

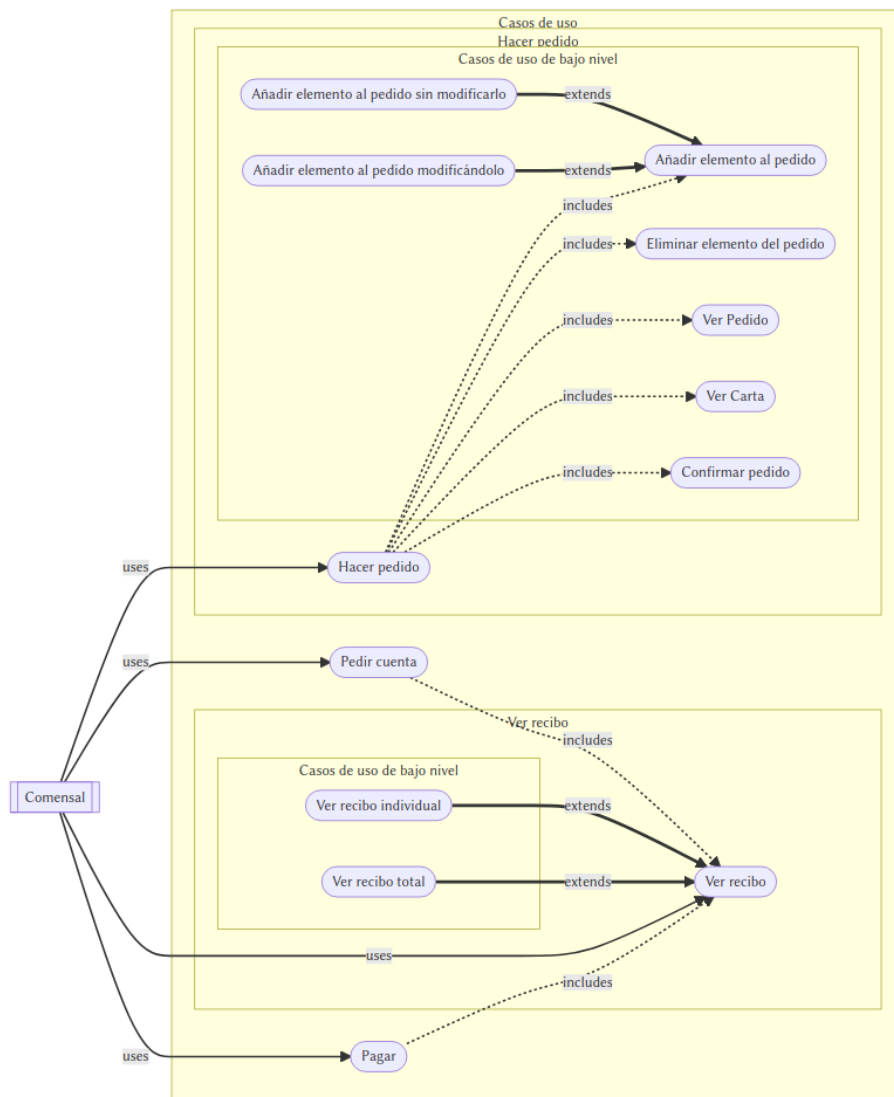
## Casos de Uso

Este documento recoge los casos de uso que contempla QRest en sus distintos ámbitos, desde el punto de vista de los usuarios desde la vista de la aplicación.

Diferenciamos en dos categorías de casos de uso, de alto y bajo nivel:

- **Casos de uso de alto nivel:** Aquellos del alto nivel de abstracción que se compone de otros casos de uso de bajo nivel.
- **Casos de uso de bajo nivel:** Aquellas acciones concretas que puede hacer un usuario en la aplicación.

Se representan en el siguiente diagrama:



## Glosario

### Actores

- **Empleado:** Entendido como una persona que opera el restaurante.
- **Comensal:** Dicho de un cliente que está sentado en la mesa que. Comparte pedido con los comensales de la misma mesa.

### Pantallas

- **Entrada:** Métodos para acceder a la pantalla en cuestión. Puede ser otra pantalla u otro método.
- **Salida:** Pantallas a las que se puede acceder desde la pantalla en cuestión.
- **Ruta:** Punto de acceso en la API.

