

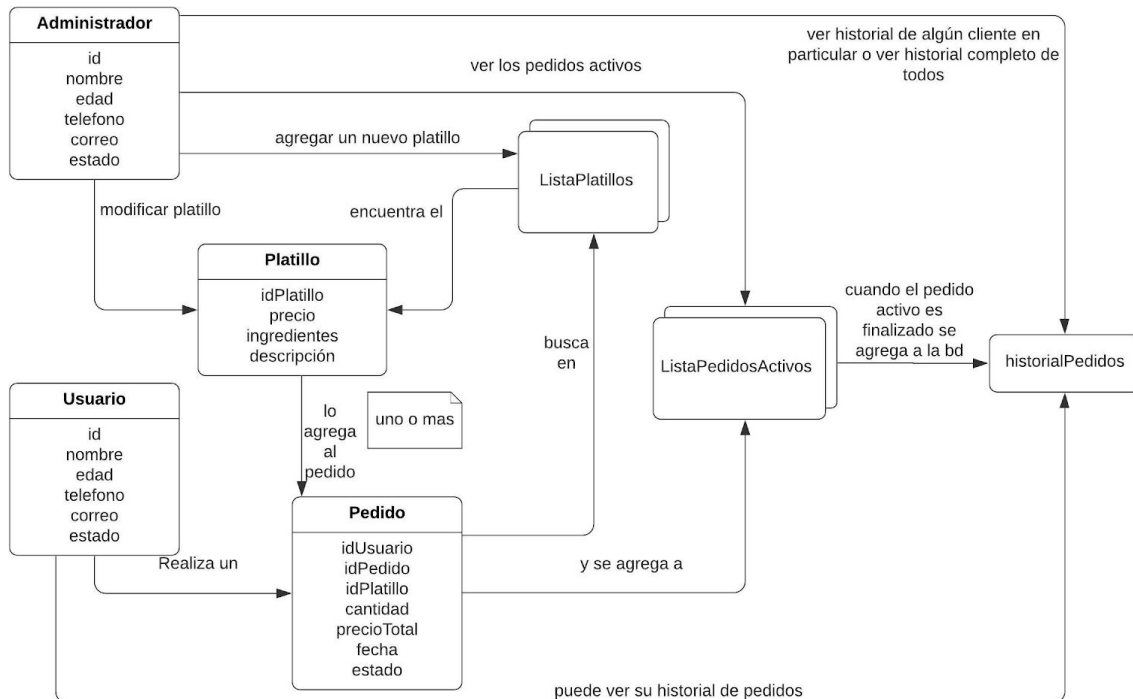
Software de ingeniería INFO248  
Profesores – Raimundo Vega / Cristian Olivares  
Ingeniería Civil en Informática

# Neuss producciones Ltda.

Análisis y diseño del proyecto de software

**Integrantes:** - Andrés Gutiérrez Salas  
- Miguel Nahuelpán Aliante  
- Esteban Tejeda Webar

# 1. Modelo de clases conceptuales



## 2. Casos de uso

### 1.1 Realizar pedido

#### 1.1.1 Descripción

- Caso de uso CU01: Realizar pedido.
- Actores: Clientes.
- Propósito: Realizar un pedido de uno o más platillos que desea comprar
- Pre-condición: el cliente debe tener una cuenta y haber iniciado sesión.
- Resumen: El cliente abre la aplicación en su dispositivo móvil, inicia sesión ingresando su nombre de usuario y contraseña, el sistema autentifica al cliente. El cliente selecciona la opción de agregar los platillos que desea consumir de una lista de platillos disponibles.
- Tipo: primario

#### 1.1.2 Flujo normal de eventos

Acción de los actores	Respuesta del sistema
1. El cliente ingresa a la aplicación móvil	
	2. El sistema solicita que ingrese su nombre de usuario y la contraseña
3. El usuario ingresa los datos de nombre de usuario y contraseña	
	4. El sistema valida al usuario y lo lleva a la ventana principal
5. El usuario selecciona los platillos que desea pedir	
	6. El sistema crea una lista de platillos
7. El usuario pide los platillos	
	8. El sistema almacena el pedido y despliega el estado del pedido "En revisión"

