



## **TAREA INTEGRADORA # 1:**

### **La casa Dorada**



## **ALGORITMOS Y PROGRAMACIÓN II**

**Grupo: 03**

**Integrantes:**

**Camilo González Velasco - A00370263**

**Johan Stiven Ricardo Sibaja - A0037069**

**DOCENTE: Juan Manuel Reyes G.**

**Semestre 2021 - A**

## REQUERIMIENTOS FUNCIONALES

*El sistema debe estar en la capacidad de:*

**Req.1. Gestionar** usuarios del sistema con nombre, contraseña e identificación. No permitirá que se agreguen usuarios con el mismo usuario de sistema y se agregan también como empleados.

**Req.1.1. Agregar** un nuevo usuario con los datos de nombre , apellido, identificación, usuario de sistema y contraseña a la lista de usuarios y empleados del restaurante.

**Req.1.2. Actualizar** la información del usuario que se referencie.

**Req.1.3. Deshabilitar** el acceso a la información del usuario que se referencie.

**Req.1.4. Eliminar** el usuario al que se esté referenciando de la lista de usuarios, si es eliminado de la lista de empleados también deberá ser eliminado de la lista de usuarios.

**Req.2. Gestionar** clientes con nombre, contraseña , identificación , dirección y teléfono. No se permitirán agregar clientes con la misma identificación. No podrán ser eliminados o deshabilitados mientras sean referenciados por un pedido o por un usuario, ya sea el usuario que lo registró o el que modificó su información.

**Req.2.1. Agregar** un nuevo cliente con todos sus datos a la lista de clientes del restaurante.

**Req.2.2. Actualizar** cualquier dato del cliente que se seleccione haciendo doble clic sobre el registro de la lista.

**Req.2.3. Deshabilitar** el acceso a la información del cliente que se seleccione.

**Req. 2.4. Eliminar** el cliente que se seleccione de la lista de clientes.

**Req.3. Gestionar** empleados con nombre , apellido , identificación y una cantidad de órdenes entregadas. No se permitirán agregar empleados con la misma identificación y no podrán ser eliminados o deshabilitados mientras sean referenciados por un pedido o por un

usuario, ya sea el usuario que registró al empleado o el usuario que modificó su información.

**Req.3.1. Agregar** un nuevo empleado con todos sus datos a la lista de empleados del restaurante.

**Req.3.2. Actualizar** cualquier dato que se desee del empleado haciendo doble clic en el registro de la lista.

**Req 3.3. Deshabilitar** el acceso a la información del usuario que se seleccione.

**Req 3.4. Eliminar** el empleado que se seleccione de la lista de empleados.

**Req.4. Gestionar todos los productos que ofrece el restaurante.** Esto implica las operaciones: crear un nuevo objeto, actualizar un objeto existente, eliminar o deshabilitar. No se podrán gestionar productos u objetos que no estén creados o que no entren bajo la siguiente condición que permita coordinar al mismo. Esa condición es: *todo objeto* en el sistema podrá ser eliminado siempre que no esté referenciado desde otro objeto.

**Req.4.1. Agregar** un nuevo producto. Todo producto tiene un nombre, un tipo (plato principal, adicional o bebida -pero estos pueden variar-), un conjunto de ingredientes, y tamaños con sus respectivos precios.

**Req.4.2. Actualizar** la información de los productos existentes haciendo doble clic en el registro de la lista.

**Req.4.3. Deshabilitar** un producto siempre será posible y esto implica que ese producto no estará disponible, en adelante, para ser referenciado desde otro objeto. si se deshabilita un producto, este ya no se mostrará entre los productos disponibles al momento de hacer un pedido.

