llama al método `frameParaMostrarResumen`, el cual en la ventana de interacción, con ayuda de la clase **FieldFrame**, muestra el mensaje "¡Gracias {nombre del cliente}, Su pedido ha sido creado con éxito!", y los botones "Ver resumen" y "Salir".

- Si se selecciona “Ver resumen” :
 - se llama al método `mostrarResumen` el cual a su vez llama al método `mostrarResumenPedido` que se encarga de calcular el valor total del pedido, determinar los valores de los descuentos (si aplica) y los cobros adicionales asociados a la reserva (si aplica) (estos valores son asignados como atributos a la instancia **factura** asociada a la **reserva**). Luego **compila** todos los elementos del pedido (platos seleccionados y cantidades, precio unitario, precio total, plato de cortesía (si aplica), información de los descuentos (si aplica), cobros adicionales asociados a la reserva (si aplica) e imprime toda la información del resumen, el cual se muestra en una ventana emergente aparte.

Nota: mientras no se presione el botón “Salir” el resumen se puede mostrar todas las veces que se quiera, presionando nuevamente el botón “Ver Resumen”

- Si se selecciona “Salir”:
 - Se cierra la ventana y se hace la serialización
- **opción 2:** si la mesa asociada a la reserva es de tipo deluxe:

Interacción 3:

Determinaremos el tipo de mesa de la reserva, y teniendo en cuenta la política del restaurante, ofreceremos o no un plato de cortesía dependiendo el tipo de la mesa que se esté atendiendo.

Para determinar si al cliente se le debe ofrecer un plato de cortesía, se llama al método `obtener_tipo_mesa(reservaAsociada)` de la clase **restaurante** que determina si el objeto de tipo **mesa** asociado a la **reserva** es de tipo deluxe, el método retorna `True` o `False` y se asigna este valor a la variable `OfrecerCortesia`.

- En caso de que `OfrecerCortesia == False`:
 - se procede llamar al método `frameParaMostrarResumen` para que ejecute las acciones ya descritas para método y mostrar el resumen del pedido.
- En caso de que `OfrecerCortesia == True`:
 - Si la mesa es deluxe, se ofrece un plato de cortesía, entonces se interactúa con la clase **menuCortesias** para mostrar en la zona de interacción, con ayuda de la clase **FieldFrame**, el mensaje "¡Su reserva con mesa tipo deluxe incluye un plato de cortesía!", junto con una lista desplegable que contienen las opciones de platos de cortesía y el botón Guardar selección que al darle clic llama al método `guardarCortesia`, el cual primeramente verifica que se haya seleccionado una opción válida de la lista desplegable (que no se haya no seleccionado nada), en caso de que no, se interactúa con la clase **BoxVacio** para mostrar el mensaje de error.

En caso de que la selección sea válida se procede a asignar el plato de cortesía al pedido asociado a la reserva y a llamar al método `frameParaMostrarResumen` para que ejecute las acciones ya descritas para método y mostrar el resumen del pedido.

Funcionalidad 4 (Gestión de Recompensas):

Interacción 1: Se realiza el registro de puntos: interactúan la clase factura con la clase restaurante

una vez finalizado el pedido y el proceso de pago se realiza el cálculo de puntos este puede variar de acuerdo al tipo de pedido, volumen, y frecuencia se actualizan y se almacenan en la base de datos del restaurante la forma de redimir los puntos se ve afectada de acuerdo a la cantidad a la que pertenezcan en seguida se le presentan las diferentes formas de redimir los puntos acumulados hasta el momento .

Restaurante-cliente: asignación de puntos por frecuencia

Conteo de Visitas: Se realiza una consulta en la base de datos para contar cuántas veces ha visitado el restaurante el cliente dentro del rango especificado.

Criterios de Acumulación: Se establecen criterios específicos para la acumulación de puntos, que pueden incluir:

- Número mínimo de visitas requeridas para empezar a acumular puntos.
- Escalas que determinan cuántos puntos se otorgan por cada visita adicional
- Si el cliente ha alcanzado el umbral mínimo de visitas, se calcula la cantidad total de puntos acumulados por frecuencia.
- Los puntos se suman al saldo actual del cliente.
- Actualización en la Base de Datos: El sistema actualiza automáticamente el registro de puntos del cliente en la base de datos, reflejando la nueva cantidad total después del conteo.

cliente-factura :se asignan puntos a los clientes en función del monto total de su factura:

- Verificación del Monto Total: El sistema verifica el monto total de la factura contra los rangos establecidos

Segunda interacción: gestión de recompensa por mal servicio:

Devolución del producto:

- Iniciación del Proceso: El cliente solicita la devolución del pedido, proporcionando detalles sobre el motivo (producto defectuoso, insatisfacción, etc.).
- Condiciones del Producto: El sistema verifica el estado del producto devuelto para determinar si es elegible para reembolso o ajuste en la deuda.
- si es elegible el sistema le presenta opciones de recompensa:
 1. Reembolso Directo: Si el cliente prefiere un reembolso, se le ofrece un porcentaje del monto total pagado, considerando su fidelidad.
 2. Puntos Adicionales: Se pueden otorgar puntos adicionales como compensación por la devolución, incentivando futuras compras.
 3. Descuentos en Futuras Compras: Ofrecer un cupón o descuento en su próxima compra como gesto de agradecimiento por su lealtad.

4. Crédito en Restaurante: Permitir que el cliente utilice el monto devuelto como crédito para futuras compras.

Mal servicio:

Se centra en preguntar al cliente qué dificultades presentó y se procede a:

- Clasificación por Tipo de Servicio: Se clasifica el tipo de servicio que ha sido mal evaluado, mesero, pedido, factura, reserva
- El cliente recibe la notificación y tiene la opción de aceptar la recompensa ofrecida.
- Si el cliente acepta, se confirma automáticamente la recompensa en su cuenta y se registra el tipo de recompensa para que el cliente pueda hacer uso de ella:

tercera interacción: Canjeo de puntos

- los puntos se utilizan para una reserva

Se usan los puntos actualizados que son acumulables SOLO a partir de la frecuencia

El sistema verifica si el número de puntos solicitados por el Cliente es suficiente para cubrir parte del costo del nuevo reserva y si es susceptible a recibir este tipo de descuento

1. Actualización del Saldo de Puntos:
 - Si la redención es válida, se deducen los puntos utilizados del saldo total del Cliente, actualizando su estado.
2. Confirmación al Cliente:
 - Finalmente, se notifica al Cliente sobre la redención exitosa y se le proporciona un resumen del nuevo saldo de puntos, así como detalles sobre su pedido y factura actualizada.

-los puntos se utilizan para productos específicos

1. Selección del Producto El cliente elige un producto específico del menú que desea comprar y verifica si puede utilizar sus puntos para obtener un descuento.
2. el sistema verifica si la cantidad de puntos es susceptible para la elección, mostrándole al cliente qué productos puede elegir del menú con la cantidad de puntos correspondiente a cada producto
3. Se actualizan los puntos
4. Confirmación al Cliente: Finalmente, se notifica al Cliente sobre la redención exitosa y se le proporciona un resumen del nuevo saldo de puntos

Los puntos se utilizan para servicios exclusivos

1. el cliente puede acceder a servicios como mesas deluxe
2. Dependiendo de la capacidad de la mesa se le muestra la cantidad que necesita, la mesa con mayor capacidad será la que mayor cantidad de puntos acumulados necesita y así en orden descendente.
3. Verificación de Disponibilidad: El sistema verifica y si el cliente tiene suficientes puntos para cubrir el servicio elegido

4. Si el descuento es válido, se deducen los puntos utilizados del saldo total del cliente.
5. Se notifica al cliente

Funcionalidad 5 (Calificación del servicio):

Nota: Los casos por consumo local o consumo a domicilio difieren en: El sistema para calificar (la encuesta no es la misma). Su apartado exclusivo (en consumo local se accede al apartado exclusivo cuando el cliente tiene una mesa tipo deluxe, en cambio con consumo a domicilio se accede cuando el cliente tiene un domicilio prioritario), en caso de que no tenga acceso a un apartado exclusivo imprime un mensaje donde se le da a conocer la posibilidad de tener uno. La calificación afecta al mesero o al domiciliario dependiendo del caso.

Interacción 1: **El cliente califica el servicio.**

Interactúan la clase Cliente con la clase Calificación

El cliente, al final de la comida, califica su experiencia general. La calificación puede involucrar diferentes criterios que se llevarán a cabo mediante un pequeño formulario donde cada pregunta se califica del 1 al 5 donde, se preguntará respecto a: calidad de la comida, amabilidad y calidad del mesero, tiempo de espera, etc. O en caso contrario de consumo a domicilio, se exonera la calidad del mesero.

La calificación es almacenada y contiene la retroalimentación y comentarios del cliente.

Caso por consumo local:

Interactúan la clase Mesero con la clase Calificación

La calificación del cliente se utiliza para evaluar el desempeño del mesero. Dependiendo de la puntuación, el mesero puede recibir retroalimentación sobre su servicio. El desempeño del mesero se actualiza tomando en cuenta su calificación actual y la nueva que tendrá ahora.

Interactúan la clase Pedido con la clase Calificación

El objeto de tipo Calificación creado se asocia al objeto de tipo Pedido. El tiempo de espera del pedido puede influir en la calificación del servicio. Si el pedido es entregado tarde, el cliente puede puntuar más bajo. La calificación refleja este aspecto, afectando la percepción general del servicio y modificando con base en el promedio actual la percepción general del restaurante.

Caso por consumo a domicilio:

Interactúan la clase Domiciliario con la clase Calificación

La calificación del cliente se utiliza para evaluar el desempeño del domiciliario. Dependiendo de la puntuación, el domiciliario puede recibir retroalimentación sobre su servicio. El desempeño del domiciliario se actualiza tomando en cuenta su calificación actual y la nueva que tendrá ahora.

Interactúan la clase Domicilio con la clase Calificación

El objeto de tipo Calificación creado se asocia al objeto de tipo Domicilio. El tiempo de espera del pedido puede influir en la calificación del servicio. Si el pedido es entregado tarde, el cliente puede puntuar más bajo. La calificación refleja este aspecto, afectando la percepción general del servicio y modificando con base en el promedio actual la percepción general del restaurante.

Interacción 2: La calificación afecta la factura final

Caso por consumo local:

Interactúan la clase Factura con la clase Calificación

Al final de la calificación el cliente recibe la factura. La calificación impacta en la factura, por ejemplo, un cliente que otorga una calificación muy baja podría recibir un descuento dependiendo de cuan bajo sea y se acomodará el valor de la factura y se imprime el objeto de tipo Factura.