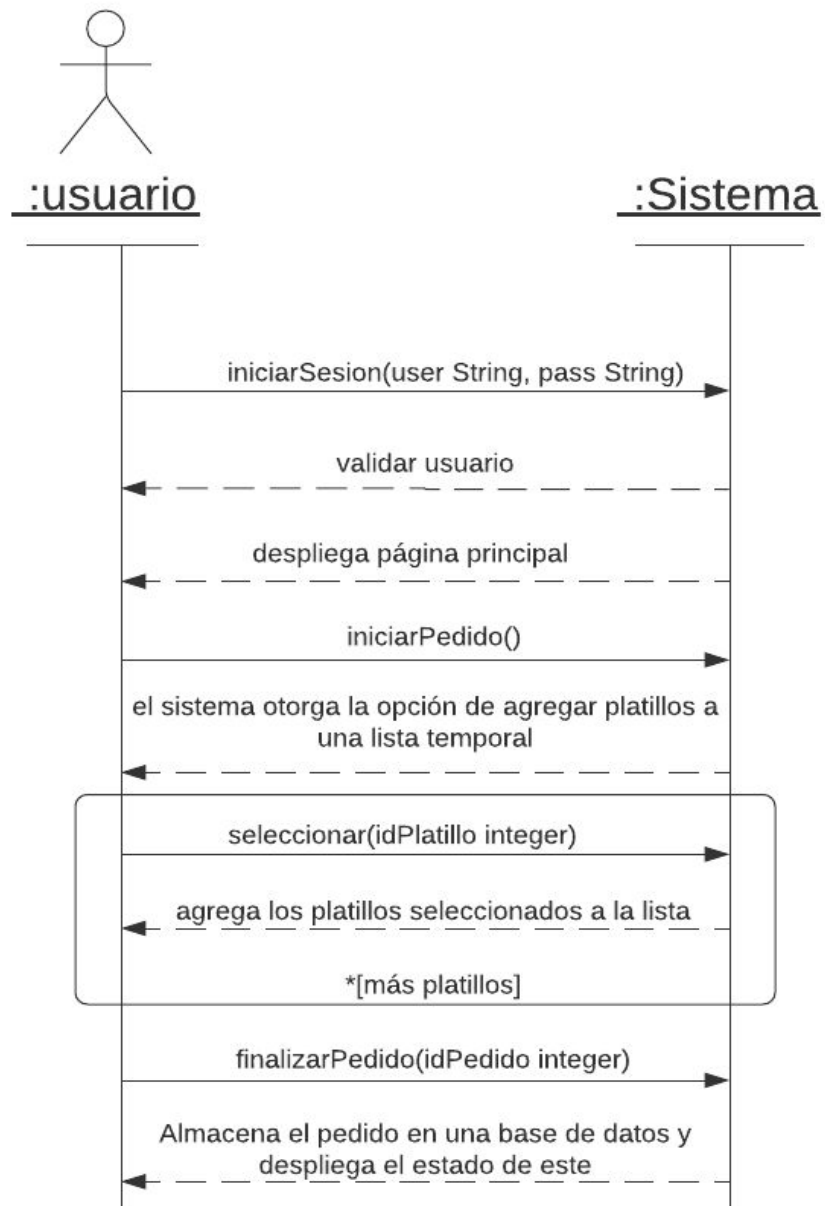
#### 1.1.3 Curso alternativo:

Línea 2: el cliente ingresa un nombre de usuario o una contraseña no válida, el sistema vuelve a solicitar que el usuario ingrese sus datos.

Línea 5: el usuario no ha seleccionado platillos, por ende el sistema no crea la lista de platillo.

Línea 6: el sistema no puede crear la lista de platillos, ya sea por que el platillo seleccionado esté en fuera de stock o no queden ingredientes para su preparación. El sistema devuelve un mensaje de error y no genera la lista de platillos.

#### 1.1.4 Diagrama de secuencia



### 1.1.5 Contratos

#### Contrato CO01: iniciarSesion

- Nombre: iniciarSesion(user String, pass String)
- Responsabilidades: Autentifica al usuario y despliega la página principal de la aplicación.
- Pre-condiciones: El sistema debe tener un registro de usuarios para poder identificarlo, el sistema de validación será por la componente de inicio de sesión de google.
- Pos-condiciones: se modifica usuario.estado="activo".

#### Contrato CO02: iniciarPedido

- Nombre: iniciarPedido()
- Responsabilidades: habilita la opción de seleccionar platillos para hacer un pedido.
- Pre-condiciones: Haber iniciado sesión.
- Pos-condiciones: Se estableció un enlace con la lista de platillos disponibles.

