



Software de ingeniería INFO248

Profesor: Raimundo Vega/ Cristian Olivares/ Mirko Geregat

Ingeniería Civil en Informática

Neuss producciones Ltda.

Análisis y diseño del proyecto de software

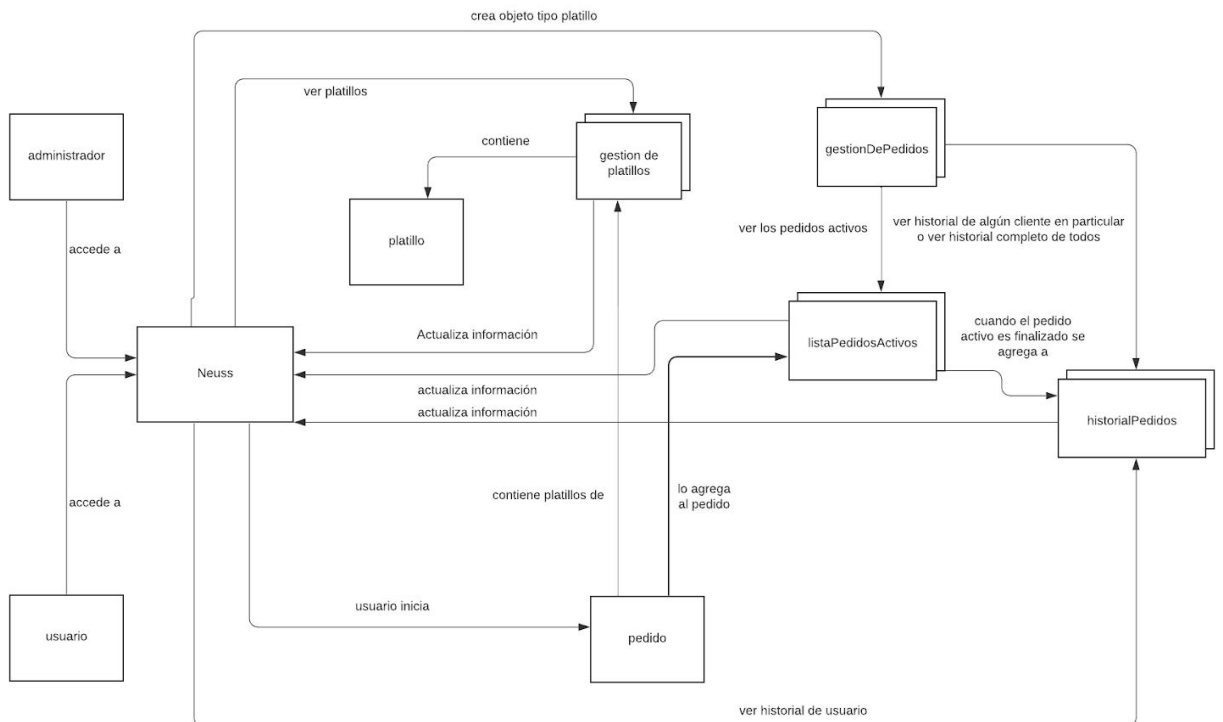
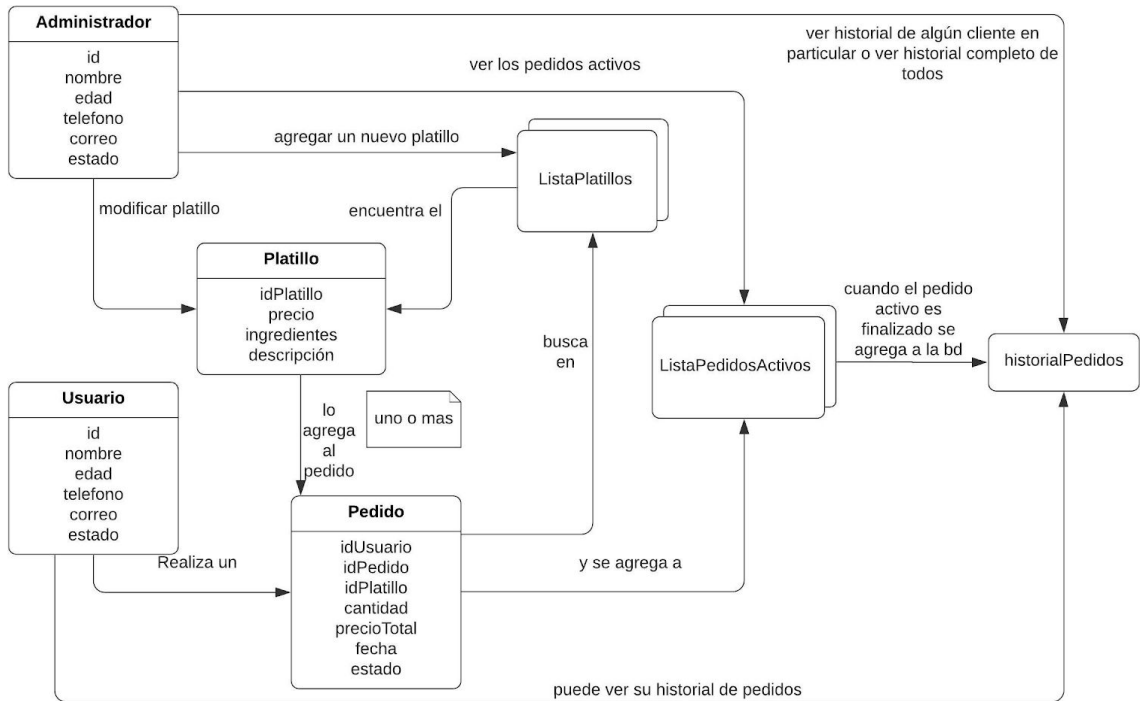
Integrantes: - Andrés Gutiérrez Salas
- Miguel Nahuelpán Aliante
- Esteban Tejeda Webar
- Felipe Zabala