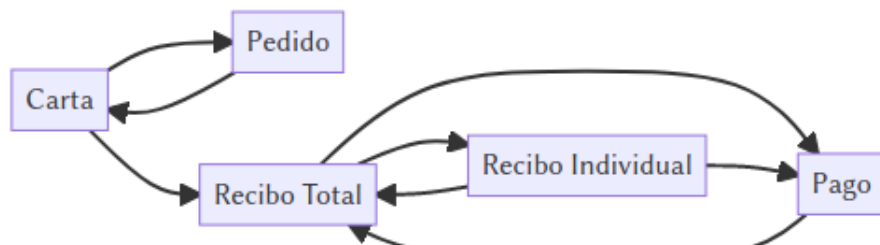
### Carta

**Elemento:** Cualquier bebida, entrante, plato, postre, etc. de la carta que puede añadirse al pedido.

**Elemento simple:** Aquel que no hay que elegir nada. Es decir, no tiene variantes, ni extras opcionales, ni se le pueden quitar ingredientes. Ejemplo: Nestea.

**Elemento complejo:** Aquel para el que hay que elegir algo. Tiene variantes o extras o se pueden eliminar ingredientes.

### Pantallas



### Carta

Es la **pantalla inicial**. Se ven los elementos de la carta del restaurante y pueden añadir o eliminar unidades de cualquier elemento de la carta. Todos los comensales de una misma mesa están asociados al mismo pedido y ven en tiempo real cualquier actualización (añadir o eliminar unidades de algún elemento) que haga cualquier otro comensal de la misma mesa.

- **Ruta:** mesa/{Identificador de la mesa}
- **Entrada:** QR, Pantalla Pedido.

- **Salidas:** Pantalla Pedido, Pantalla Recibo Total.

### **Pedido**

Se ven todos los elementos que han pedido los comensales de la mesa. Esto incluye el elemento concreto; con sus variantes, extras, e ingredientes eliminados, en caso de elementos complejos; y la cantidad.

- **Ruta:** mesa/{Identificador de la mesa}/pedido
- **Entrada:** Pantalla Carta.
- **Salidas:**
  - Carta.
  - Pedido Confirmado.

### **Recibo Total**

En esta pantalla se ve el recibo del pedido de toda la mesa en todas las comandas. Además del estdo de pago de cada elemento y quién lo ha pedido.

- **Ruta:** mesa/{Identificador de la mesa}/recibo
- **Entrada:** Pantalla Carta.
- **Salidas:**
  - Recibo Individual.
  - Pago.

### **Recibo Individual**

En esta pantalla se ve el recibo del total del comensal en todas las comandas. Además del estdo de pago de cada elemento y quién lo ha pedido.

- **Ruta:** mesa/{Identificador de la mesa}/recibo/{comensal}
- **Entrada:** Pantalla Recibo total.
- **Salidas:**
  - Recibo Total.
  - Pago.

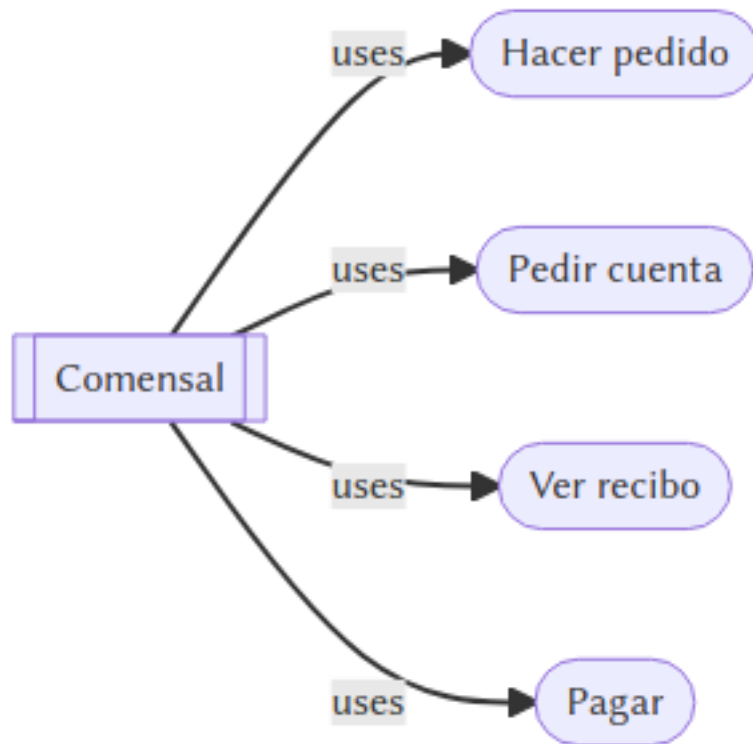
### **Pago**

En esta pantalla se tramita el pago.