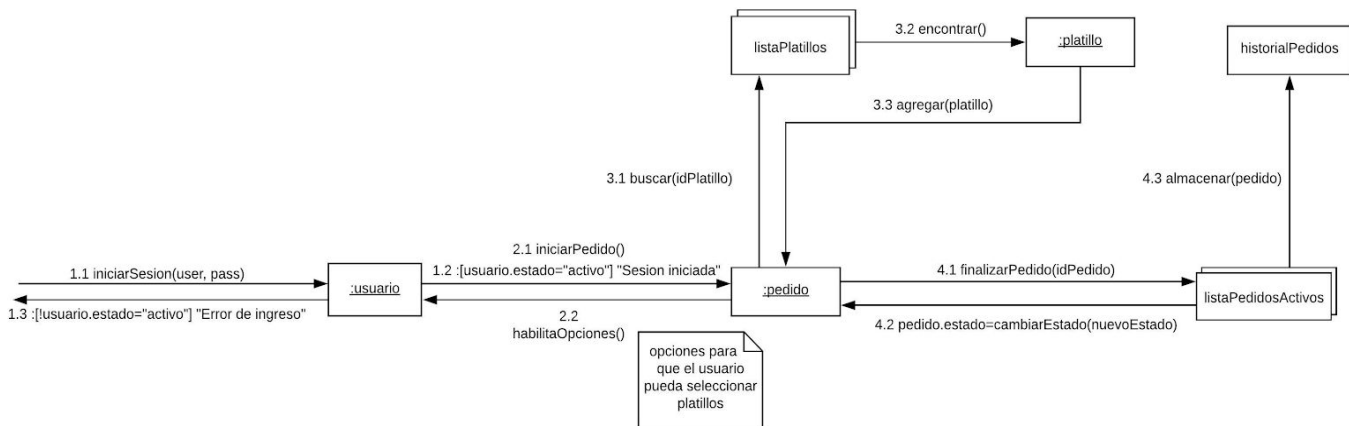
#### Contrato CO03: seleccionar

- Nombre: seleccionar(idPlatillo integer)
- Responsabilidades: si es el primer platillo que selecciona, crea una lista y agrega idPlatillo a la lista. Si no es el primer platillo que selecciona, agrega idPlatillo a lista ya existente.
- Pre-condiciones: Haber iniciado sesión y haber iniciado el pedido.
- Pos-condiciones: Se crea una instancia de pedido, lista con los platillos seleccionados.

#### Contrato CO04: finalizarPedido

- Nombre: finalizarPedido(idPedido integer)
- Responsabilidades: Añade a la base de datos un pedido de un usuario y se agrega a una lista de pedidos en los servidores de la empresa de los pedidos activos, se envía un mensaje de comprobación del pedido.
- Pre-condiciones: Haber iniciado sesión y haber seleccionado el/los platillos.
- Pos-condiciones: Se actualiza la lista de pedidos activos, y se establece un enlace con historialPedidos.

### 1.1.6 Diagramas de colaboración



## 1.2 Atender pedido

### 1.2.1 Descripción

- Caso de uso: CU02: Atender pedido
- Actores: Administradores.
- Propósito: Responder a la solicitud de un cliente
- Pre-condición: El administrador debe tener una cuenta y haber iniciado sesión. Un cliente debe haber realizado un pedido.
- Resumen: El administrador abre la aplicación, inicia sesión ingresando su nombre de usuario y contraseña, el sistema autentifica al administrador. El administrador selecciona un botón para ver pedidos activos y escoge uno para atender. Dependiendo de las circunstancias el administrador modifica el estado del pedido. Termina cuando el pedido es entregado o cancelado.
- Tipo: Primario