Índice

Índice	2
1. Modelo de clases conceptuales	4
2. Casos de uso	5
2.1 Realizar pedido	5
2.1.1 Descripción	5
2.1.2 Flujo normal de eventos	5
2.1.3 Curso alternativo	6
2.1.4 Diagrama de secuencia	7
2.1.5 Contratos	7
2.1.5.1 Contrato CO01: iniciarSesion	7
2.1.5.2 Contrato CO02: iniciarPedido	8
2.1.5.3 Contrato CO03: seleccionar	8
2.1.5.4 Contrato CO04: finalizarPedido	8
2.1.6 Diagramas de colaboración	9
2.2 Atender pedido	10
2.2.1 Descripción	10
2.2.2 Flujo normal de eventos	10
2.2.3 Curso alternativo	11
2.2.4 Diagrama de secuencia	12
2.2.5 Contratos	12
2.2.5.1 Contrato CO01: iniciarSesion	12
2.2.5.2 Contrato CO02: atenderPedido	12
2.2.5.3 Contrato CO03: seleccionar	13
2.2.6 Diagramas de colaboración	13
2.3 Agregar platillo	14
2.3.1 Descripción	14
2.3.2 Flujo normal de eventos	14
2.3.3 Curso alternativo	15
2.3.4 Diagrama de secuencia	15
2.3.5 Contratos	16
2.3.5.1 Contrato CO01: iniciarSesion	16
2.3.5.2 Contrato CO02: verOfertas	16
2.3.5.3 Contrato CO03: añadirOferta	16
2.3.6 Diagramas de colaboración	16
2.4 Revisar su historial (cliente)	17
2.4.1 Descripción	17
2.4.2 Flujo normal de eventos	17
2.4.3 Curso alternativo	17

2.4.4 Diagrama de secuencia	18
2.4.5 Contratos	18
2.4.5.1 Contrato CO01: iniciarSesion	18
2.4.5.2 Contrato CO02: verHistorial	18
2.4.5.4 Contrato CO03: verPedido	18
2.4.6 Diagramas de colaboración	19
2.5 Fijar stock de platillos	20
2.5.1 Descripción	20
2.5.2 Flujo normal de eventos	20
2.5.3 Curso alternativo	21
2.5.4 Diagrama de secuencia	22
2.5.5 Contratos	23
2.5.5.1 Contrato CO01: iniciarSesion	23
2.5.5.2 Contrato CO02: seleccionarListaPlatillos	23
2.5.5.3 Contrato CO03: seleccionarPlatillo	23
2.5.5.4 Contrato CO04: actualizarStock	23
2.5.6 Diagramas de colaboración	24
3. Diagrama de clases	25

1. Modelo de clases conceptuales



2. Casos de uso

2.1 Realizar pedido

2.1.1 Descripción

- **Caso de uso CU01:** Realizar pedido.
- **Actores:** Clientes.
- **Propósito:** Realizar un pedido de uno o más platillos que desea comprar.
- **Precondición:** 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.
- **Referencias cruzadas:** RF10, RF18, RF14, RF15, RF16, RF19