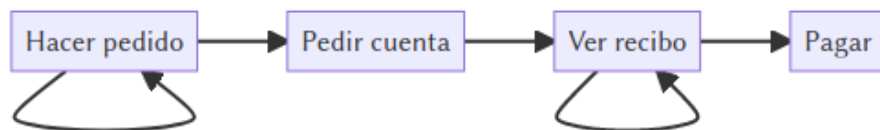
- **Ruta:** mesa/{Identificador de la mesa}/pago/{comensal}
- **Entrada:** Pantalla Recibo total.
- **Salida:** Recibo Total.

### **Casos de uso**

Los casos de uso de alto nivel son:

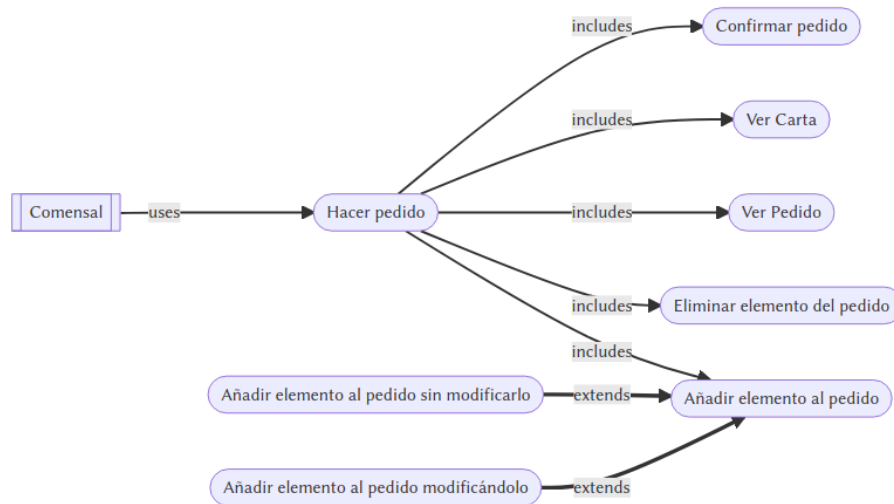


Que se representan en el siguiente diagrama de flujo:



#### Casos de uso de *Hacer pedido*

El caso de uso de alto nivel *Hacer pedido* se representa en el siguiente diagrama:



### Hacer pedido

- **Precondición:** Pantalla Carta.
- **Postcondición de éxito:**

### Escenario Principal:

1. El comensal navega por la carta.
2. El comensal ejecuta *Añadir un elemento al pedido* o *Eliminar un elemento del pedido*.
3. El comensal ejecuta *Ver pedido*.
4. El comensal ejecuta *Confirmar pedido* de forma satisfactoria.

### Escenarios Alternativos:

3a. El comensal vuelve a modificar el peiddo:

1. Vuelve al paso 2.

4a. El comensal vuelve a la carta:

1. El comensal selecciona *Volver a la carta*.
2. El sitema ejecuta *Ver Carta*.
3. Vuelve al paso 1.

4b. El comensal modifica el peido.

1. El comensal ejecuta *Añadir un elemento sin modificar* o *Eliminar un elemento del pedido*.
2. Vuelve al paso 4.

### Ver Pedido

- **Precondición:** Pantalla Carta.
- **Postcondición:** Pantalla Pedido.

#### Escenario principal:

1. El comensal selecciona *Ver pedido*.
2. El sistema redirige a la pantalla *Pedido*.
3. El sistema hace una llamada a la *API* para recoger los datos del *Pedido*.
4. El sistema genera el HTML de la pantalla *Pedido*.
5. El JS del HTML resalta los elementos que han sido pedidos por el comensal.
6. El sistema muestra al comensal la pantalla de *Pedidos*.