### 1.2.2 Flujo normal de eventos

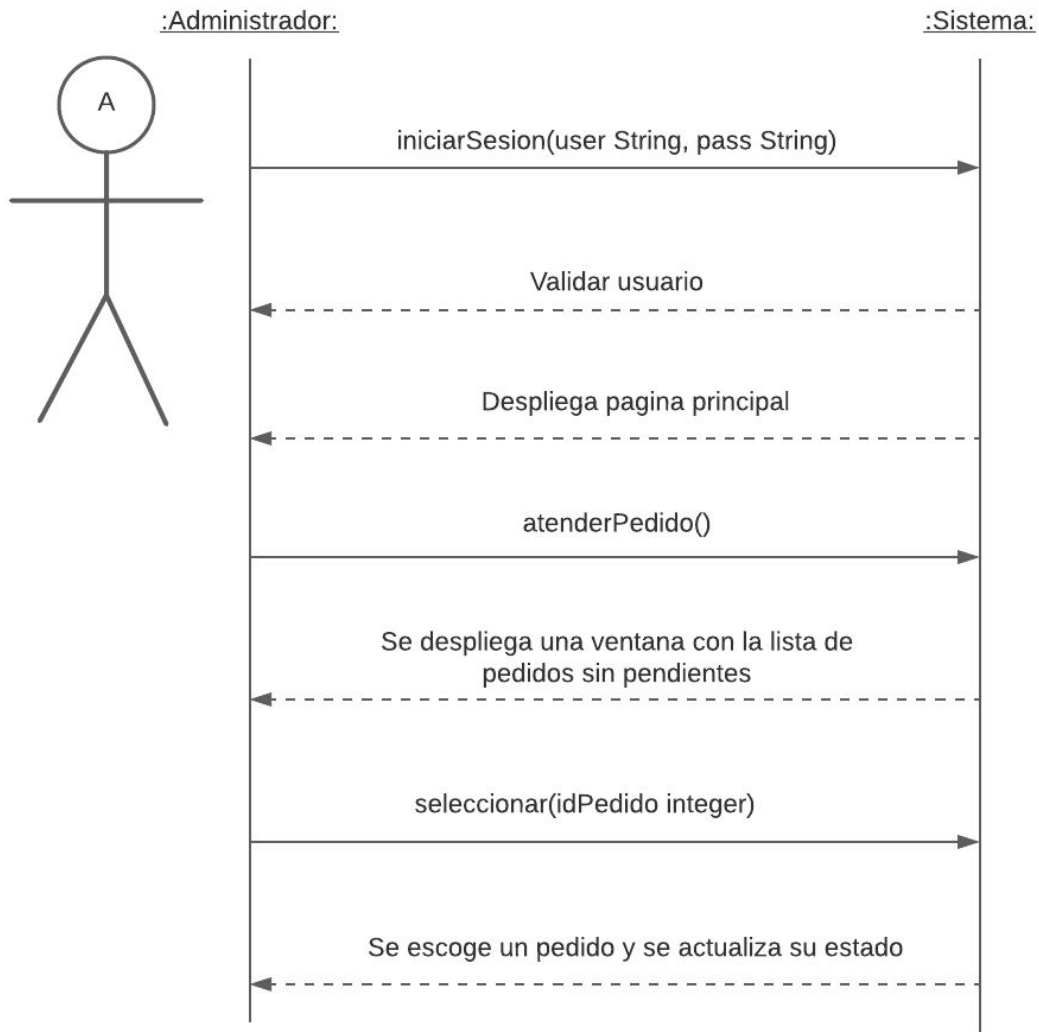
Acción de los actores	Respuesta del sistema
1. El administrador ingresa a la aplicación móvil	
	2. El sistema solicita que ingrese su nombre de usuario y la contraseña
3. El administrador ingresa los datos de nombre de usuario y contraseña	
	4. El sistema valida al usuario y lo lleva a la ventana principal
5. El administrador selecciona la opción de “Atender pedidos”	
	6. El sistema despliega una ventana con los pedidos pendientes
7. El administrador escoge uno para atender y preparar.	
	8. El sistema despliega una ventana para escoger el nuevo estado del pedido.
9. El administrador cambia el estado del pedido a “En preparación”	
	10. El sistema refleja el cambio, tanto para el administrador como para el cliente que hizo el pedido.
11. El administrador teniendo el pedido preparado, cambia el estado del pedido a “En camino”	
	12. El sistema refleja el cambio, tanto para el administrador como para el cliente que hizo el pedido.

### 1.2.3 Curso alternativo:

Línea 2: el cliente ingresa un nombre de usuario o una contraseña no válida, el sistema vuelve a solicitar que el usuario ingrese sus datos.

Línea 9: El administrador envía un mensaje al usuario que su pedido no podrá ser realizado por algún inconveniente (falta de ingredientes, problemas de personal, etc.)

#### 1.2.4 Diagrama de secuencia



#### 1.2.5 Contratos

##### Contrato CO01: iniciarSesion

- Nombre: `iniciarSesion(user String, pass String)`
- Responsabilidades: Autentifica al usuario y despliega la página principal de la aplicación.
- Pre-condiciones: El sistema debe tener un registro de usuarios para poder identificarlo, el sistema de validación será por la componente de inicio de sesión de google.
- Pos-condiciones: se modifica `usuario.estado="activo"`.