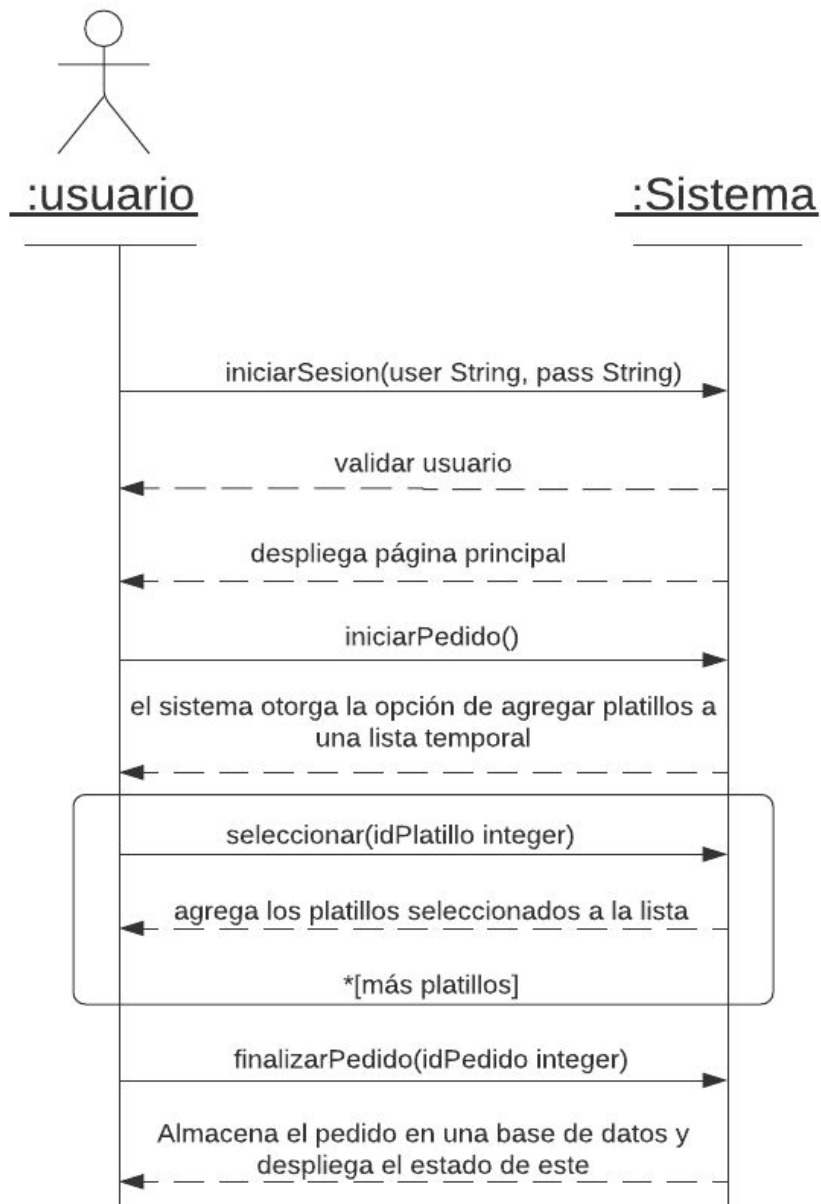
2.1.2 Flujo normal de eventos

Acción de los actores	Respuesta del sistema
1. El cliente ingresa a la aplicación.	
	2. El sistema solicita que inicie sesión
3. El usuario inicia sesión.	
	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 solicita un medio de pago
9. El usuario realiza el pago.	
	10. El sistema valida el pago.
	11. El sistema almacena el pedido y despliega el estado del pedido "En revisión".

2.1.3 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 platillos.
- **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.
- **Línea 10:** el pago no es válido, se vuelve a solicitar el pago (Línea 9).

2.1.4 Diagrama de secuencia



2.1.5 Contratos

2.1.5.1 Contrato CO01: iniciarSesion

- **Nombre:** `iniciarSesion(user String, pass String)`.
- **Responsabilidades:** Autentifica al usuario y despliega la página principal de la aplicación.

- **Precondiciones:** 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.
- **Poscondiciones:** se modifica usuario.estado="activo".

2.1.5.2 Contrato CO02: iniciarPedido

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

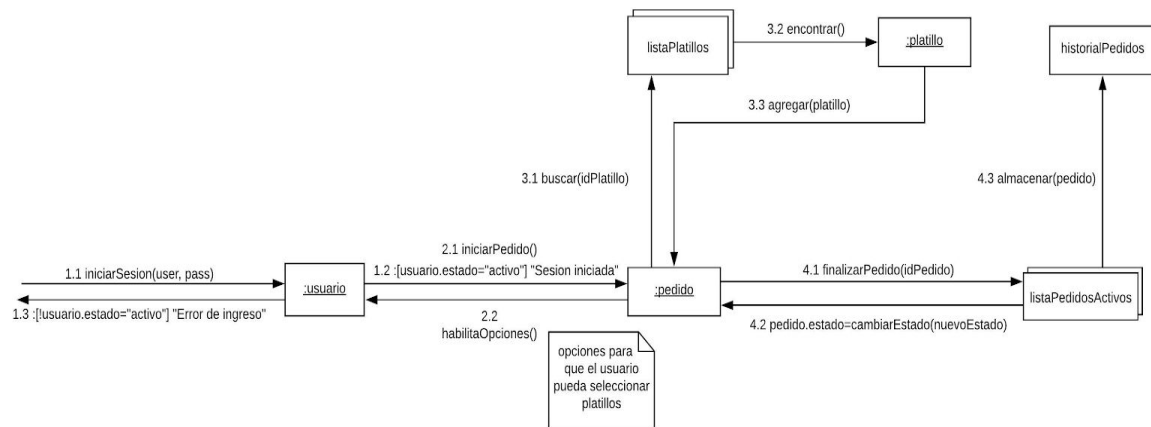
2.1.5.3 Contrato CO03: seleccionar

- **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.
- **Precondiciones:** Haber iniciado sesión y haber iniciado el pedido.
- **Poscondiciones:** Se crea una instancia de pedido, lista con los platillos seleccionados.

2.1.5.4 Contrato CO04: finalizarPedido

- **Nombre:** finalizarPedido(idPedido integer)
- **Responsabilidades:** Solicita el pago y lo valida, 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.
- **Precondiciones:** Haber iniciado sesión y haber seleccionado el/los platillos.
- **Poscondiciones:** Se actualiza la lista de pedidos activos, y se establece un enlace con historialPedidos.

2.1.6 Diagramas de colaboración



2.2 Atender pedido

2.2.1 Descripción

- **Caso de uso CU02:** Atender pedido.
- **Actores:** Administradores.
- **Propósito:** Responder a la solicitud de un cliente.
- **Precondición:** 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.
- **Referencias cruzadas:** RF1, RF2, RF3, RF7, RF6, RF8.