### Ver Carta

- **Precondición:** Ninguna.
- **Postcondición:** Pantalla Carta.

#### Escenario principal:

1. El sistema redirige a la pantalla *Carta*.
2. El sistema hace una llamada a la *API* para recoger los datos de la *Carta*.
3. El sistema hace una llamada a la *API* para recoger los datos del *Pedido*.
4. El sistema genera el HTML de la pantalla *Carta*.
5. El JS del HTML filtra segun alérgenos del comensal y resalta las sugerencias.
6. El sistema muestra al comensal la pantalla de *Carta*.

### Confirmar Pedido

- **Precondición:** Pantalla Pedido.
- **Postcondición de éxito:** Pantalla Carta.

#### Escenario principal:

1. El comensal selecciona *Confirmar pedido*.
2. El sistema muestra un modal para pedirle al usuario que confirme la decisión.
3. El comensal selecciona *Aceptar*.
4. El sistema llama a la *API* para confirmar el pedido.
5. El sistema ejecuta *Ver Carta*.

#### Escenario alternativo:

- 3a. El comensal cancela la confirmación del pedido.
  1. El comensal selecciona *Cancelar*.
  2. El sistema cierra el modal. (Sigue en la pantalla *Pedido*).

### Añadir elemento al pedido sin modificar

- **Precondición:** Pantalla Carta ó Pedido.
- **Postcondiciones:**
  - Se ha añadido una unidad del elemento al pedido.
  - Se ha incrementado una unidad de ese elemento en las vistas de los comensales.

### Escenario principal:

1. El comensal selecciona ‘+’ sobre el elemento a añadir.
2. El sistema llama a la API para añadir el elemento al pedido con la información del elemento y el identificador del cliente.
3. El JS incrementa en 1 la cantidad de ese elemento.
4. El sistema retransmite al JS del resto de comensales la petición para que incrementen en 1 ese elemento.

### Añadir elemento al pedido modificándolo

- **Precondición:** Pantalla Carta.
- **Postcondiciones de éxito:**
  - Se ha añadido **cantidad** unidades del elemento al pedido.
  - Se ha incrementado en **cantidad** unidades de ese elemento en las vistas de los comensales.

### Escenario principal:

1. El comensal selecciona ‘+’ sobre el elemento a añadir.
2. El sistema muestra un modal para seleccionar las variantes, extras e ingredientes.
3. El comensal selecciona las variantes que quiera (si las hay), selecciona los extras que quiera (si los hay) y deselecciona los ingredientes no deseados (si los hay).
4. El comensal introduce la cantidad.
5. El comensal selecciona *Aceptar*.
6. El sistema llama a la API para añadir el elemento al pedido con la información del elemento, la cantidad y el identificador del cliente.
7. El JS incrementa en **cantidad** la cantidad de ese elemento.
8. El sistema retransmite al JS del resto de comensales la petición para que incrementen en **cantidad** ese elemento.

### Escenario alternativo:

[3-5]a. El comensal cierra el modal.

1. El comensal selecciona *Cancelar*.
2. El sistema cierra el modal. (Sigue en la pantalla Carta).

### Eliminar elemento del pedido



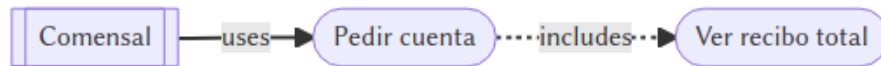
- **Precondición:** Pantalla Carta ó Pedido.
- **Postcondiciones:**
  - Se ha eliminado una unidad del elemento del pedido.
  - Se ha decrementado en una unidad ese elemento en las vistas de los comensales.

**Escenario principal:**

1. El comensal selecciona '-' sobre el elemento a eliminar.
2. El sistema llama a la API para eliminar el elemento al pedido con la información del elemento y el identificador del cliente.
3. El JS decrementa en 1 la cantidad de ese elemento.
4. El sistema retransmite al JS del resto de comensales la petición para que decrementen en 1 ese elemento.

**Casos de uso de *Pedir cuenta***

El caso de uso de alto nivel \_Pedir cuenta se representa en el siguiente diagrama:



**Pedir cuenta**

- **Precondiciones:**
  - Pantalla Carta.
  - Al menos una comanda (pedido) confirmado.
- **Postcondiciones de éxito:** Se ha generado el recibo.