#### Contrato CO02: atenderPedido

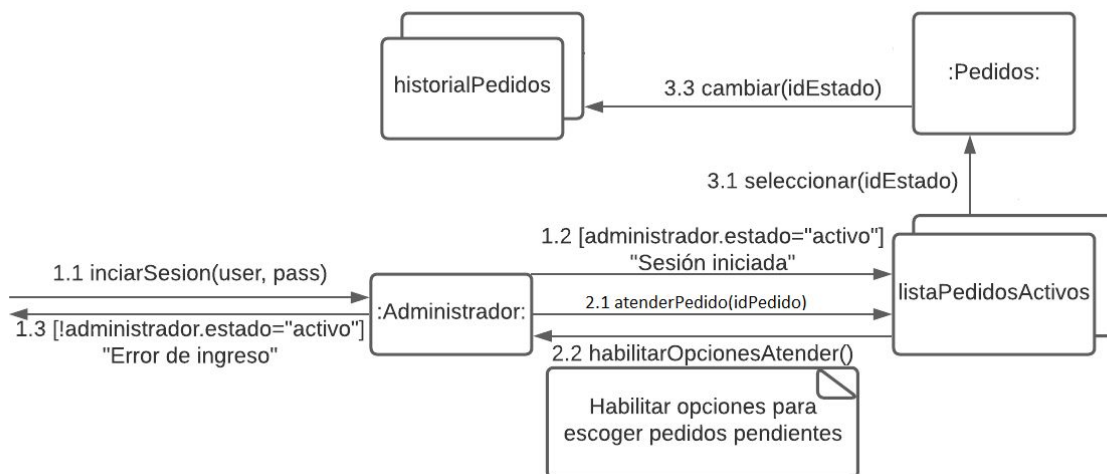
- Nombre: atenderPedido(idPedido)
- Responsabilidades: Despliega una ventana con todos los pedidos que han sido ingresados al sistema previamente.
- Pre-condiciones: El sistema debe tener un registro de usuarios para poder identificarlo, el sistema de validación será por la componente de inicio de sesión de google. El sistema debe tener al menos un pedido sin concretar.
- Pos-condiciones: Se genera un enlace con la lista de pedidos pendientes. Se establece un enlace con historialPedidos.

#### Contrato CO

##### 03: seleccionar

- Nombre: seleccionar(idPedido integer)
- Responsabilidades: Se escoge un pedido, para leer los detalles y actualizar el estado a “En preparación” o “En camino”.
- Pre-condiciones: El sistema debe tener un registro de usuarios para poder identificarlo, el sistema de validación será por la componente de inicio de sesión de google. El sistema debe tener al menos un pedido sin concretar.
- Pos-condiciones: Se actualiza la información en pantalla sobre “estado”.

#### 1.2.6 Diagramas de colaboración



## 1.3 Agregar platillo

### 1.3.1 Descripción

- Caso de uso: CU03: Agregar platillo
- Actores: Administradores
- Propósito: Actualizar la variedad de ofertas disponibles.
- Pre-condición: El administrador debe tener una cuenta y haber iniciado sesión.
- Resumen: El administrador abre la aplicación, inicia sesión ingresando su nombre de usuario y contraseña, el sistema autentifica al administrador. El administrador selecciona un botón para ver añadir oferta. Se despliega un menú donde podrá ingresar el nombre, los ingredientes, una descripción, el precio y una foto referencial. El administrador valida el ingreso guardándolo.
- Tipo: Secundario.