2.2.2 Flujo normal de eventos

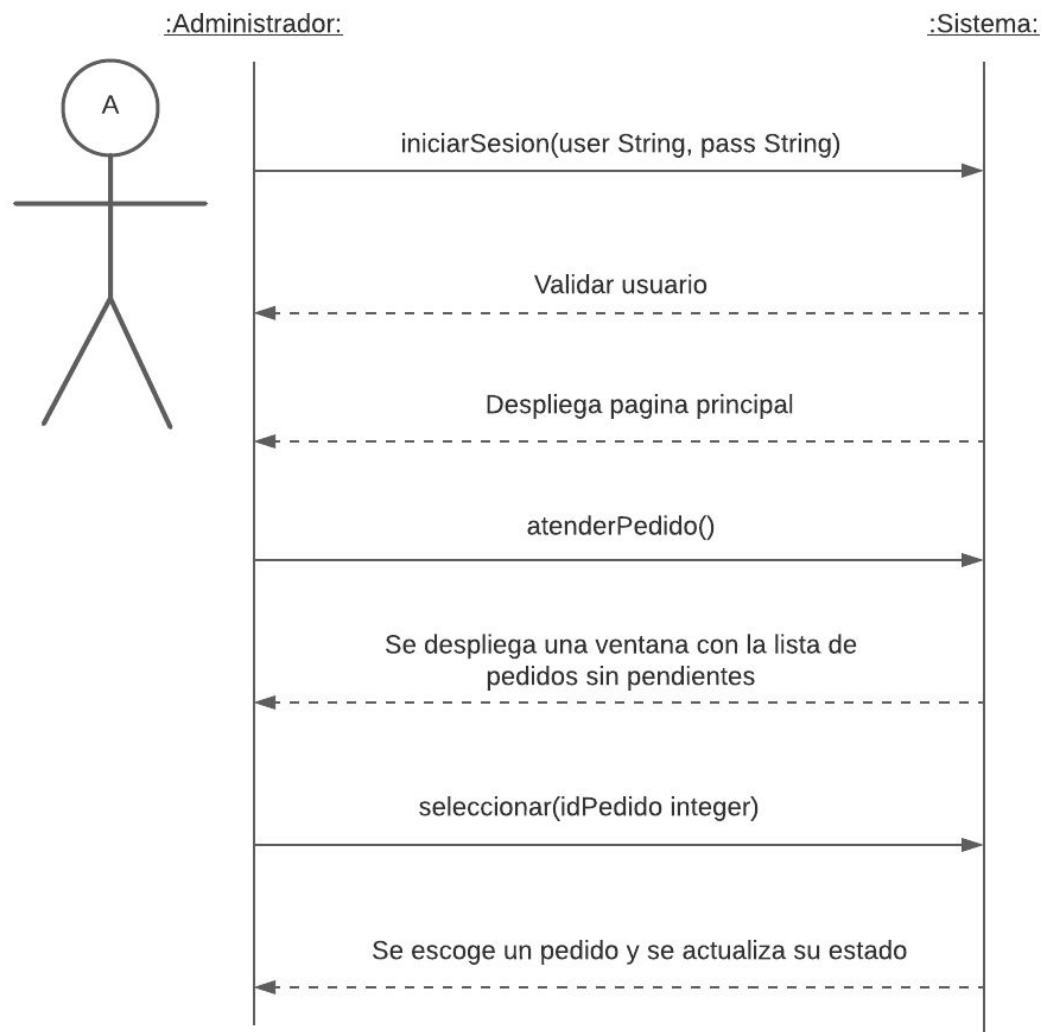
Acción de los actores	Respuesta del sistema
1. El administrador ingresa a la aplicación móvil.	
	2. El sistema solicita que inicie sesión.
3. El administrador inicia sesión.	
	4. El sistema valida al usuario y lo lleva a la ventana principal.
5. El administrador selecciona la opción de “Administrar pedidos”.	
	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.
13. Cuando el pedido es recibido por el cliente, se cambia el estado del pedido a "Entregado".	
	14. Almacena el pedido en un historial de pedidos.

2.2.3 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.)

2.2.4 Diagrama de secuencia



2.2.5 Contratos

2.2.5.1 Contrato CO01: iniciarSesion

- **Nombre:** iniciarSesion(user String, pass String).
- **Responsabilidades:** Autentifica al usuario y despliega la página principal de la aplicación.
- **Precondiciones:** 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.
- **Poscondiciones:** se modifica usuario.estado="activo".

2.2.5.2 Contrato CO02: atenderPedido

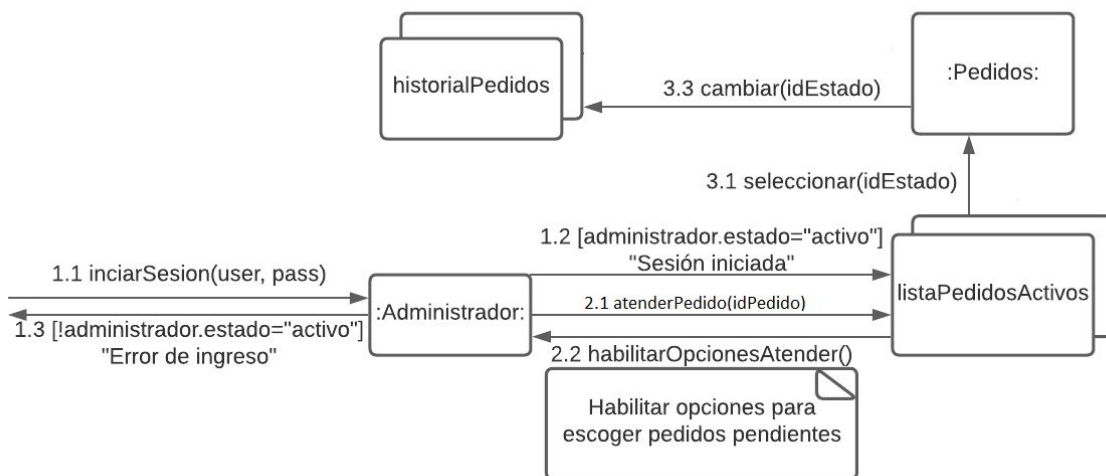
- **Nombre:** atenderPedido(idPedido).