Interactúan la clase Cliente con la clase Factura

El cliente paga la factura, la cual incluye la calificación del servicio. El cliente realiza el pago y, al mismo tiempo, la calificación se integra como parte de la experiencia completa del servicio. La calificación no solo se registra en el sistema, sino que se hace visible en la factura final como parte de la experiencia del cliente. Por ejemplo, al interior de la factura, podría aparecer un campo donde se explica el puntaje que el cliente dejó (ej., "Calificación del servicio: 4/5").

Interactúan la clase Mesero con la clase Factura

El mesero recibe la información de la calificación y la factura asociada con su servicio, lo que puede influir en las estadísticas de su desempeño. Por ejemplo, si su desempeño por las calificaciones baja, también bajará su prioridad para ser escogido para futuras reservas.

Caso por consumo a domicilio:

Interactúan la clase Domicilio con la clase Calificación

Al final de la calificación el cliente recibe el monto a pagar. La calificación por medio de interactuar con el atributo costoEnvio de la clase Domicilio, impacta en el total, por ejemplo, un cliente que otorga una calificación muy baja podría recibir un descuento dependiendo de cuan bajo sea y se acomodará el valor de la factura y se imprime el resumen del pedido.

Interactúan la clase Cliente con la clase Domiciliario

El cliente paga el pedido, la cual incluye la calificación del servicio. El cliente realiza el pago y, al mismo tiempo, la calificación se integra como parte de la experiencia completa del servicio. La calificación no solo se registra en el sistema, sino que se

hace visible en la factura final como parte de la experiencia del cliente. Por ejemplo, al interior de la factura, podría aparecer un campo donde se explica el puntaje que el cliente dejó (ej., "Calificación del servicio: 3/5").

Nota: En consumo a domicilio no se afecta la prioridad para ser escogido el domiciliario ya que esto dependerá del tipo de domicilio (prioritario o no).

Interacción 3: **Reputación del restaurante basada en la calificación**

Interactúan la clase calificación con la clase Restaurante

La calificación final del cliente se utiliza para calcular la reputación general del restaurante. Si muchas calificaciones son bajas, la reputación del restaurante podría bajar, lo que afecta su imagen. La calificación de los clientes se utiliza para modificar la reputación general del restaurante.

Caso por consumo local:

Interactúan la clase Pedido con la clase Restaurante

La calidad del pedido, el tiempo de espera se registra para ver cómo afecta las calificaciones generales del restaurante. Si los pedidos suelen llegar tarde, la calificación será más baja afectando o mejorando la calificación del restaurante dado el caso.

Interactúan la clase Mesero con la clase Restaurante

Las calificaciones de los meseros se envían al restaurante, quien puede tomar decisiones sobre la formación o reestructuración del equipo. Si un mesero recibe múltiples calificaciones negativas (últimas 3 calificaciones por debajo o igual 2) puede ser despedido.

Caso por consumo a domicilio:

Interactúan la clase Domicilio con la clase Restaurante

Según el tiempo de espera, se registra para ver cómo afecta las calificaciones generales del restaurante. Si los pedidos suelen llegar tarde, la calificación será más baja afectando o mejorando la calificación del restaurante dado el caso.

6. Manejo de Excepciones

ErrorAplicacion: Indica un error ocurrido durante la ejecución de la aplicación

Captura:

ValidacionDatos: Ocurre cuando los datos que se ingresan no son los correctos o no se encuentran registrados

```
except ValueError as e:  
    messagebox.showerror("Error de validación", str(e))
```

Captura:

IdNoEncontrado: Ocurre cuando no se encuentra el id en la base de datos del restaurante

Captura:

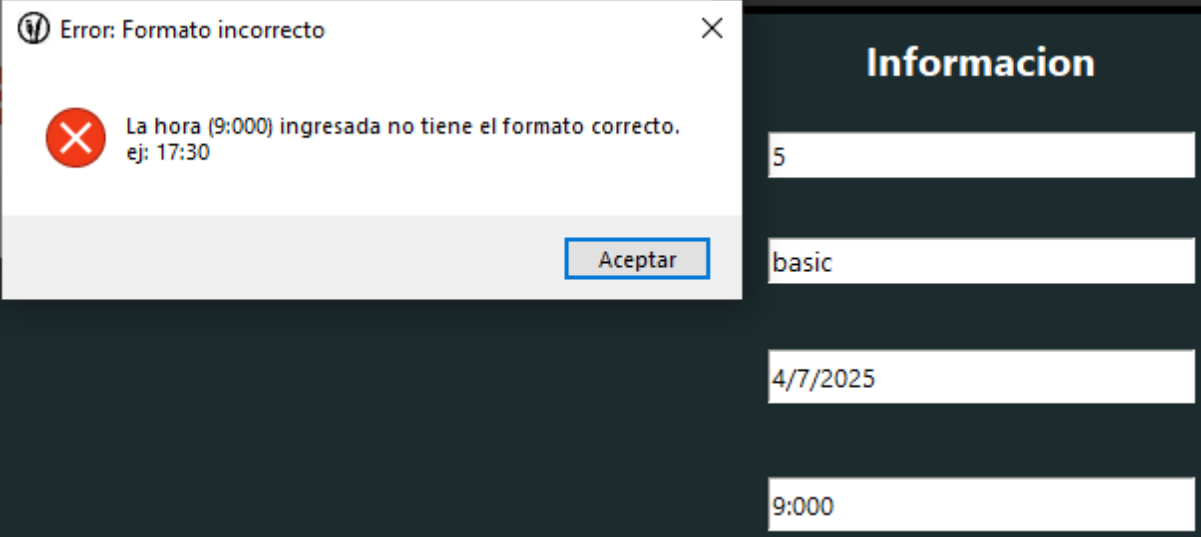
```
class IdNoEncontrado(ValidacionDatos):  
    def __init__(self, id):  
        self.mensaje_error_valor = f"la identificación: {id} no se encuentra en el sistema."  
        messagebox.showerror("Error: Identificación no encontrada", self.mensaje_error_valor)  
        super().__init__(self.mensaje_error_valor)
```

The screenshot shows a web application interface with a dark theme. At the top, there's a header with "Confirmación de existencia de reserva" on the left and "CC/CE" on the right. Below the header, the main content area says "Ingrese el número de Identificación asociado a su reserva". To the right of this text is a text input field containing the number "700". Below the input field is a button labeled "Verificar". To the right of the "Verificar" button, an error message dialog box is displayed. The dialog box has a title bar that says "Error: Identificación no encontrada" and a close button (X). The main content of the dialog box shows a red circle with a white 'X' icon, followed by the text "la identificación: 700 no se encuentra en el sistema." At the bottom right of the dialog box is a button labeled "Aceptar".

```
1542 | | | else:  
1543 | | |     raise IdNoEncontrado(id) #cuando la id ingresada no tenga reserva
```

HoraInvalida: Ocurre cuando la hora que se ingresa no está en el formato correcto.

Captura:

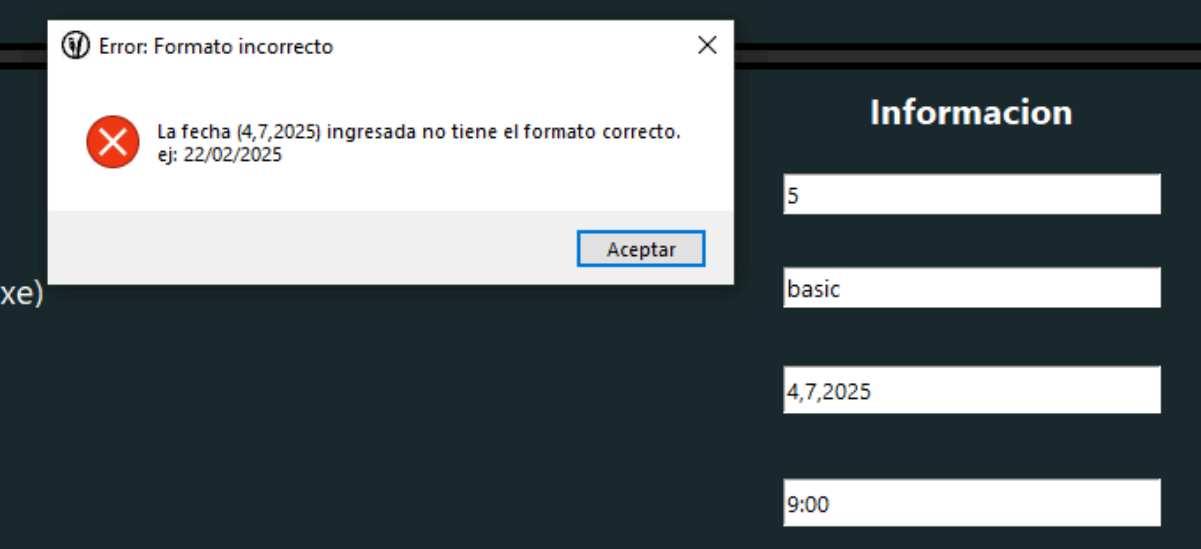


The screenshot shows a web form titled "Informacion" with four input fields. The first field contains the number "5", the second contains the text "basic", the third contains the date "4/7/2025", and the fourth contains the time "9:000". An error dialog box is open in the foreground, displaying the message: "Error: Formato incorrecto. La hora (9:000) ingresada no tiene el formato correcto. ej: 17:30". The dialog box has a red 'X' icon and an "Aceptar" button.

```
503         if not restaurante.validar_hora(horaReserva):
504             raise HoraInvalida(horaReserva)
```

FechaInvalida: Ocurre cuando la fecha ingresada no se encuentra con el formato adecuado para que pueda ser ejecutada en el sistema.