### 1.3.2 Flujo normal de eventos

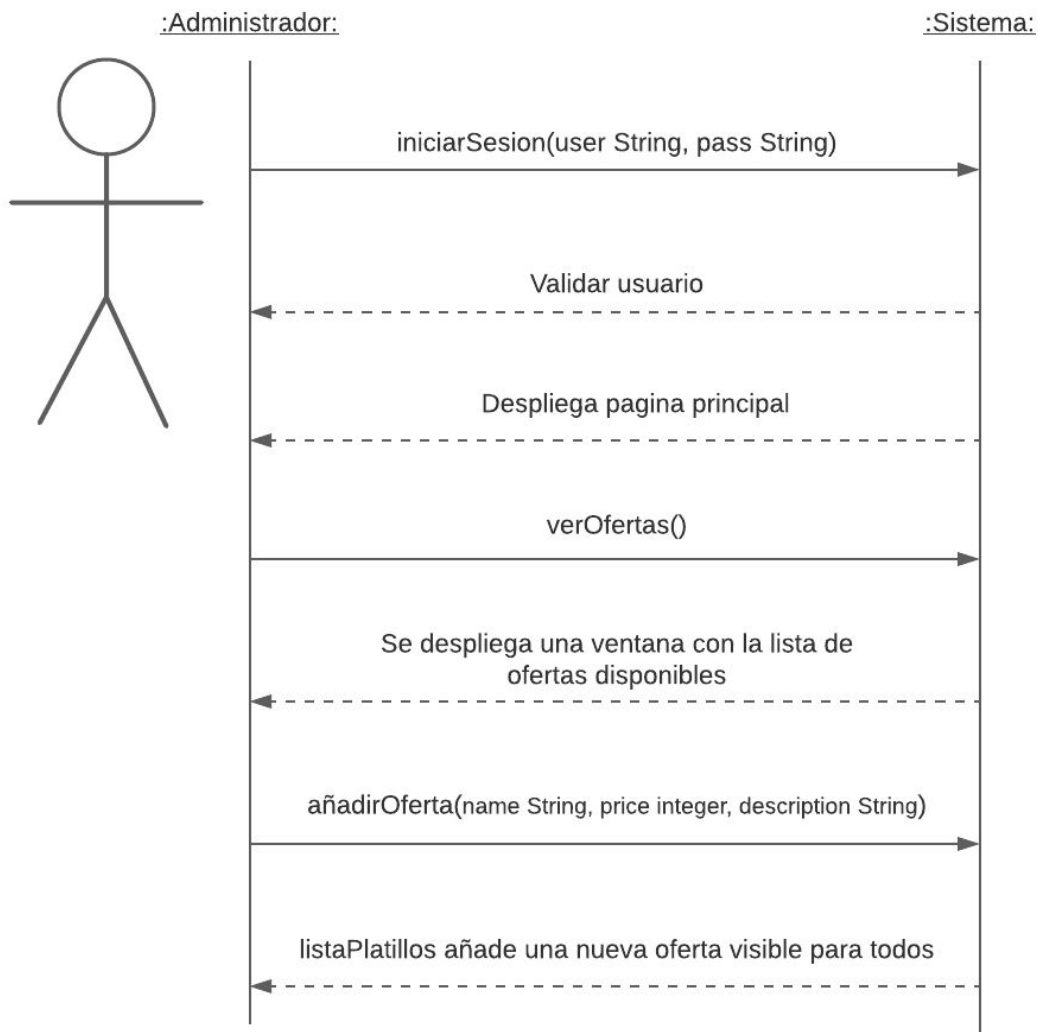
Acción de los actores	Respuesta del sistema
1. El administrador ingresa a la aplicación móvil	
	2. El sistema solicita que ingrese su nombre de usuario y la contraseña
3. El administrador ingresa los datos de nombre de usuario y contraseña	
	4. El sistema valida al usuario y lo lleva a la ventana principal
5. El administrador selecciona el botón "Actualizar ofertas"	
	6. El sistema despliega una ventana que muestra los campos a llenar para una nueva oferta
7. El usuario rellena toda la información pertinente (nombre, ingredientes, etc.)	
	8. El sistema valida el ingreso, añadiendolo a lista visible desde la aplicación.

### 1.3.3 Curso alternativo:

Línea 2: el cliente ingresa un nombre de usuario o una contraseña no válida, el sistema vuelve a solicitar que el usuario ingrese sus datos.

Línea 8: Si hay algún problema con los datos ingresados (nombre ya en uso o campo vacío), el sistema lo hará notorio y pedirá que este dato sea reingresado.

### 1.3.4 Diagrama de secuencia



### 1.3.5 Contratos

#### Contrato CO01: iniciarSesion

- Nombre: `iniciarSesion(user String, pass String)`
- Responsabilidades: Autentifica al usuario y despliega la página principal de la aplicación.
- Pre-condiciones: El sistema debe tener un registro de usuarios para poder identificarlo, el sistema de validación será por la componente de inicio de sesión de google.
- Pos-condiciones: se modifica `usuario.estado="activo"`.