- **Responsabilidades:** Despliega una ventana con todos los pedidos que han sido ingresados al sistema previamente.
- **Precondiciones:** 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.
- **Poscondiciones:** Se genera un enlace con la lista de pedidos pendientes. Se establece un enlace con historialPedidos.

2.2.5.3 Contrato CO03: seleccionar

- **Nombre:** seleccionar(idPedido integer).
- **Responsabilidades:** Se escoge un pedido, para leer los detalles y actualizar el estado a “En preparación” o “En camino”.
- **Precondiciones:** 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.
- **Poscondiciones:** Se actualiza la información en pantalla sobre “estado”.

2.2.6 Diagramas de colaboración



2.3 Agregar platillo

2.3.1 Descripción

- **Caso de uso CU03:** Agregar platillo.
- **Actores:** Administradores.
- **Propósito:** Actualizar la variedad de ofertas disponibles.
- **Precondición:** 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:** Primario.
- **Referencias cruzadas:** RF1, RF2, RF6, RF3

2.3.2 Flujo normal de eventos

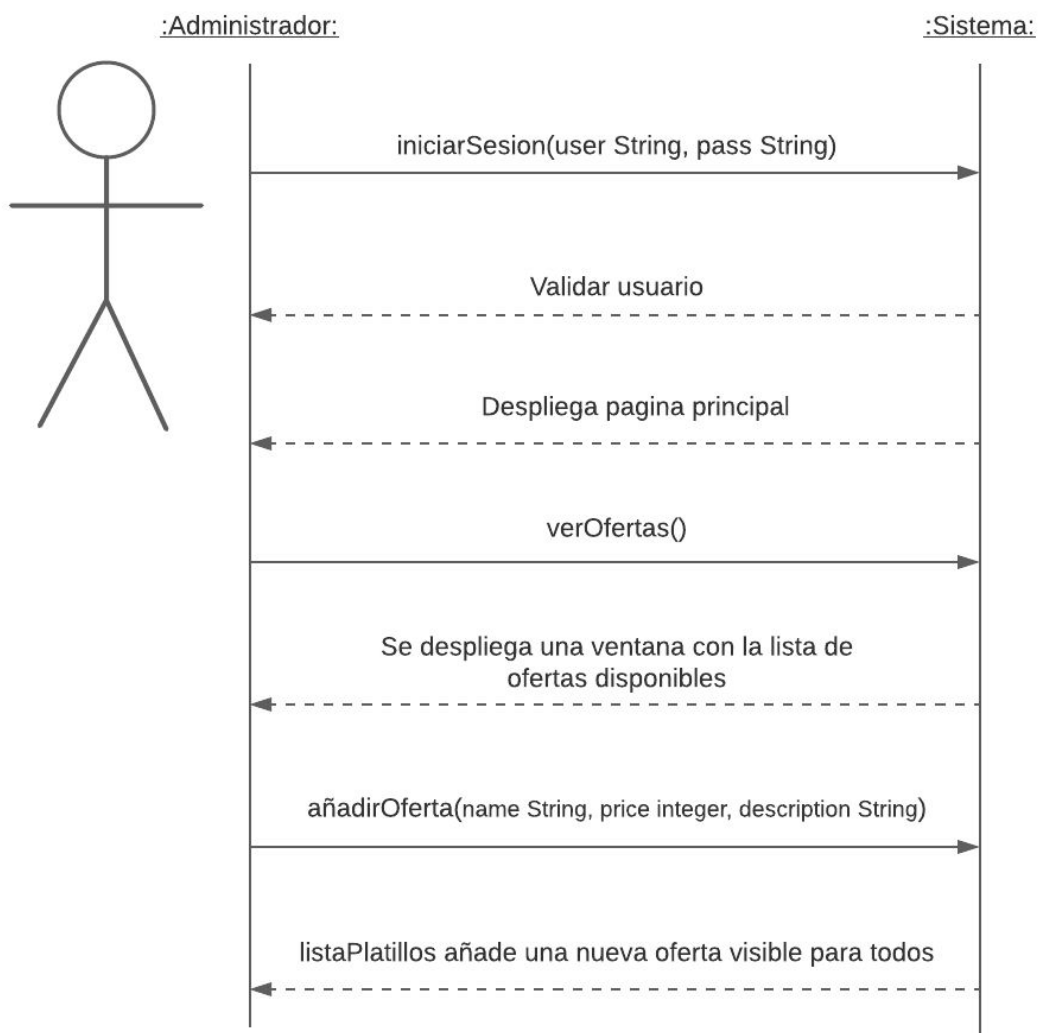
Acción de los actores	Respuesta del sistema
1. El administrador ingresa a la aplicación móvil.	
	2. El sistema solicita que inicie sesión.
3. El administrador inicia sesión.	
	4. El sistema valida al usuario y lo lleva a la ventana principal.
5. El administrador selecciona el botón "Administrar platillos".	
	6. El sistema despliega la lista de platillos disponibles a modificar.
7. El administrador escoge la opción de agregar platillo.	
	8. El sistema despliega una ventana que muestra los campos a llenar para agregar un nuevo platillo.
9. El usuario rellena toda la información pertinente (nombre, ingredientes, etc.)	

