



---

# RESTAURANTE LA CASA DORADA

---

Requerimientos Funcionales



29 DE MARZO DE 2021

UNIVERSIDAD ICESI

Santiago Arévalo Valencia – Alexander Sánchez Sánchez

# REQUERIMIENTOS FUNCIONALES

El sistema debe estar en la capacidad de:

## Req1.

**Gestionar** todos los productos con un nombre, un tipo (plato principal, adicional o bebida, o alguna otra variante), un conjunto de ingredientes, disponibilidad y tamaños con sus respectivos precios que ofrece el restaurante. Todo producto en el sistema podrá ser eliminado siempre que no esté referenciado desde otro objeto.

- ◆ **Req1.1** Crear un producto junto con el usuario que creó el mismo.
- ◆ **Req1.2** Actualizar un producto preexistente junto con el usuario que lo modificó.
- ◆ **Req1.3** Eliminar un producto preexistente.
- ◆ **Req1.4** Deshabilitar un producto preexistente. Esto es, en esencia, que no podrá ser usado (referenciado) por algún otro objeto y, por tanto, no estará disponible.
- ◆ **Req1.5** Habilitar un producto previamente deshabilitado. Es decir, cambiará su estado de no disponible a disponible y podrá ser usado nuevamente por algún otro objeto.

## Req2.

**Gestionar** todos los ingredientes con un nombre y disponibilidad. Todo ingrediente en el sistema podrá ser eliminado siempre que no esté referenciado desde otro objeto.

- ◆ **Req2.1** Crear un ingrediente junto con el usuario que creó el mismo.
- ◆ **Req2.2** Actualizar un ingrediente preexistente junto con el usuario que lo modificó.
- ◆ **Req2.3** Eliminar un ingrediente preexistente.
- ◆ **Req2.4** Deshabilitar un ingrediente preexistente. Esto es, en esencia, que no podrá ser usado (referenciado) por algún otro objeto (producto) y, por tanto, no estará disponible.
- ◆ **Req2.5** Habilitar un ingrediente previamente deshabilitado. Es decir, cambiará su estado de no disponible a disponible y podrá ser usado nuevamente por algún otro objeto (producto).

## Req3.

**Gestionar** todos los tipos de producto con un nombre y disponibilidad. Todo tipo de producto en el sistema podrá ser eliminado siempre que no esté referenciado desde otro objeto.

- ◆ **Req3.1** Crear un tipo de producto junto con el usuario que creó el mismo.
- ◆ **Req3.2** Actualizar un tipo de producto preexistente junto con el usuario que lo modificó.
- ◆ **Req3.3** Eliminar un tipo de producto preexistente.
- ◆ **Req3.4** Deshabilitar un tipo de producto preexistente. Esto es, en esencia, que no podrá ser usado (referenciado) por algún otro objeto (producto) y, por tanto, no estará disponible.
- ◆ **Req3.5** Habilitar un tipo de producto previamente deshabilitado. Es decir, cambiará su estado de no disponible a disponible y podrá ser usado nuevamente por algún otro objeto (producto).

## Req4.

**Gestionar** todos los clientes con nombres, apellidos, un número de identificación (opcional), dirección, teléfono y un campo de observaciones y disponibilidad. Todo cliente en el sistema podrá ser eliminado siempre que no esté referenciado desde otro objeto.

- ◆ **Req4.1** Crear un cliente junto con el usuario que creó el mismo.
- ◆ **Req4.2** Actualizar un cliente preexistente junto con el usuario que lo modificó.
- ◆ **Req4.3** Eliminar un cliente preexistente.
- ◆ **Req4.4** Deshabilitar un cliente preexistente. Esto es, en esencia, que no podrá ser usado (referenciado) por algún otro objeto y, por tanto, no estará disponible.
- ◆ **Req4.5** Habilitar un cliente previamente deshabilitado. Es decir, cambiará su estado de no disponible a disponible y podrá ser usado nuevamente por algún otro objeto.

## Req5.

**Gestionar** todos los empleados con nombres, apellidos, un número de identificación y disponibilidad. Todo empleado en el sistema podrá ser eliminado siempre que no esté referenciado desde otro objeto.