Captura:

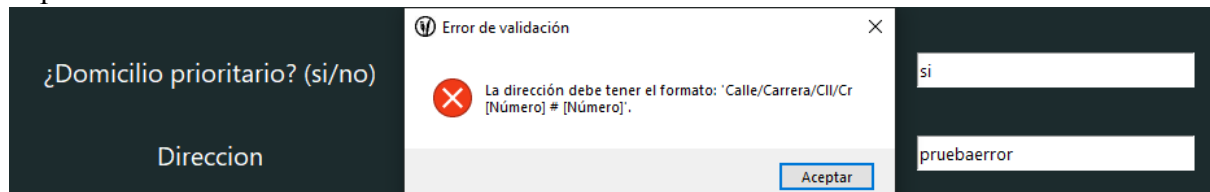


The screenshot shows a web form titled "Informacion" with four input fields. The first field contains the number "5", the second contains the text "basic", the third contains the date "4,7,2025", and the fourth contains the time "9:00". An error dialog box is open in the foreground, displaying the message: "Error: Formato incorrecto. La fecha (4,7,2025) ingresada no tiene el formato correcto. ej: 22/02/2025". The dialog box has a red 'X' icon and an "Aceptar" button.

```
501         if not restaurante.validar_fecha(fecha):
502             raise FechaInvalida(fecha)
```

DireccionErronea: Ocurre cuando la dirección que se ingresa no tiene el formato adecuado

Captura:

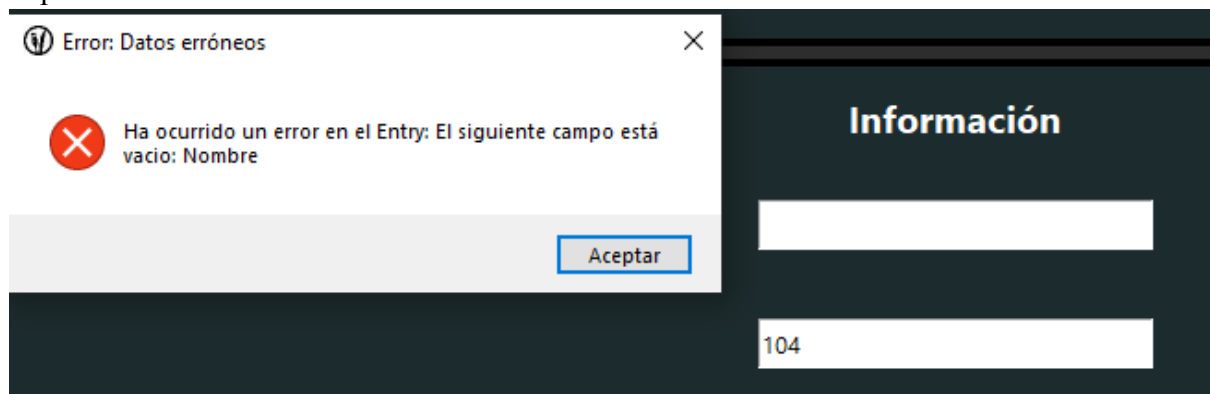


The screenshot shows a dark-themed application window. On the left, a label "¿Domicilio prioritario? (si/no)" is above a "Direccion" button. On the right, there are two text input fields; the first contains "si" and the second contains "pruebaerror". A white dialog box titled "Error de validación" is centered, displaying a red 'X' icon and the message: "La dirección debe tener el formato: 'Calle/Carrera/Cll/Cr [Número] # [Número]'." An "Aceptar" button is at the bottom right of the dialog.

```
# Validar Dirección (Formato: "Calle/Carrera/Cll/Cr [Número] # [Número]")
patron_direccion = r"^(Calle|Carrera|Cll|Cr) \d+ # \d+$"
if not re.match(patron_direccion, direccion.strip(), re.IGNORECASE):
    raise ValueError("La dirección debe tener el formato: 'Calle/Carrera/Cll/Cr [Número] # [Número]'")
```

DatosEntrada: Ocurre cuando pasa algún error con los entry.

Captura:



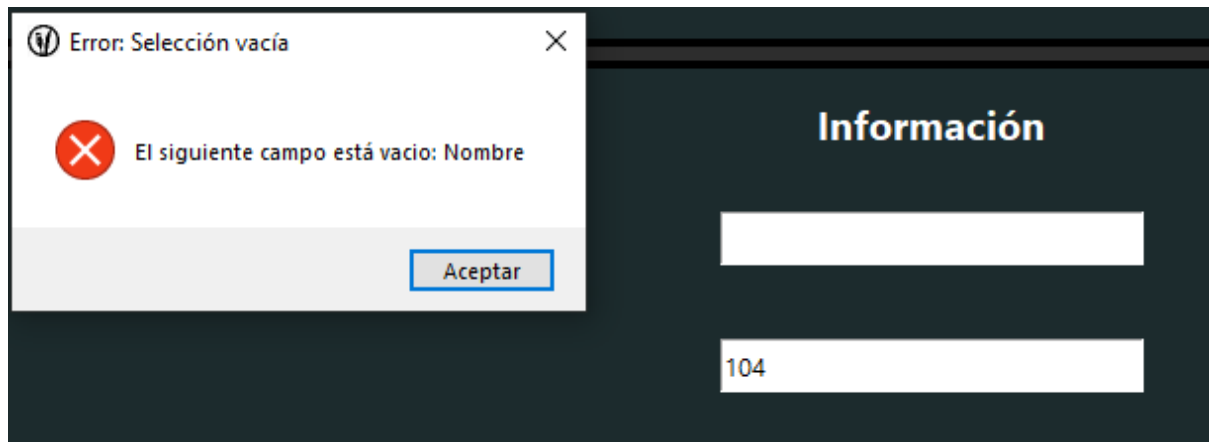
The screenshot shows a dark-themed application window. On the right, a section titled "Información" contains two text input fields; the second field contains the number "104". A white dialog box titled "Error: Datos erróneos" is centered, displaying a red 'X' icon and the message: "Ha ocurrido un error en el Entry: El siguiente campo está vacío: Nombre". An "Aceptar" button is at the bottom right of the dialog.

Datos Entrada es usado como clase padre de EntryVacio

```
56 class EntryVacio(DatosEntrada):
57     def __init__(self, entryVacio):
58         self.mensaje_error = f"El siguiente campo está vacío: {entryVacio}"
59         messagebox.showerror("Error: Selección vacía", self.mensaje_error)
60         super().__init__(self.mensaje_error)
```


EntryVacio: Ocurre cuando al entry no se le ingresan datos, es decir, está vacío

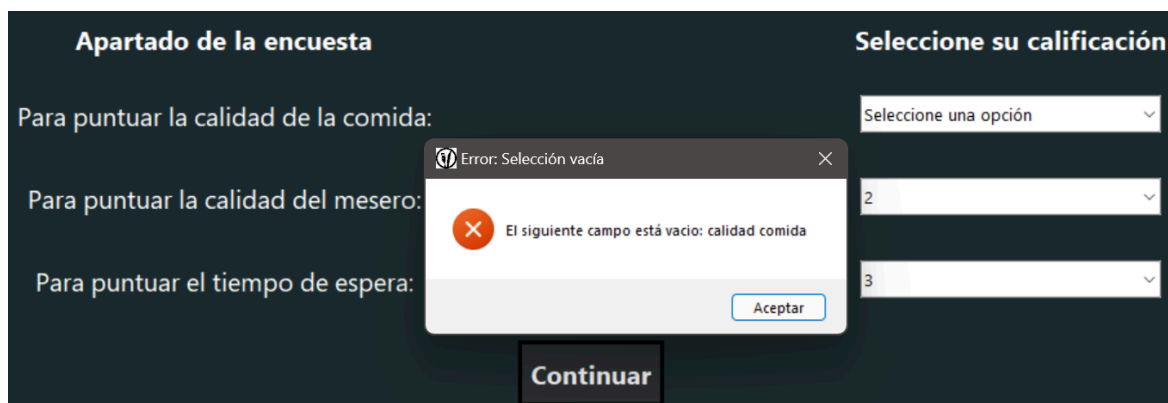
Captura:



```
528     def segundo_fieldFrame(fieldFrame1):
529         informacion1 = fieldFrame1.obtener_datos()
530         try:
531             if not informacion1[0]:
532                 print("Entry vacío")
533                 raise EntryVacio("Nombre")
534             if not informacion1[1]:
535                 raise EntryVacio("Identificacion")
```

BoxVacio: Ocurre cuando al combobox no se le selecciona valores

Captura:



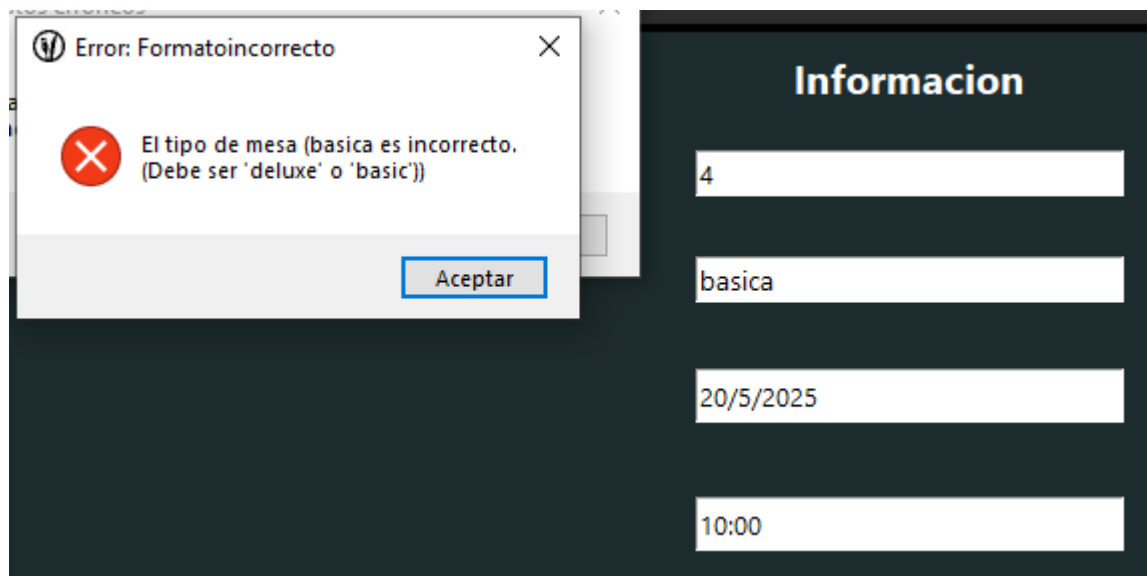
```

1059     try:
1060         calidadComida = int(labelv3.obtener_datos()[0])
1061     except:
1062         raise BoxVacio('calidad comida')
1063
1064     try:
1065         calidadComida = int(labelv3.obtener_datos()[1])
1066     except:
1067         raise BoxVacio('calidad mesero')
1068
1069     if len(labelv3.obtener_datos()) == 3:
1070         try:
1071             calidadComida = int(labelv3.obtener_datos()[2])
1072         except:
1073             raise BoxVacio('tiempo espera')

```

ExcepcionTipoMesa: Ocurre cuando se ingresa en el campo entry un dato diferente a los solicitados

Captura:



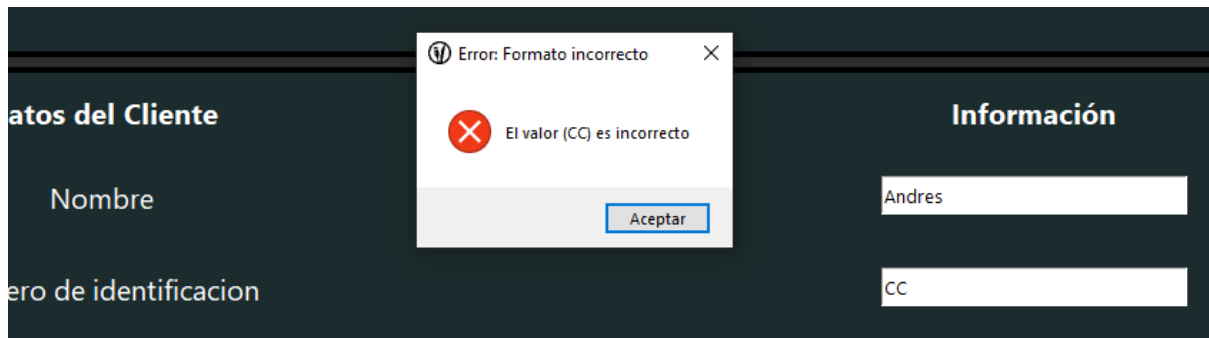
```

505     if tipoMesa not in ["basic","deluxe"]:
506         raise ExcepcionTipoMesa(tipoMesa)

```

IdEntryNoNumérico: Ocurre cuando se ingresan datos que no son exclusivamente números

Captura:



The screenshot shows a web form with two main sections: 'Datos del Cliente' on the left and 'Información' on the right. In the 'Datos del Cliente' section, there are input fields for 'Nombre' and 'Número de identificación'. In the 'Información' section, there are input fields for 'Andres' and 'CC'. A modal error message is displayed in the center, titled 'Error: Formato incorrecto'. The message contains a red 'X' icon and the text 'El valor (CC) es incorrecto'. There is an 'Aceptar' button at the bottom of the modal.

```
543         except(ValueError):  
544             IdEntryNoNumerico(informacion1[1])  
545
```

7.FieldFrame

Descripción:

El fieldFrame de el programa posee un constructor, y diferentes métodos, explicados a continuación:

Constructor:

Recibe 9 parámetros cada uno con una función específica:

ventana: Indica en qué ventana o frame estará ubicado el fielFrame.(tipo frame o ventana)

tituloCriterios: Indica el título que llevará la columna de los criterios (tipo String)

criterios: Establece los nombres de los criterios, es decir, la información que se le solicitará al usuario (tipo lista de Strings)

tituloValores: Indica el título que llevará la columna de los valores, es decir, donde el usuario ingresará la información (tipo String)

valores: Aquí se ingresan los valores por defecto que se mostrarán al usuario. (tipo lista)

criteriosNoEditables: Indica los nombres de los criterios que no podrán ser modificados por el usuario.

tipo: Indica el tipo de fieldFrame que se creará, puede ser tipo 0,1,2,3; el tipo 0 habilita la columna de valores de tipo entry; el tipo 1 habilita la columna de valores como combobox, en este caso las opciones del combobox se pasarán como una lista al parámetro valores, Ej: combobox con opciones: “Rojo”, “Amarillo”, “Azul”, valores=[[“Rojo”, “Amarillo”, “Azul”]], y así para cada criterio, separando las listas con comas; el tipo 2 permite hacer preguntas de si o no; el tipo 3 sólo muestra texto.

comandoContinuar: Aquí se ingresa el nombre de la función que se debe ejecutar al presionar el botón “Continuar” que también guarda los datos ingresados en la columna valores.

comandoCancelar: Se encarga de hacer la función que se establezca cuando en el Fieldframe al boton se la da la opción de ‘NO’

Métodos:

def obtener_datos(): Devuelve una lista con la información ingresada por el usuario

def guardar_datos(): Guarda los datos que estén en la columna de valores, e imprime estos datos por consola.

def get_value(criterio): Devuelve el valor que tiene asignado el criterio que se ingresa como parámetro en esta función.

Capturas de Pantalla:

Aquí se puede ver claramente como se usa en 3 ocasiones el FieldFrame en la funcionalidad 1 para solicitar la información al usuario:

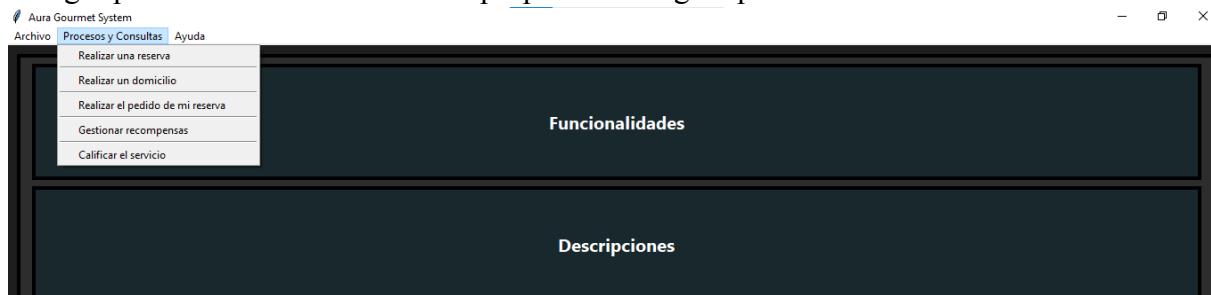
```
477     fieldFrame2.destroy()
478     frameMesas = FieldFrame(framev4,"Mesas disponibles",["Mesa"],"Numero de mesa",valores=[mesas],tipo=1,comandoContinuar=lambda: eleccion_m
479     frameMesas.grid(sticky="new")
480     print("Vamos bien")
481
482     def segundo_fieldFrame(fieldFrame1):
483         informacion1 = fieldFrame1.obtener_datos()
484         fieldFrame1.destroy()
485         fieldFrame2 = FieldFrame(framev4,"Datos de la Reserva",["Personas","Tipo de Mesa(basic o deluxe)","Fecha","Hora"],"Informacion",comandoC
486         fieldFrame2.grid(sticky="new")
487
488     labelv3.destroy()
489     fieldFrame1 = FieldFrame(framev4,"Datos del Cliente",["Nombre","Numero de identificacion"],"Información",comandoContinuar=lambda:segundo_fie
490     fieldFrame1.grid(sticky="new")
```

8.Manual de Usuario

El siguiente manual detallado le permitirá utilizar el sistema de manera adecuada (**por favor, leer con mucha atención**).

Al ejecutar el sistema lo primero que observará será la ventana principal, donde se encuentra una bienvenida a nuestro sistema, las hojas de vida y fotos de los desarrolladores(si desea visualizarlas de click sobre el cuadro donde dice “Hojas de vida (Click para cambiar)” y para cambiar de un desarrollador a otro de click nuevamente) , imágenes relacionadas con el restaurante(para cambiar de imagen pase el cursor sobre la imagen), y un botón que le permitirá acceder al sistema, llamado “Ingresar”.

Dé click en el botón “Ingresar”, este lo(a) dirigirá a una ventana con información importante, por favor, dé click en “Procesos y consultas”(ubicado en la parte superior izquierda, bajo el nombre del sistema), y elija la funcionalidad a la cual desea ingresar, para ayudarlo(a) a navegar por cada funcionalidad se le proporciona la guía para cada una a continuación:



Funcionalidad 1 (Realizar Reserva):

Inicio del proceso:

A form titled 'Inicio del proceso:' for the 'Realizar Reserva' functionality. It is divided into two main sections: 'Datos del Cliente' on the left and 'Información' on the right. Under 'Datos del Cliente', there are two labels: 'Nombre' and 'Numero de identificacion'. Under 'Información', there are two input fields. The first input field contains the text 'Andres'. The second input field contains the text '1234567'. At the bottom of the form, there are two buttons: 'Continuar' and 'Limpiar campos'.

Se mostrará una pantalla con campos para ingresar nombre y número de identificación sin comas o puntos(sólo el número)

Ingresar los datos y hacer clic en "Continuar".

Datos de la reserva:

Datos de la Reserva	Informacion
Personas	<input type="text" value="6"/>
Tipo de Mesa(basic o deluxe)	<input type="text" value="deluxe"/>
Fecha	<input type="text" value="24/2/2025"/>
Hora	<input type="text" value="12:00"/>
<div><div>Continuar</div><div>Limpiar campos</div></div>	

Ingresar la siguiente información: Cantidad de personas (mínimo 1, máximo 10), Tipo de mesa (basic o deluxe), Fecha de la reserva (Formato: dd/mm/aaaa), Hora de la reserva (Formato: hh:mm, dentro del horario del restaurante(7 y 24), en horario militar).

Hacer clic en "Continuar".

Elección de la mesa:

Mesas disponibles	Numero de mesa
Mesa	<div><div>Seleccione una opción</div><div>5</div></div>
<div>Continuar</div>	

Se mostrará una lista de mesas disponibles según la cantidad de personas y el tipo de mesa.

Seleccionar el número de mesa deseado y hacer clic en "Continuar".

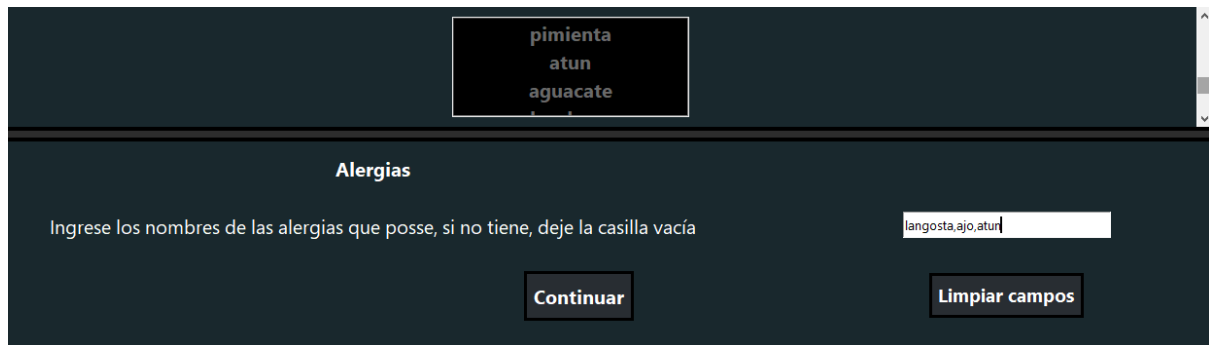
Personalización (Solo para mesas deluxe):

Personalizacion de la reserva	
Decoración de la mesa	<input type="text" value="Moderno"/>
¿Desea agregar 1 hora de permanencia en el restaurante?	<input type="text" value="No"/>
<div>Continuar</div>	

Si se eligió una mesa deluxe, se solicitará: Estilo de decoración (Ejemplo: Elegante, Rústico, Moderno), Hora adicional (Opcional).