	10. El sistema valida el ingreso, añadiendolo a lista visible desde la aplicación.
--	--

2.3.3 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.

2.3.4 Diagrama de secuencia



2.3.5 Contratos

2.3.5.1 Contrato CO01: iniciarSesion

- **Nombre:** iniciarSesion(user String, pass String).
- **Responsabilidades:** Autentifica al usuario y despliega la página principal de la aplicación.
- **Precondiciones:** 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.
- **Poscondiciones:** se modifica usuario.estado="activo".

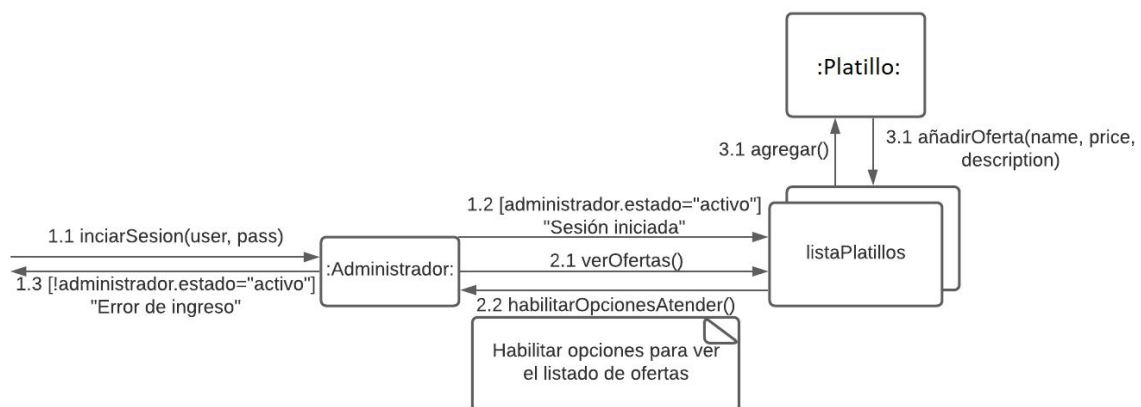
2.3.5.2 Contrato CO02: verOfertas

- **Nombre:** verOfertas().
- **Responsabilidades:** Se despliega la ventana con las ofertas disponibles.
- **Precondiciones:** 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.
- **Poscondiciones:** Se genera un enlace con listaPlatillos.

2.3.5.3 Contrato CO03: añadirOferta

- **Nombre:** añadirOferta(name String, price integer, description String)
- **Responsabilidades:** Agrega un platillo a listaPlatillos.
- **Precondiciones:** 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.
- **Poscondiciones:** listaPlatillos añade un platillo nuevo.

2.3.6 Diagramas de colaboración



2.4 Revisar su historial (cliente)

2.4.1 Descripción

- **Caso de uso:** Revisar su historial (cliente).
- **Actores:** Cliente.
- **Propósito:** Ver todas las compras realizadas por el cliente en el pasado.
- **Precondición:** 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.
- **Referencias cruzadas:** RF10, RF17

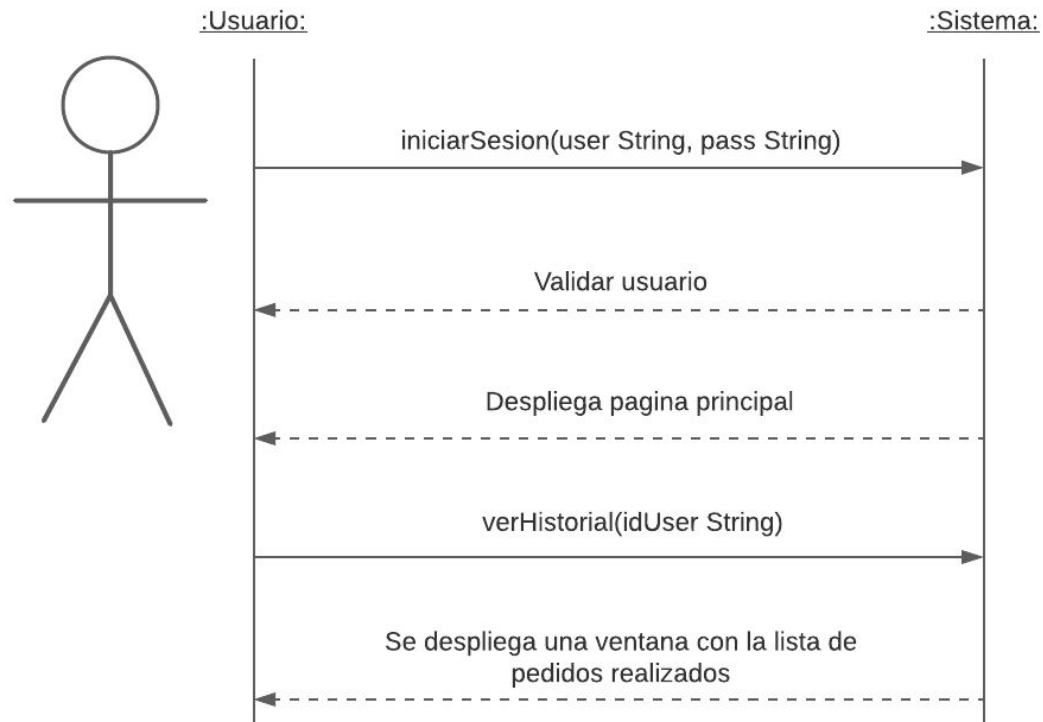
2.4.2 Flujo normal de eventos

Acción de los actores	Respuesta del sistema
1. El cliente ingresa a la aplicación móvil.	
	2. El sistema solicita que inicie sesión.
3. El cliente inicia sesión.	
	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.