**Req.4.4. Eliminar** los productos existentes

**Req.5. Gestionar todos los tipos de producto que ofrece el restaurante.** Esto implica las operaciones: crear un nuevo objeto, actualizar un objeto existente, eliminar o deshabilitar. No se podrán gestionar productos u objetos que no estén creados o que no entren bajo la

siguiente condición que permita coordinar al mismo. Esa condición es: *todo objeto* en el sistema podrá ser eliminado siempre que no esté referenciado desde otro objeto.

**Req.5.1. Agregar** un nuevo tipo de producto a la lista de tipos del restaurante

**Req.5.2. Actualizar** la información de los productos existentes haciendo doble clic en el registro de la lista.

**Req.5.3. Deshabilitar** el acceso al tipo de producto que se seleccione y a su información.

**Req.5.4. Eliminar** los tipos de productos existentes de la lista de tipos de producto del restaurante.

**Req.6. Gestionar** ingredientes con un nombre. No se permitirá agregar ingredientes con el mismo nombre. No se podrá eliminar o deshabilitar un ingrediente mientras sea referenciado por un producto , por el usuario que lo creó o el que actualizó su información.

**Req.6.1. Agregar** un nuevo ingrediente a la lista de ingredientes del restaurante.

**Req.6.2. Actualizar** la información del ingrediente que se seleccione.

**Req.6.3. Deshabilitar** el acceso al ingrediente que se seleccione y a su información.

**Req.6.4. Eliminar** el ingrediente que seleccione de la lista de ingredientes del restaurante.

**Req.7 Gestionar** Pedidos con un código autogenerado, una dirección, un empleado encargado de entregarlo , un cliente que es el que está haciendo el pedido, una lista de productos solicitados por el cliente y puede tener o no comentarios.

**Req.7.1. Agregar** un nuevo pedido con las referencias a los objetos que lo componen a la lista de pedidos del restaurante.

**Req.7.2. Actualizar** el estado de un pedido de manera ascendente; es decir, comienza con un estado de “REQUESTED” y va avanzando hasta “DELIVERED”, pero nunca de “DELIVERED” hacia “REQUESTED”.

**Req.7.3. Deshabilitar** el acceso al pedido que se seleccione y a la información del mismo.

**Req.7.4. Eliminar** un pedido de la lista de pedidos del restaurante

**Req.8. Listar** los principales campos de los de cada objeto del programas. Se deben mostrar las principales características de los productos, pedidos, clientes , usuarios, tipo de productos y empleados.

**Req. 8.1. Listar** en pantalla todos los productos en orden de precio ascendente.

**Req.8.2. Listar** en pantalla todos los ingredientes en orden alfabético descendente.

**Req.9. Importar** información de clientes, productos y pedidos con la suficiente información para crear los objetos con los que estén relacionados, por ejemplo, en el archivo de pedido debe haber información suficiente para crear un nuevo empleado, cliente, y producto.

**Req.10. Generar** dos reportes. Un listado de empleados consolidando el número de pedidos entregados y la suma de los valores de dichos pedidos. Esto en un rango de fecha y hora, inicial y final. Por defecto, el valor en la fecha y hora inicial son las 00:00 del día actual, y la fecha final debe ser por defecto 23:59 del día actual. Un listado de los productos consolidando el número de veces que se pidió y la cantidad de dinero total que se pagó por todos los productos (número de veces por precio del producto). Este reporte también debe solicitar un rango de fechas inicial y final igual que el anterior, con los mismos valores por defecto. Estas fechas y horas siempre se podrán cambiar. En los dos reportes se debe agregar una fila al final que totalice las columnas numéricas.

**Req.11. Buscar** eficientemente un cliente dado un nombre e indicar el tiempo que tardó la búsqueda. Esta opción debe estar en el campo que permite realizar los pedidos. Se indicará si el cliente fue encontrado o no, en caso de que no lo encuentre se indica que se verifique el nombre que se escribió.

## DIAGRAMA DE CLASES

[<https://online.visual-paradigm.com/share.jsp?id=313235343538392d31#diagram:workspace=iedetnlc&proj=0&id=1>]