Si se selecciona la hora adicional, el sistema verificará disponibilidad y agregará el costo adicional de 30,000.

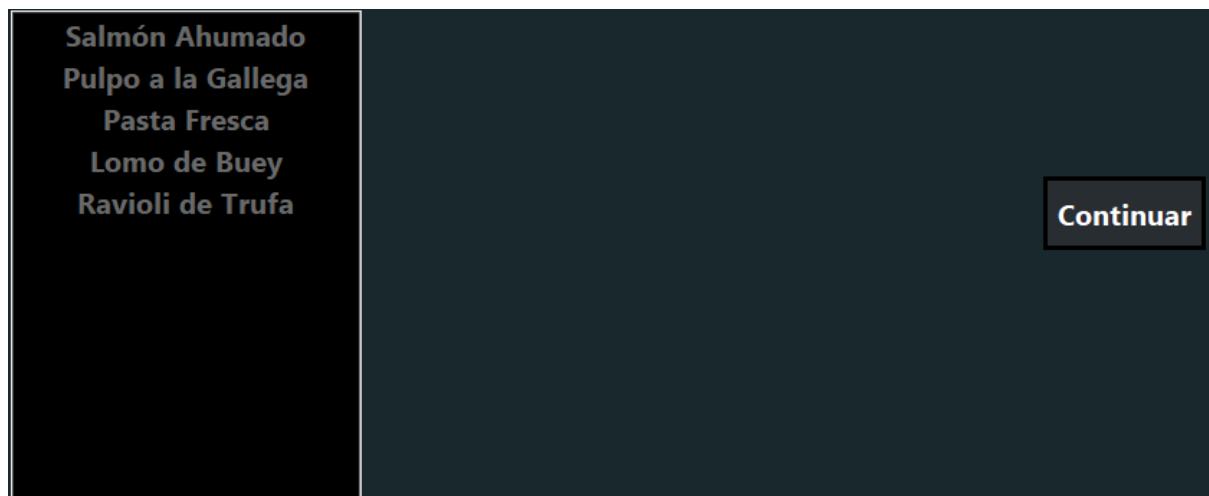
Ingreso de alergias (Solo para mesas deluxe):



The screenshot shows a dark-themed web form titled "Alergias". At the top, there is a scrollable list of ingredients: "pimienta", "atun", and "aguacate". Below the title, a text prompt reads "Ingrese los nombres de las alergias que posee, si no tiene, deje la casilla vacía". To the right of this prompt is a text input field containing the text "langosta, ajo, atun". At the bottom of the form, there are two buttons: "Continuar" and "Limpiar campos".

Ingresar los ingredientes que aparecen en la lista a los que el cliente es alérgico (separados por comas) y luego oprima “Continuar”

El sistema filtrará el menú y mostrará solo los platos que no contienen esos ingredientes, dé click en “Continuar”



The screenshot shows a dark-themed web interface. On the left, a scrollable list of menu items is displayed: "Salmón Ahumado", "Pulpo a la Gallega", "Pasta Fresca", "Lomo de Buey", and "Ravioli de Trufa". On the right side of the interface, there is a large button labeled "Continuar".

Confirmación de la reserva:

Se mostrará un resumen con todos los datos de la reserva:

- Nombre del cliente
- Número de identificación
- Número de personas
- Tipo de mesa
- Fecha y hora de la reserva
- Numero de la reserva (Generada automáticamente por el sistema)

- Personalización (si aplica)
- Alergias (si aplica)

Nombre: Andres

Identificación: 1234567

Número de personas: 6

Tipo de Mesa: deluxe

Fecha de Reserva: 2025-02-24 12:00:00

Número de reserva: RNK78

Decoración: Moderno

Hora Adicional: No

Alergias: ['langosta,ajo,atun']

Continuar

Limpiar campos

Si toda la información es correcta, hacer clic en "Continuar", que lo sacará del sistema.

Funcionalidad 2 (Realizar un Domicilio):

Al ingresar a esta funcionalidad se encontrará con la siguiente pantalla:

Datos del Cliente

Nombre

Número de identificación

¿Domicilio prioritario? (sí/no)

Direccion

Continuar

Información

Limpiar campos

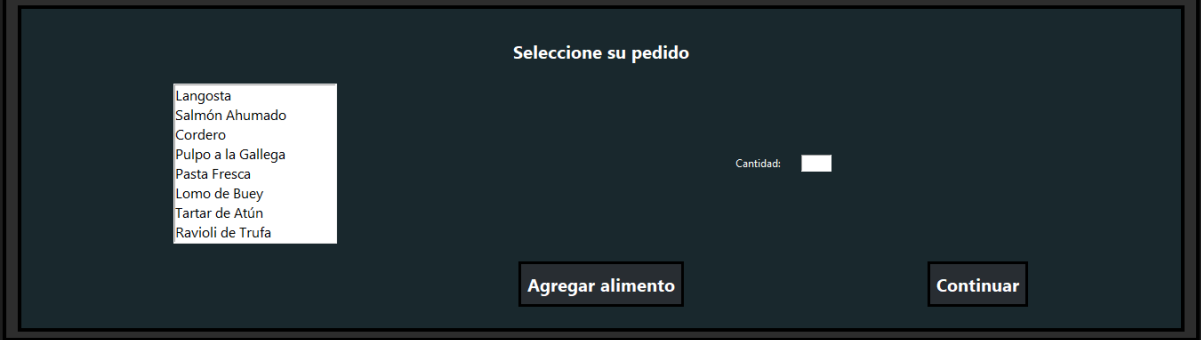
Deberá hacer click sobre los campos en blanco y rellenar los campos blancos en la parte de información, estas son algunas recomendaciones adicionales:

1. El nombre debe consistir meramente de letras
2. La identificación no debe contener puntos, comas, espacios o algun otro caracter especial, además debe tener una longitud de entre 8 a 12 dígitos para que sea considerada válida
3. En el apartado de domicilio prioritario sólo se aceptan las siguientes dos respuestas "sí" o "no".
4. En el apartado de Direccion, debe ingresar su dirección en el siguiente formato: calle/carrera/cll/cr [numero] # [numero]

ejemplo: cll 123 # 456

En caso de no acatar las recomendaciones, el sistema le lanzará un mensaje de error y le hará saber cual es el fallo en su registro, de ser ese el caso, seguir las indicaciones del mensaje.

Al completar el registro inicial con éxito podrá ver la siguiente ventana:



Seleccione su pedido

- Langosta
- Salmón Ahumado
- Cordero
- Pulpo a la Gallega
- Pasta Fresca
- Lomo de Buey
- Tartar de Atún
- Ravioli de Trufa


Cantidad:

Agregar alimento **Continuar**

1. Deberá seleccionar con click izquierdo en la lista de la izquierda el alimento que desea pedir a domicilio
2. Debe hacer click en la casilla de cantidad y digitar la cantidad que desea consumir.
3. Debe hacer click en el botón de “Agregar alimento” para procesar la elección de alimento.

podrá repetir el proceso las veces que desee hasta completar su domicilio, en el momento que desee no seguir agregando más alimentos y continuar con el proceso deberá darle click al botón de “Continuar”.

Se le mostrará el resumen de los datos ingresados, además de los platos pedidos con su respectivo precio.



Resumen del Domicilio

Nombre: mateo

Identificación: 123456789

Domicilio Prioritario: si

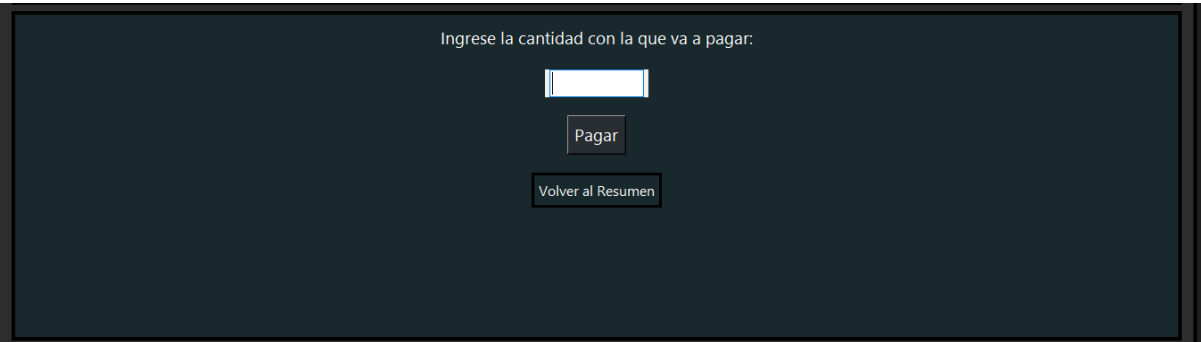
Platos pedidos:

- Pasta Fresca: 1 unidades (60000 COP)

****Precio Total: 60000 COP****

Proceder al Pago

Para proceder con el pago debera darle al botón de Proceder al pago



Ingrese la cantidad con la que va a pagar:

Pagar

Volver al Resumen

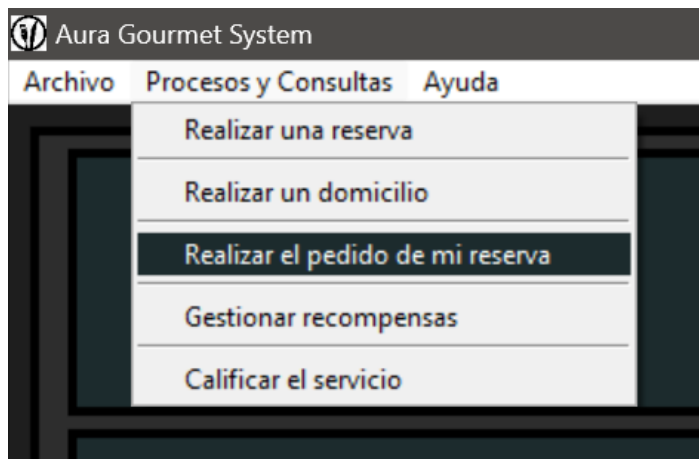
En la casilla en blanco deberá ingresar el valor con el que desea pagar a nuestro domiciliario por el domicilio, para así comprobar la disponibilidad que tiene para devolverle el dinero

exacto, Para por último darle al botón de pagar para procesar su pago. Si no está seguro de algún dato podrá volver al resumen desde el botón de “Volver al resumen”.