2.4.3 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.

2.4.4 Diagrama de secuencia



2.4.5 Contratos

2.4.5.1 Contrato CO01: iniciarSesion

- **Nombre:** iniciarSesion(user String, pass String).
- **Responsabilidades:** Autentifica al usuario y despliega la página principal de la aplicación.
- **Precondiciones:** 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.
- **Poscondiciones:** se modifica usuario.estado="activo".

2.4.5.2 Contrato CO02: verHistorial

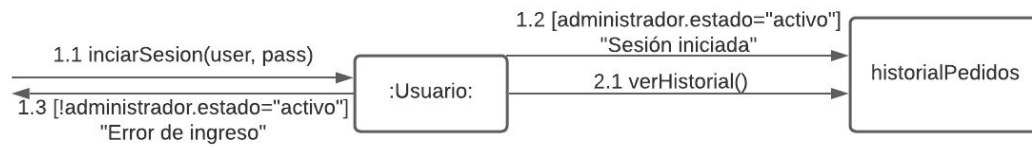
- **Nombre:** verHistorial(idUser String).
- **Responsabilidades:** Se despliega la ventana con el historial de pedidos del cliente.
- **Precondiciones:** El usuario debe haber realizado al menos un pedido para que el botón esté disponible.
- **Poscondiciones:** Se genera un enlace con historialPedidos.

2.4.5.4 Contrato CO03: verPedido

- **Nombre:** verPedido(idPedido String).

- **Responsabilidades:** Se despliega la ventana con los detalles de los pedidos.
- **Precondiciones:** El usuario debe tener acceso a su historial.
- **Poscondiciones:** -

2.4.6 Diagramas de colaboración



2.5 Fijar stock de platillos

2.5.1 Descripción

- **Caso de uso CU05:** Fijar stock de platillos.
- **Actores:** Administrador.
- **Propósito:** Fijar el stock diario de platillos que la empresa podrá preparar ese día.
- **Precondición:** El cliente debe tener una cuenta y haber iniciado sesión. Tiene que haber insumos suficientes, como para preparar la cantidad indicada. El platillo que se desee agregar, debe estar ingresado en el sistema.
- **Resumen:** El administrador abre la aplicación en su dispositivo móvil. El administrador inicia sesión con su usuario y contraseña, el sistema autentifica al administrador. El administrador selecciona la opción de “Fijar stock diario” para algún platillo. El administrador añade la cantidad máxima que va a poder venderse ese día por la aplicación.
- **Tipo:** Primario.
- **Referencias cruzadas:** RF1, RF6, RF7, RF20

2.5.2 Flujo normal de eventos

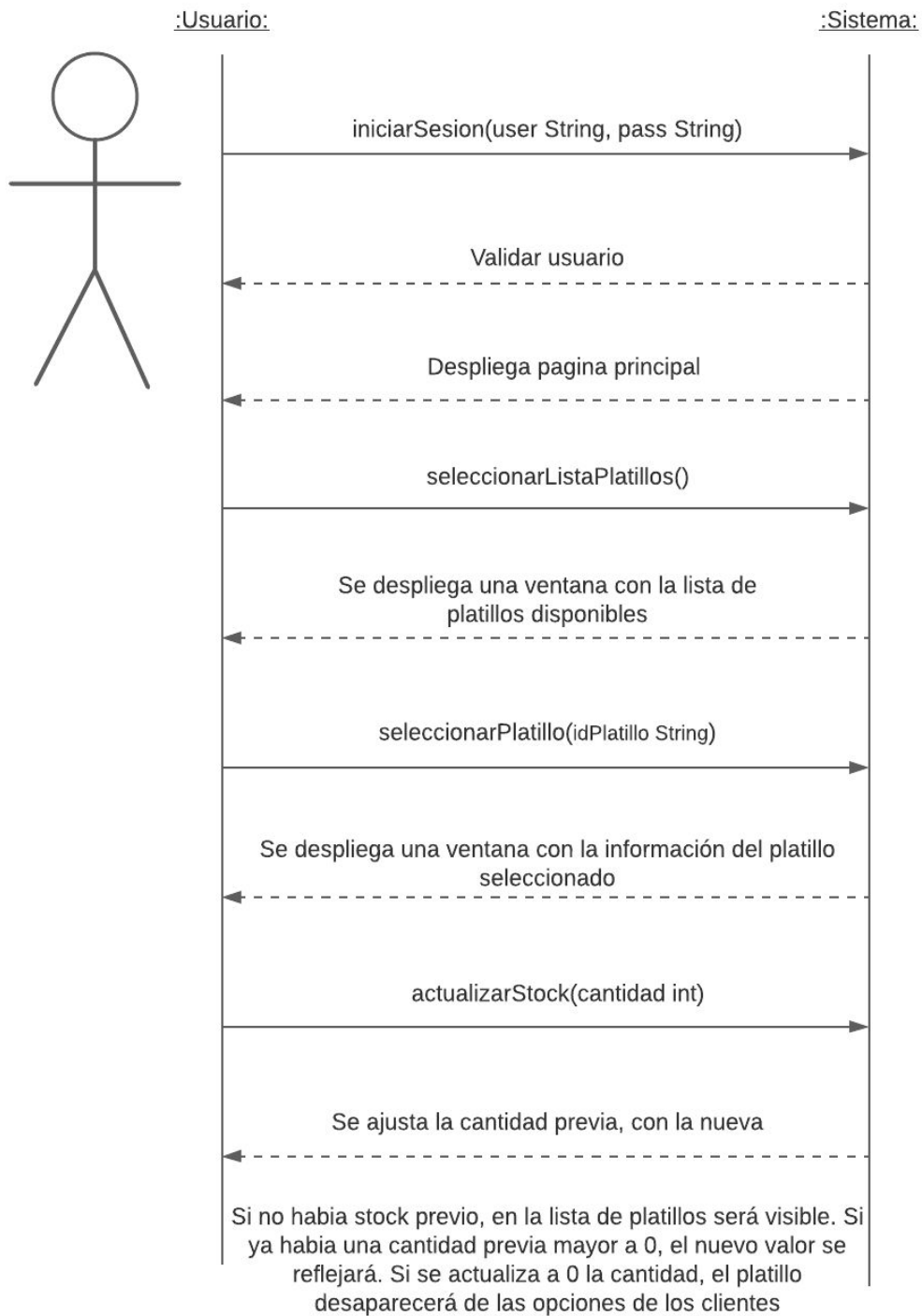
Acción de los actores	Respuesta del sistema
1. El administrador ingresa a la aplicación móvil.	
	2. El sistema solicita que ingrese su nombre de usuario y la contraseña.
3. El administrador ingresa los datos de nombre de usuario y contraseña.	
	4. El sistema valida al usuario y lo lleva a la ventana principal.
5. El administrador selecciona la lista de platillos en el sistema.	
	6. El sistema mostrará en pantalla los platillos en el sistema.
7. El administrador selecciona atender un platillo.	
	8. El sistema muestra las opciones para ese platillo.
9. El administrador fija una cantidad de stock.	