**Escenario principal:**

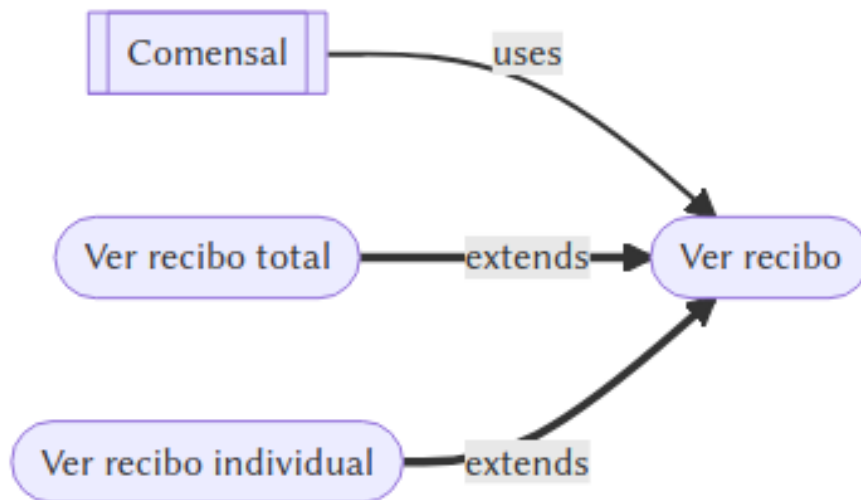
1. El comensal selecciona *Pedir Cuenta*.
2. El sistema muestra un modal para confirmar la decisión.
3. El comensal selecciona *Aceptar*.
4. El sistema llama a la *API* para generar el recibo.
5. El sistema ejecuta *Ver recibo total*.

**Escenario alternativo:**

- 3a. El comensal no pide la cuenta.
  1. El comensal selecciona *Cancelar*.
  2. El sistema cierra el modal. (Sigue en la pantalla Carta).

**Casos de uso de *Ver recibo***

El caso de uso de alto nivel *Ver recibo* se representa en el siguiente diagrama:



**Ver recibo total** **Precondición:** Se ha ejecutado *Generar recibo*. **Post-condición:** Pantalla Recibo total.

**Escenario principal:**

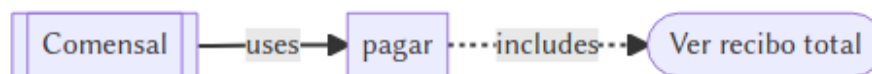
1. El sistema llama a la *API* para obtener la información del recibo.
2. El sistema genera el HTML de la pantalla *Recibo total*.
3. El JS del HTML resalta los elementos pagados.
4. El sistema muestra al comensal la pantalla *Recibo total*.

**Ver recibo total** **Precondición:** Se ha ejecutado *Generar recibo*. **Post-condición:** Pantalla Recibo individual.

**Escenario principal:**

1. El sistema llama a la *API* para obtener la información del recibo del comensal.
2. El sistema genera el HTML de la pantalla *\_Recibo individual*.
3. El sistema muestra al comensal la pantalla *\_Recibo individual*.

**Casos de uso de *Pagar***



**Pagar** **Precondición:** Pantalla Recibo total o individual. **Postcondición de éxito:** Se han pagado los elementos correspondientes.

**Escenario principal:**

1. El comensal selecciona *Pagar*.
2. El sistema redirige a la pasarela de pago.
3. El comensal rellena los datos bancarios necesarios.
4. El comensal selecciona *Pagar*.
5. La pasarela de pago comprueba que los datos sean correctos.
6. Los bancarios son correctos y se puede realizar el pago.
7. La pasarela de pago tramita el pago.
8. La pasarela de pago hace una llamada a la *API* para confirmar el pago.
9. El sistema marca los elemento correspondientes como pagado.
10. El sistema ejecuta *Ver recibo total*.

**Escenario alternativo:**

[3-4] Se cancela el pago.

1. El comensal cancela el pago.
  2. El sistema ejecuta *Ver recibo total*.
- 5a. Los datos bancarios no son correctos.
1. Los datos bancarios introducidos no son correctos.
  2. La pasarela de pago informa de los errores.
  3. Se vuelve al paso 3.