Funcionalidad 3 (Realizar el pedido de mi reserva):

1. acceder a la funcionalidad

Para acceder a ella, en las opciones de “Procesos y consultas” seleccione “Realizar el pedido de mi reserva”



2. verificación de reserva

Observará una ventana en la que se le pide que ingrese el número de identificación asociado a su reserva, introduzca la identificación y de clic al botón “Verificar”. Asegúrese de:

- No dejar el campo vacío
- Introducir un número de identificación en formato correcto (solo números)
- introducir una identificación a la que ya le haya creado una reserva previamente

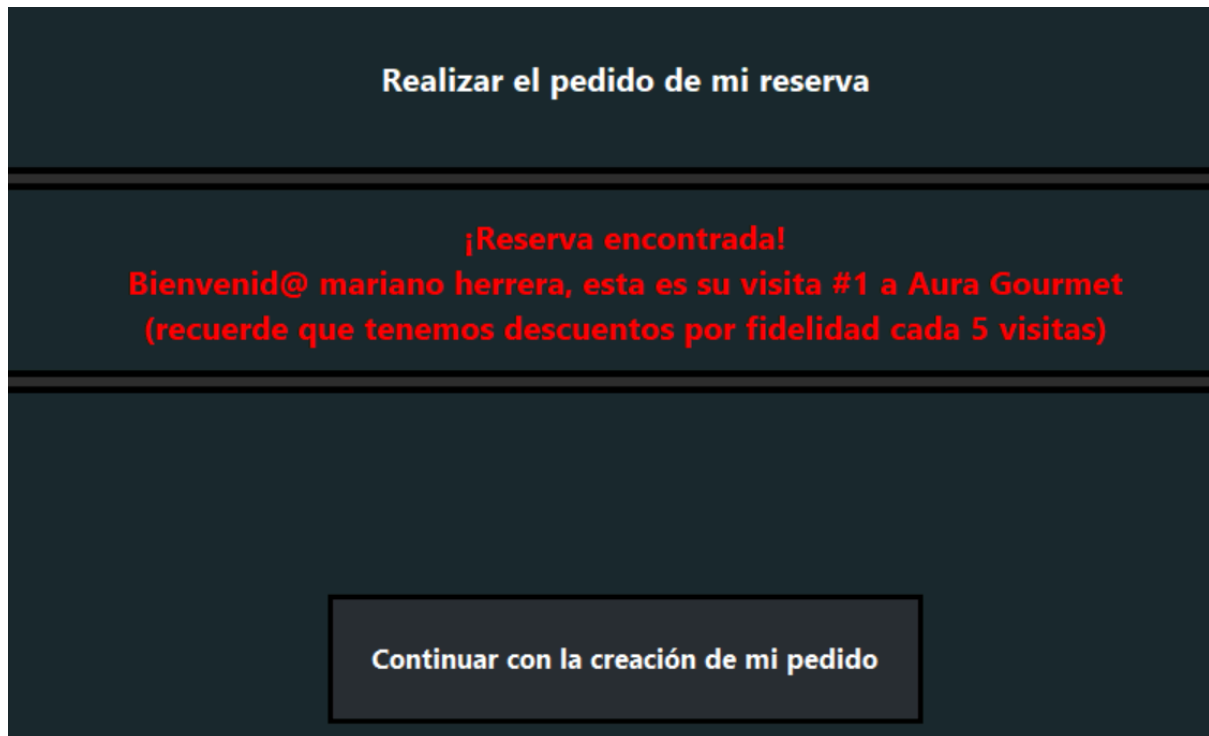
si no se cumple cualquiera de los requisitos anteriores obtendrá una ventana emergente de error, ciérrela e introduzca un dato correcto.

Nota: puede utilizar la identificación 2004 para probar la funcionalidad

A screenshot of a web form titled 'Realizar el pedido de mi reserva'. The form has a dark blue background with white text. It contains a message: 'Desde esta ventana podrá seleccionar los platos y las cantidades que desee ordenar para su mesa, por favor ingrese todos los datos que se le solicitan a continuación'. Below this, there is a section for 'Confirmación de existencia de reserva' with a label 'CC/CE'. It asks the user to 'Ingrese el número de Identificación asociado a su reserva' and provides a text input field containing the number '2004'. At the bottom, there are two buttons: 'Verificar' and 'Limpiar campos'.

3. Mensaje de bienvenida:

Luego de verificar su reserva de forma correcta observará un mensaje de bienvenida donde se le dará información de su número de visitas al restaurante y si se le aplicará algún descuento por fidelidad (Descuento solo se aplica para visitas múltiples de 5: 5, 10, 15...). De clic en el botón “Continuar con la creación de mi pedido”



The screenshot shows a dark-themed interface. At the top, a header bar contains the text "Realizar el pedido de mi reserva". Below this, a message in red text reads: "¡Reserva encontrada! Bienvenid@ mariano herrera, esta es su visita #1 a Aura Gourmet (recuerde que tenemos descuentos por fidelidad cada 5 visitas)". At the bottom, there is a button labeled "Continuar con la creación de mi pedido".

4. Añadir elementos al pedido

Luego, observará el mensaje “A continuación seleccione el nombre del plato y la cantidad de unidades que desea pedir” y dos listas desplegables correspondientes al nombre del plato y las unidades a pedir.

Seleccione en cada una de ellas la opción de su preferencia y de clic en el botón “Agregar plato”, asegúrese de no dejar la selección vacía u obtendrá una ventana emergente con un mensaje de error, si la obtiene, ciérrela y haga la selección de forma correcta.



The screenshot shows a dark-themed interface. At the top, a header bar contains the text "Realizar el pedido de mi reserva". Below this, a message in red text reads: "¡Reserva encontrada! Bienvenid@ mariano herrera, esta es su visita #1 a Aura Gourmet (recuerde que tenemos descuentos por fidelidad cada 5 visitas)". Below the message, there is a section titled "A continuación seleccione el nombre del plato y la cantidad de unidades que desea pedir". This section contains two labels: "plato" and "unidades a pedir". To the right of these labels are two dropdown menus. The first dropdown menu is labeled "SALMÓN AHUMADO" and the second dropdown menu is labeled "3". At the bottom, there is a button labeled "Agregar plato".

5. Pedir más o terminar de pedir:

Luego observará un mensaje que le confirma que las unidades del plato seleccionado fueron añadidas con éxito y “¿Qué desea hacer ahora?”

- Seleccione el botón “Continuar Pidiendo” si desea agregar más platos a su pedido y vuelva a repetir los mismos pasos del punto 4
- Seleccione el botón “Finalizar pedido” si ya no desea añadir más platos

Realizar el pedido de mi reserva

¡Reserva encontrada!
Bienvenid@ mariano herrera, esta es su visita #1 a Aura Gourmet
(recuerde que tenemos descuentos por fidelidad cada 5 visitas)

3 unidades de SALMÓN AHUMADO añadidas con éxito!
¿Qué desea hacer ahora?

Continuar Pidiendo **Finalizar Pedido**

Al seleccionar “Finalizar Pedido”

6. Selección de plato de cortesía (aplica solo para reservas con mesa tipo deluxe)

Si su mesa es deluxe, el restaurante le ofrece un plato de cortesía, observará una lista desplegable con 5 platos diferentes, seleccione la cortesía de su preferencia y de clic al botón “Guardar selección”, asegúrese de no dejar la selección vacía u obtendrá una ventana emergente con un mensaje de error, si la obtiene, ciérrela y haga la selección de forma correcta.

Realizar el pedido de mi reserva

¡Reserva encontrada!
Bienvenid@ mariano herrera, esta es su visita #1 a Aura Gourmet
(recuerde que tenemos descuentos por fidelidad cada 5 visitas)

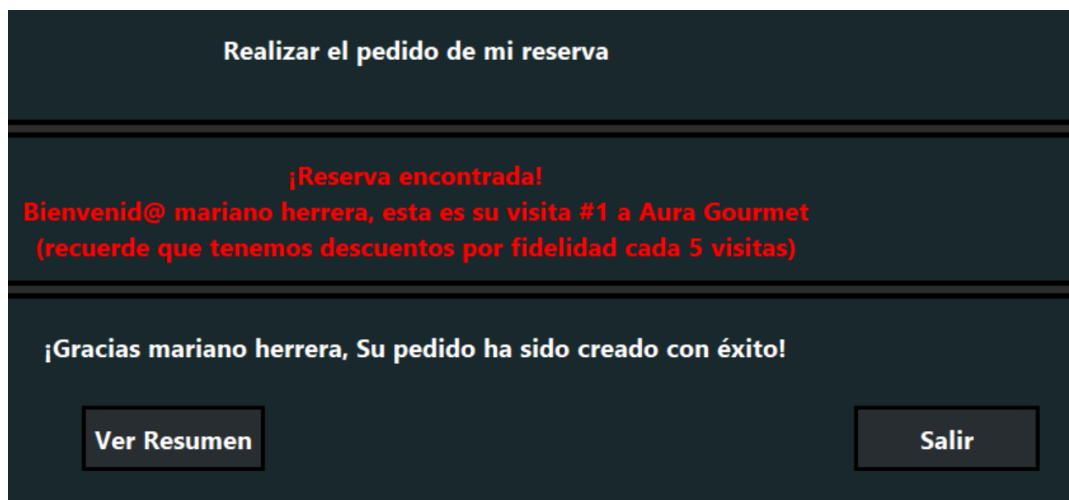
¡Su reserva con mesa tipo deluxe incluye un plato de cortesía! **seleccione una opción**

plato de cortesía COPA DE HELADO

Guardar selección

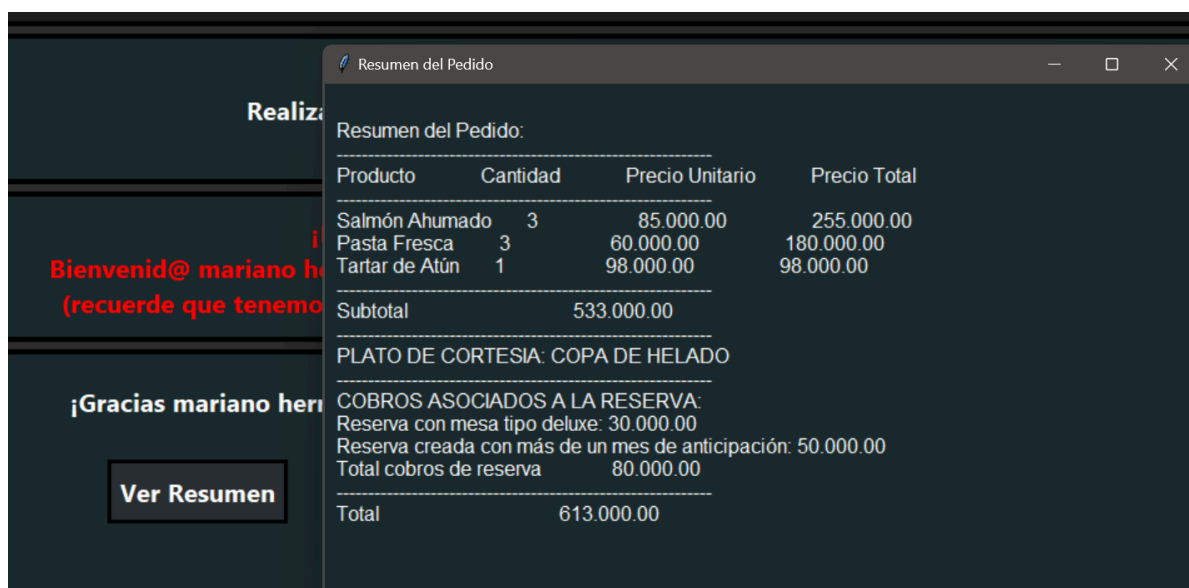
7. Ver resumen o cerrar la aplicación

Observará un mensaje de agradecimiento y los botones “Ver Resumen” y “Salir”