	10. El sistema actualiza el stock disponible.
--	---

2.5.3 Curso alternativo

- **Línea 2:** el administrador ingresa un nombre de usuario o una contraseña no válida, el sistema vuelve a solicitar que el usuario ingrese sus datos.
- **Línea 5:** la lista se encuentra vacía.

2.5.4 Diagrama de secuencia



2.5.5 Contratos

2.5.5.1 Contrato CO01: iniciarSesion

- **Nombre:** iniciarSesion(user String, pass String).
- **Responsabilidades:** Autentifica al usuario y despliega la página principal de la aplicación.
- **Precondiciones:** 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.
- **Poscondiciones:** se modificó usuario.estado="activo".

2.5.5.2 Contrato CO02: seleccionarListaPlatillos

- **Nombre:** seleccionarListaPlatillos().
- **Responsabilidades:** Muestra la lista de platillos en el sistema.
- **Precondiciones:** usuario.estado = "activo".
- **Poscondiciones:** Se actualizó la ventana principal del administrador y se desplegó la listas de platillos existentes en la aplicación.

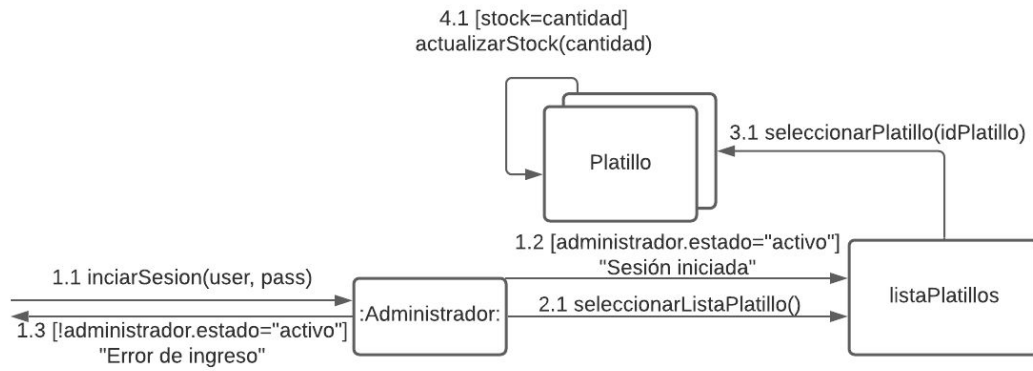
2.5.5.3 Contrato CO03: seleccionarPlatillo

- **Nombre:** seleccionarPlatillo(idPlatillo).
- **Responsabilidades:** Selecciona un platillo de las lista existente.
- **Precondiciones:** La lista de platillos no debe estar vacía, al menos un platillo registrado.
- **Poscondiciones:** Se seleccionó el objeto que coincide con idPlatillo.

2.5.5.4 Contrato CO04: actualizarStock

- **Nombre:** actualizarStock(cantidad).
- **Responsabilidades:** agregar o modificar un stock existente a un platillo para que sea visible a los usuarios.
- **Precondiciones:** el platillo debe estar seleccionado idPlatillo != "".
- **Poscondiciones:** Se modificó stock, si no existía stock previamente se visibilizó el platillo (platillo.stock>0).

2.5.6 Diagramas de colaboración



3. Diagrama de clases