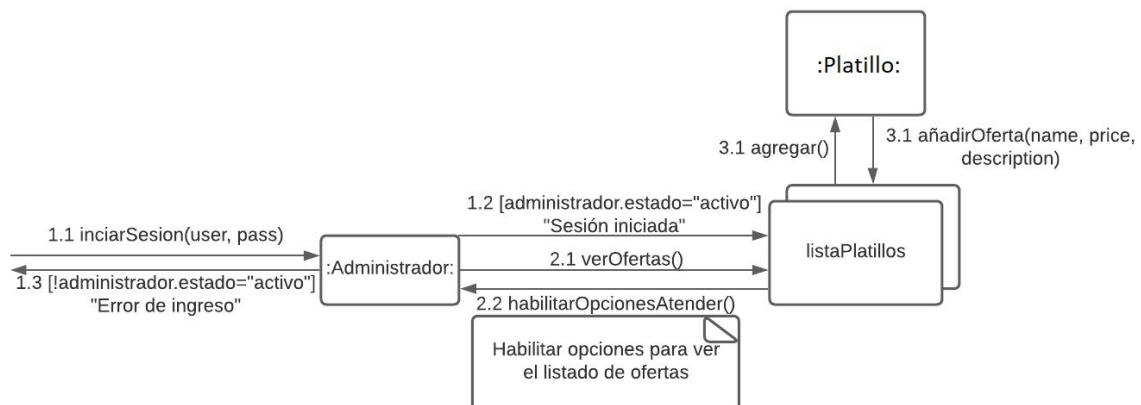
#### Contrato CO02: verOfertas

- Nombre: verOfertas()
- Responsabilidades: Se despliega la ventana con las ofertas disponibles.
- Pre-condiciones: El sistema debe tener un registro de usuarios para poder identificarlo, el sistema de validación será por la componente de inicio de sesión de google.
- Pos-condiciones: Se genera un enlace con listaPlatillos.

#### Contrato CO03: añadirOferta

- Nombre: añadirOferta(name String, price integer, description String)
- Responsabilidades: Actualiza listaPlatillos con un nuevo platillo.
- Pre-condiciones: El sistema debe tener un registro de usuarios para poder identificarlo, el sistema de validación será por la componente de inicio de sesión de google.
- Pos-condiciones: listaPlatillos añade un platillo nuevo.

#### 1.3.6 Diagramas de colaboración



#### 1.4 Revisar historial de un cliente

##### 1.4.1 Descripción

- Caso de uso: Revisar historial de un cliente

- Actores: Cliente
- Propósito: Ver todas las compras realizadas por el cliente en el pasado.
- Pre-condición: El cliente debe tener una cuenta y haber iniciado sesión. El cliente debe haber realizado al menos una compra.
- Resumen: El cliente abre la aplicación en su dispositivo móvil, inicia sesión ingresando su nombre de usuario y contraseña, el sistema autentifica al cliente. El cliente selecciona la opción de ver historial y se desplegará una ventana donde se ven los pedidos realizados previamente por el cliente.
- Tipo: Secundario