- Seleccione el botón “Ver Resumen” para ver toda la información asociada a su pedido en una ventana nueva.

ejemplo de resumen:



- Seleccione el botón “Salir” si ya observó su resumen y ya desea cerrar la aplicación, asegúrese de cerrar la aplicación únicamente dando clic en este botón

Funcionalidad 4 (Gestión de Recompensas):

Esta opción describe cómo los clientes interactúan con el sistema de recompensas y puntos acumulados al realizar compras, y su frecuencia de visitas a el restaurante. Los puntos pueden canjearse por productos, servicios exclusivos, descuentos, y otros beneficios. A continuación, se explican los pasos detallados para gestionar recompensas, redimir puntos, y cómo los usuarios pueden evaluar el funcionamiento del sistema.

1. Acceso al Sistema

Para acceder al sistema de gestión de recompensas, el usuario debe estar registrado en el sistema del restaurante y estar autenticado adecuadamente. No se requieren credenciales como nombre de usuario o contraseña explícitamente para el proceso de recompensas, ya que el sistema se basa en la identificación del cliente, pero se asume que el sistema de gestión de clientes se encarga de la autenticación.

2. Requisitos Previos

El cliente debe haber realizado al menos un pedido en el restaurante para que se registren las compras asociadas a su identificación.

- **Identificación del cliente:** El sistema necesita identificar al cliente mediante su número de identificación único (generalmente, el ID de cliente).
- **Historial de pedidos:** El sistema verifica las compras realizadas por el cliente para calcular los puntos disponibles para recompensas.

3. Flujo de Funcionamiento del Sistema

3.1 Ingreso de la Identificación del Cliente

Al iniciar la funcionalidad de gestión de recompensas, el sistema pedirá al usuario que ingrese la **identificación del cliente**.

Paso 1: El sistema solicita el número de identificación del cliente para verificar el historial de compras:

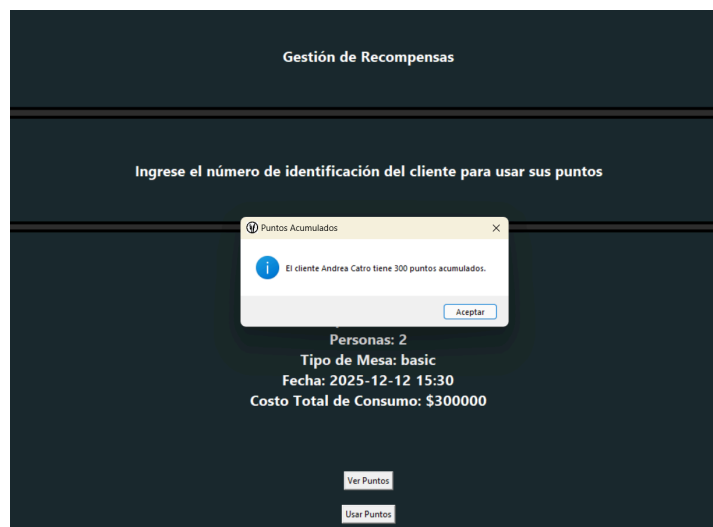
- Ejemplo de entrada: 123456789

3.2 Cálculo de Puntos

Una vez ingresada la identificación, el sistema procesa el historial asociados al cliente y calcula los puntos acumulados en base a dos criterios:

1. **Puntos por costo:** Se calcula el total gastado en compras y se convierte en puntos (1 punto por cada \$100 gastados).
2. **Puntos por visitas:** Se otorgan puntos por la cantidad de visitas realizadas (20 puntos por cada visita registrada).

El sistema muestra el total de puntos acumulados:



3.3 Opciones para Usar los Puntos

Una vez calculados los puntos, el cliente puede elegir cómo desea utilizar sus puntos. Las opciones disponibles son:

1. **Reserva:** Para usar los puntos en la reserva de una mesa o servicios en el restaurante. Requiere un mínimo de 50 puntos para realizar una reserva.
2. **Servicios Exclusivos:** Uso de los puntos para acceder a mesas deluxe o servicios especiales.
3. **Productos del Menú:** Uso de los puntos para canjear productos del menú del restaurante.
4. **Devolución de Producto:** El cliente puede iniciar un proceso de devolución de productos, eligiendo entre varias opciones de recompensa (reembolso, puntos adicionales, descuentos, crédito en restaurante).
5. **Mal Servicio:** El cliente puede reportar un mal servicio y recibir una recompensa, con opciones para aceptar o rechazar la compensación.

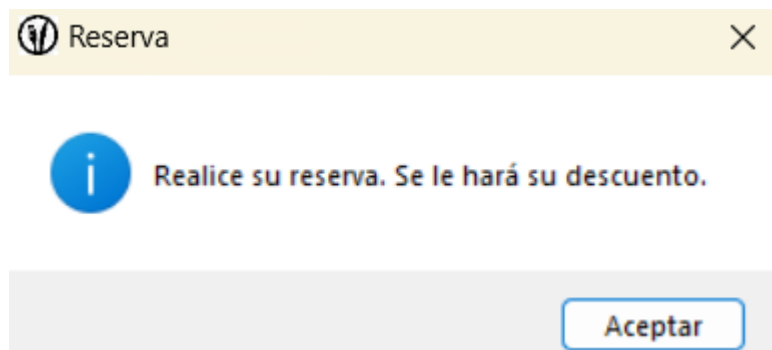
3.4 Proceso de Reserva

Si el cliente elige usar sus puntos para una **reserva**:

- El sistema solicita la cantidad de puntos que desea utilizar para la reserva.
- Si el cliente tiene suficientes puntos, la reserva es confirmada y se reducen los puntos correspondientes.

- **Requisito:** Se necesitan **al menos 50 puntos** para realizar una reserva.

Ejemplo de operación:

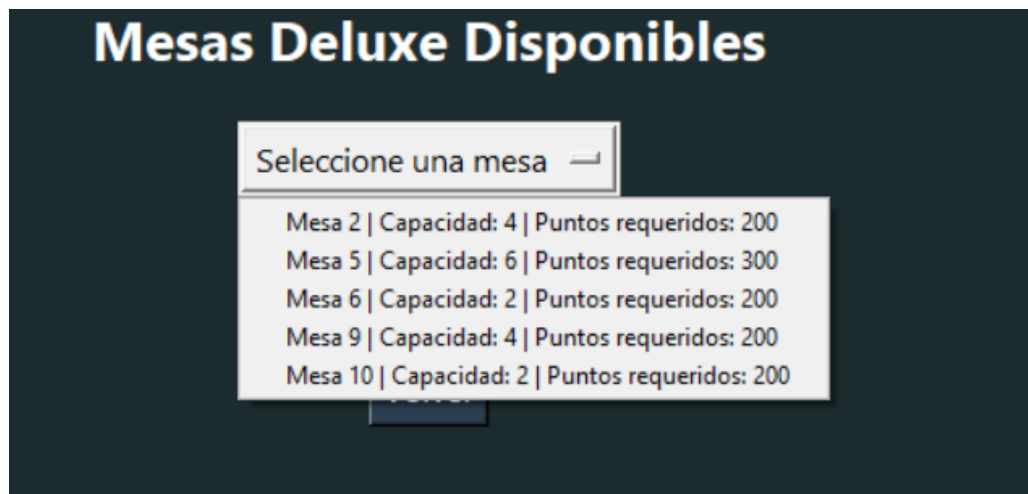


3.5 Proceso de Servicios Exclusivos

El cliente puede elegir entre mesas deluxe u otros servicios exclusivos:

- El sistema muestra una lista de opciones disponibles, junto con la cantidad de puntos necesarios.
- Si el cliente tiene suficientes puntos, puede elegir la opción deseada.

Ejemplo:



3.6 Proceso de Productos del Menú

El cliente puede utilizar los puntos para canjear productos del menú:

- El sistema muestra los productos disponibles junto con los puntos necesarios para canjear cada uno.
- El cliente selecciona el producto que desea y el sistema verifica si tiene suficientes puntos.

Ejemplo:



4. Validación del Sistema

4.1 Pruebas y Validaciones Requeridas

Para evaluar el correcto funcionamiento del sistema, los siguientes elementos deben ser validados:

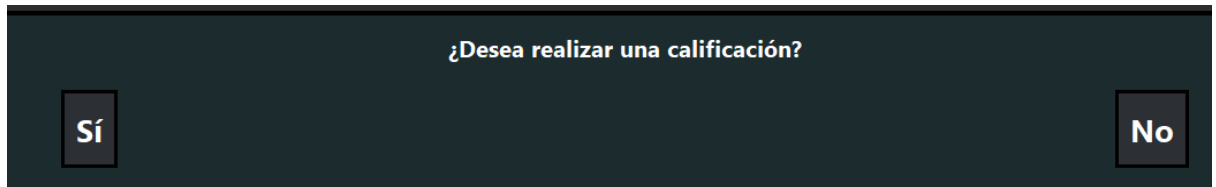
- **Acceso a la Identificación del Cliente:** Verifique que la identificación del cliente ingresada esté registrada correctamente en el sistema y que se pueda acceder al historial de Compras
- **Cálculo de Puntos:** Asegúrese de que los puntos por visitas y costo se calculen correctamente y se muestren en la interfaz.
- **Funcionalidad de Reserva:** Valide que se pueda usar los puntos para realizar una reserva y que los puntos se descuenten del saldo de puntos correctamente.
- **Servicios Exclusivos y Productos del Menú:** Compruebe que el cliente pueda canjear puntos para acceder a servicios exclusivos o productos del menú.

Se proporciona una visión general de cómo interactuar con el sistema de gestión de recompensas. En caso de que alguna funcionalidad no funcione según lo esperado, es recomendable verificar los datos de entrada y la configuración del sistema para garantizar que todos los requisitos previos estén cumplidos.