- ◆ **Req5.1** Crear un empleado junto con el usuario que creó el mismo.
- ◆ **Req5.2** Actualizar un empleado preexistente junto con el usuario que lo modificó.
- ◆ **Req5.3** Eliminar un empleado preexistente.
- ◆ **Req5.4** Deshabilitar un empleado preexistente. Esto es, en esencia, que no podrá ser usado (referenciado) por algún otro objeto y, por tanto, no estará disponible.

- ◆ **Req5.5** Habilitar un empleado previamente deshabilitado. Es decir, cambiará su estado de no disponible a disponible y podrá ser usado nuevamente por algún otro objeto.

## Req6.

**Gestionar** todos los usuarios con nombres, apellidos, un número de identificación, un nombre de usuario, una contraseña para ingresar al sistema y disponibilidad. Todo usuario en el sistema podrá ser eliminado siempre que no esté referenciado desde otro objeto. Todo usuario también es un empleado.

- ◆ **Req6.1** Crear un usuario junto con el usuario que creó el mismo.
- ◆ **Req6.2** Actualizar un usuario preexistente junto con el usuario que lo modificó.
- ◆ **Req6.3** Eliminar un usuario preexistente.
- ◆ **Req6.4** Deshabilitar un usuario preexistente. Esto es, en esencia, que no podrá ser usado (referenciado) por algún otro objeto y, por tanto, no estará disponible.
- ◆ **Req6.5** Habilitar un usuario previamente deshabilitado. Es decir, cambiará su estado de no disponible a disponible y podrá ser usado nuevamente por algún otro objeto.

## Req7.

**Gestionar** todos los pedidos con un código (autogenerado) de pedido, un estado, un listado de productos, la cantidad por cada uno, el cliente que los solicita, el empleado que lo entrega, la fecha y hora de la solicitud (de tipo Date), y las observaciones que pueda tener y disponibilidad. Al momento de registrarse el pedido, el programa debe tomar la fecha y hora de la solicitud, de la fecha y hora actual del sistema. Incluso, usted tendrá una pequeña bonificación si muestra permanentemente actualizada la fecha y hora actual en alguna parte de la ventana. Todo pedido en el sistema podrá ser eliminado siempre que no esté referenciado desde otro objeto.

- ◆ **Req7.1** Crear un pedido junto con el usuario que tomó el mismo.
- ◆ **Req7.2** Actualizar un pedido preexistente junto con el usuario que lo modificó.
- ◆ **Req7.3** Eliminar un pedido preexistente.

- ◆ **Req7.4** Deshabilitar un pedido preexistente. Esto es, en esencia, que no podrá ser usado (referenciado) por algún otro objeto y, por tanto, no estará disponible
- ◆ **Req7.5** Habilitar un pedido previamente deshabilitado. Es decir, cambiará su estado de no disponible a disponible y podrá ser usado nuevamente por algún otro objeto.

## Req8.

Insertar un cliente al listado de clientes de forma ordenada alfabéticamente descendente.

## Req9.

**Generar** reportes de los objetos principales.

- ◆ Req9.1 Generar reporte de los productos exportando un archivo csv con la información del listado de estos.
- ◆ Req9.2 Generar reporte de clientes exportando un archivo csv con la información del listado de estos.
- ◆ Req9.3 Generar reporte de pedidos exportando un archivo csv con la información del listado de estos.
- ◆ Req9.4 Generar reporte de empleados exportando un archivo csv con la información del listado de estos

## Req10

**Importar** la información de los listados principales

- ◆ Req10.1 Importar un archivo cvs con la información de productos
- ◆ Req10.2 Importar un archivo cvs con la información de clientes
- ◆ Req10.3 Impirtar un archivo cvs con la información de pedidos

## Req11

**Cargar** datos del programa (lista de productos, lista de empleados, lista de clientes, lista de ingredientes, lista de usuarios) a través de la serialización.

## Re12

**Listar** objetos a los usuarios

- ◆ Req12.1 Listar todos los productos en orden de precio ascendente

- ◆ Req12.2 Listar todos los ingredientes en orden alfabético descendente.

Cada vez que se mencione el verbo **gestionar** implica las operaciones: crear un nuevo objeto, actualizar un objeto existente, eliminar o deshabilitar.