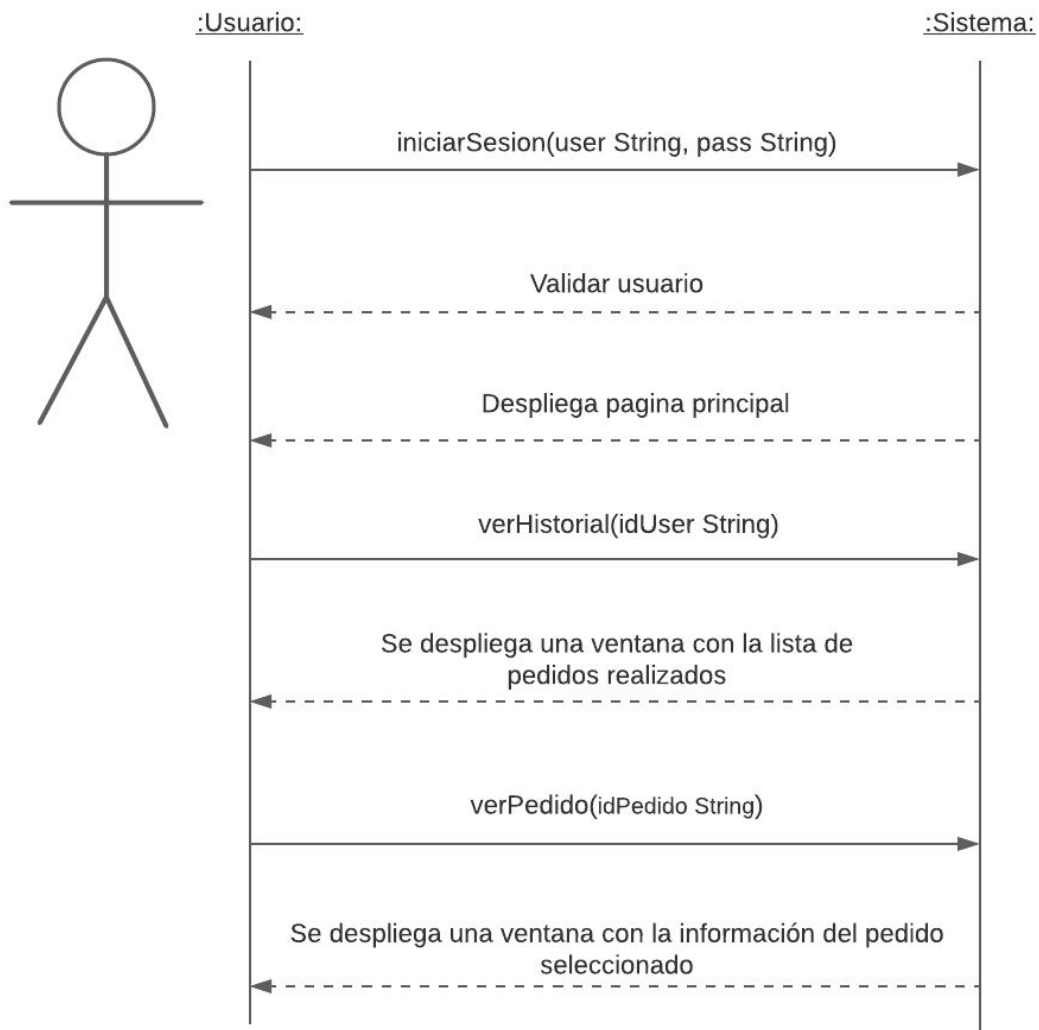
#### 1.4.2 Flujo normal de eventos

Acción de los actores	Respuesta del sistema
1. El cliente ingresa a la aplicación móvil	
	2. El sistema solicita que ingrese su nombre de usuario y la contraseña
3. El cliente ingresa los datos de nombre de usuario y contraseña	
	4. El sistema valida al usuario y lo lleva a la ventana principal
5. El cliente selecciona el botón “Ver historial de pedidos”	
	6. El sistema despliega una ventana que muestra los pedidos que ha realizado hasta la fecha el cliente

#### 1.4.3 Curso alternativo:

Línea 2: el cliente ingresa un nombre de usuario o una contraseña no válida, el sistema vuelve a solicitar que el usuario ingrese sus datos.

#### 1.4.4 Diagrama de secuencia



#### 1.4.5 Contratos

##### Contrato CO01: iniciarSesion

- Nombre: `iniciarSesion(user String, pass String)`
- Responsabilidades: Autentica al usuario y despliega la página principal de la aplicación.
- Pre-condiciones: El sistema debe tener un registro de usuarios para poder identificarlo, el sistema de validación será por la componente de inicio de sesión de google.
- Pos-condiciones: se modifica `usuario.estado="activo"`.

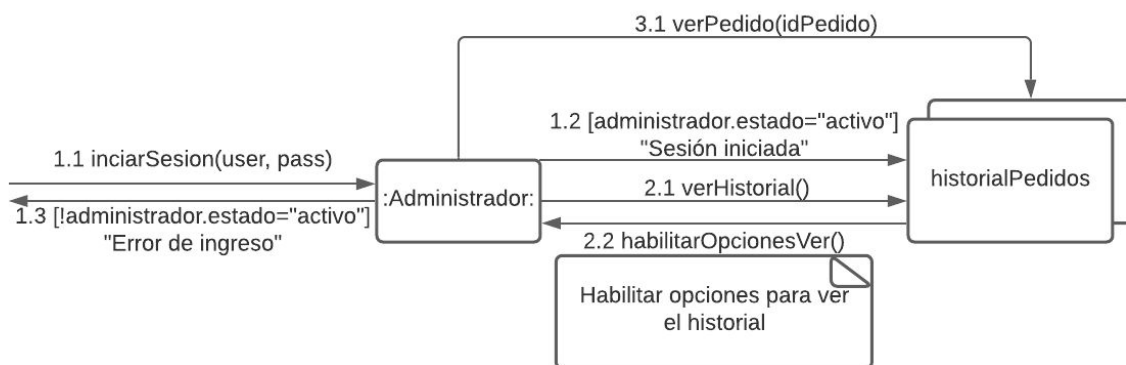
#### Contrato CO02: verHistorial

- Nombre: verHistorial(idUser String)
- Responsabilidades: Se despliega la ventana con el historial de pedidos del cliente.
- Pre-condiciones: El usuario debe haber realizado al menos un pedido para que el botón esté disponible.
- Pos-condiciones: Se genera un enlace con historialPedidos

#### Contrato CO03: verPedido

- Nombre: verPedido(idPedido String)
- Responsabilidades: Se despliega la ventana con los detalles del registro seleccionado.
- Pre-condiciones: El usuario debe tener acceso a su historial.
- Pos-condiciones: -

#### 1.4.6 Diagramas de colaboración



### 3. Diagrama de clases