Funcionalidad 5 (Calificación del servicio):

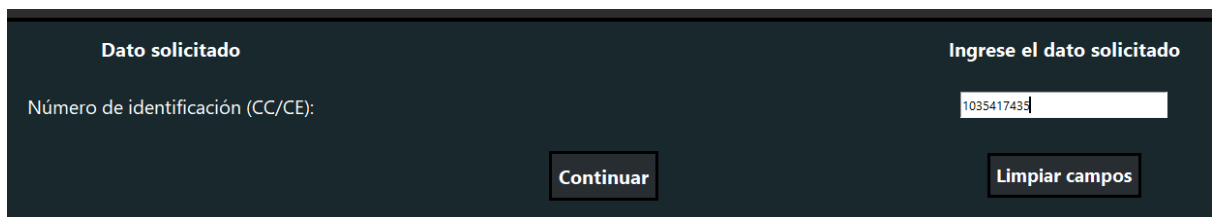
Nota: Al ingresar a esta opción del menú principal, usted confirma que ya consumió en el local, de manera presencial, o por medio de un domicilio. Todos los datos solicitados son meramente para uso interno, su calificación es de gran ayuda para mejorar nuestro servicio y brindarle a nuestros clientes la mejor experiencia.

1. Se despliega un mensaje donde se le pregunta si desea realizar una calificación.



A dark-themed modal dialog box with the title "¿Desea realizar una calificación?". It contains two buttons: "Sí" (Yes) on the left and "No" (No) on the right.

- 1.1 Ingrese si ó no, en caso de seleccionar no, será reenviado al menú principal.
2. Si escogió **Sí** en el punto anterior se mostrará en pantalla una petición de su identificación (CC,TE), donde usted debe ingresar su documento de identidad.



A dark-themed form with the title "Dato solicitado" on the left and "Ingrese el dato solicitado" on the right. Below the title, it says "Número de identificación (CC/CE):". To the right of this text is a text input field containing the number "1035417435". Below the input field are two buttons: "Continuar" (Continue) and "Limpiar campos" (Clear fields).

- 2.1 Si su identificación no se encuentra alojada en nuestro sistema se le indicará que el número no está asociado a ningún cliente, y se le dará la opción de ingresar una identificación válida
- 2.2 Si el tipo de dato ingresado es no numérico, se le indicará que la entrada no es correcta, y se le volverá a solicitar una entrada válida.
3. Si usted ingresó correctamente una identificación asociada a un cliente el sistema se encargará de enviarlo al sistema de calificación correspondiente.

Nota: Para cualquier tipo de encuesta dentro de un sistema de calificación solo debe ingresar el valor numérico que corresponda a su opción a escoger y que se encuentre dentro de las opciones, en caso de que deje una opción sin calificar le saltará una alerta y se le volverá a solicitar.

3.1 Si usted es un cliente que consumió en el local se le realizará la encuesta correspondiente

The image shows a dark-themed survey form. At the top left, it says 'Apartado de la encuesta'. On the right, it says 'Seleccione su calificación'. There are three rows of questions, each followed by a dropdown menu labeled 'Seleccione una opción':

- Para puntuar la calidad de la comida:
- Para puntuar la calidad del mesero:
- Para puntuar el tiempo de espera:

At the bottom center, there is a button labeled 'Continuar'.

3.1.1 Se le indica que puntée la calidad de nuestra comida del 1 al 5, basado en el siguiente criterio:

1. Muy mala (fría, sin sabor, ingredientes de mala calidad).
2. Mala (pudo mejorar, pero no fue buena).
3. Regular (aceptable, pero nada especial).
4. Buena (sabrosa, pero con detalles mejorables).
5. Excelente (deliciosa, fresca y bien preparada).

3.1.2 Se le indica que puntée la calidad del mesero del 1 al 5, basado en el siguiente criterio:

1. Muy mala (desatento, descortés o poco servicial).
2. Mala (lento o poco amable, pero aceptable).
3. Regular (atención básica, sin destacar).
4. Buena (amable y atento, con buen servicio).
5. Excelente (muy cordial, rápido y servicial).

3.1.3 Se le indica que puntée el tiempo de espera del 1 al 5, basado en el siguiente criterio:

1. Muy lento (tardó demasiado, fue frustrante).
2. Lento (más de lo esperado, pero aceptable).
3. Regular (tiempo promedio, ni rápido ni lento).

4 .Rápido (llegó en buen tiempo, sin demoras).

5 .Excelente (muy rápido, antes de lo esperado).

3.1.4 Se le habilita la posibilidad de dejarnos un comentario (opcional), ingrese “1” y déjenos su valioso comentario o ingrese “2” para continuar.

3.2 Si usted es un cliente que consumió a domicilio se le realizará la encuesta correspondiente

3.2.1 Se le indica que puntée la calidad de nuestra comida del 1 al 5, basado en el siguiente criterio:

1. Muy mala (fría, sin sabor, ingredientes de mala calidad).

2. Mala (pudo mejorar, pero no fue buena).

3. Regular (aceptable, pero nada especial).

4. Buena (sabrosa, pero con detalles mejorables).

5. Excelente (deliciosa, fresca y bien preparada).

3.2.2 Se le indica que puntée el tiempo de espera del 1 al 5, basado en el siguiente criterio:

1. Muy lento (tardó demasiado, fue frustrante).

2. Lento (más de lo esperado, pero aceptable).

3. Regular (tiempo promedio, ni rápido ni lento).

4 .Rápido (llegó en buen tiempo, sin demoras).

5 .Excelente (muy rápido, antes de lo esperado).

3.2.3 Se le habilita la posibilidad de dejarnos un comentario (opcional), ingrese si y déjenos su valioso comentario o ingrese no para continuar.

Por ultimo, ¿desea dejar un comentario?

Sí

No

4. A continuación su calificación es almacenada y utilizada para actualizar nuestras estadísticas.

Por consiguiente se aplicarán las siguientes consideraciones para los siguientes casos:

4.1 Para consumo en el local:

-Se le otorga un descuento del 5% si el promedio de su calificación es menor o igual que 3.

-Se le otorga un descuento del 10% si el promedio de su calificación es menor o igual que 2.

-En otro caso, no se aplicará ningún descuento.

4.1.1 Se le imprimirá una factura con:

Su nombre, su mesero encargado, su total (con descuento si aplica), el descuento aplicado y su promedio en la calificación. De la siguiente manera:

(Ejemplo)

A continuación se le proporciona un resumen de la factura de su consumo.
Muchas gracias por escogernos,
esperamos verlo pronto.

=====

FACTURA DE CONSUMO AURA GOURMET

Restaurante: Aura Gourmet

Cliente: Kevin

Mesero encargado: Nombre: Ana María Sánchez, Calificación: 2.0

Total: 81000.0

Descuento aplicado: 10% de descuento

Calificación del servicio: 2.0/5

=====

Gracias por visitarnos. ¡Esperamos verlo pronto!

=====

Continuar

4.1.2 Para clientes con mesa deluxe:

Si usted en su reserva escogió una mesa deluxe se le dará acceso al apartado exclusivo para clientes premium, allí se le proporcionará la posibilidad de visualizar el impacto en nuestro restaurante respecto a su calificación, primero tendrá que escoger si desea verla (ingresar “Sí”) o en caso contrario (ingresar “No”).

Si indicó ver los cambios causados por su calificación se le mostrará de la siguiente forma:

(Ejemplo)

ANTES DE SU CALIFICACIÓN		DESPUÉS DE SU CALIFICACIÓN
Restaurante:	3.8	3.7
Mesero:	2.0	2.0
Prioridad Mesero:	10	10
El mesero Ana María Sánchez, sigue activo		

Finalizar

Donde ***Estado del mesero***, indica si con su calificación el respectivo mesero acumuló las suficientes calificaciones negativas consecutivas para ser amonestado.

En caso de que no posea mesa deluxe se le notificará que en su próxima visita que puede obtener acceso a un apartado exclusivo por medio de una reserva con mesa deluxe.

4.1.3 Por último usted será desplazado nuevamente al menú principal dando por terminado esta funcionalidad de calificaciones.

4.2 Para consumo a domicilio:

-Se le otorga un descuento del 5% si el promedio de su calificación es menor o igual que 3.

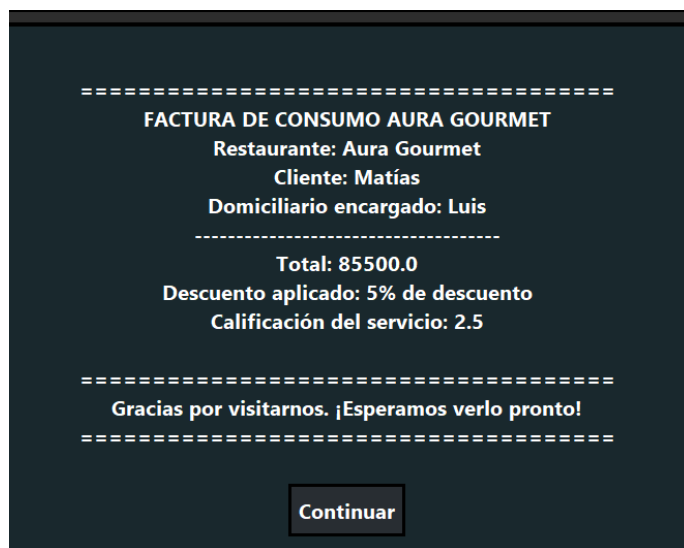
-Se le otorga un descuento del 10% si el promedio de su calificación es menor o igual que 2.

-En otro caso, no se aplicará ningún descuento.

4.2.1 Se le imprimirá una factura con:

Su nombre, su domiciliario encargado, su total (con descuento si aplica), el descuento aplicado, su promedio en la calificación. De la siguiente manera:

(Ejemplo)



4.2.2 Para clientes con domicilio prioritario:

Si usted en su domicilio escogió una pedido prioritario se le dará acceso al apartado exclusivo para clientes premium, allí se le proporcionará la posibilidad de visualizar el impacto en nuestro restaurante respecto a su

calificación, primero tendrá que escoger si desea verla (ingresar “Si”) o en caso contrario (ingresar “No”).

Si indicó ver los cambios causados por su calificación se le mostrará de la siguiente forma:

(Ejemplo)

ANTES DE SU CALIFICACIÓN		DESPUÉS DE SU CALIFICACIÓN	
Restaurante:	3.7		3.7
Domiciliario:	4.0		4.0
<div>Finalizar</div>			

4.2.3 Por último usted será desplazado nuevamente al menú principal dando por terminado esta funcionalidad de calificaciones.