

# Gestión de Stock para la restauración

## Introducción

### ¿Cuál es su funcionamiento?

Gestionar de manera automática el inventario que hay en el almacén de un restaurante y que automáticamente cree una lista de todo lo que es necesario comprar también de podría ver el stock actual por si se tiene una previsión muy fuerte de trabajo hacer un pedido mayor o en caso de que haya habido un problema de stock por un plato mal hecho, que sea erróneo o que no sea para venta al público.

### ¿Funcionamiento interno de la Base de datos?

Para realizar este sistema de gestión de datos primero deberemos de introducir los datos por cada plato -> ingredientes, dentro de ingredientes -> cantidad en gramos o mililitros que lleva y los alérgenos de estos, también para cada ingrediente necesitaremos saber cuál es el proveedor que lo trae. Será necesario saber la capacidad máxima de producto que cabe para que no haya más de lo que se puede guardar, y necesitaremos saber la cantidad de producto que hay por paquete, bolsa o caja según cómo sea el pedido mínimo.

Con las bebidas también lo necesitaremos hacer en este caso es más sencillo por que a la bebida solo le añadiremos el proveedor y la capacidad máxima que caben para no hacer un sobre pedido también hay que definir la capacidad de bebidas por cada caja.

Una vez hecho esto nuestro sistema guardará las veces que se vende X producto para que una vez llegue al pedido mínimo lo guarde en un listado el cual se actualiza constantemente para posteriormente ver el pedido que hay que hacer.

En el caso de las bebidas es mucho más sencillo por que una bebida es una unidad pero hay que definir el stock por cajas para que una vez se llega a la cantidad de cada caja marque que hay que pedir una.

## Objetivo

Nuestro objetivo es facilitar el trabajo de tener que estar semanalmente revisando stocks en almacén cámaras frigoríficas y neveras, también gracias a este proyecto el mantenimiento del stock se hace de manera mucho más efectiva ya que a la hora de revisarlo manualmente siempre se puede escapar algo en este caso nunca fallará ya y siempre los stocks estarán justos y no sería necesario realizar dos pedidos a una misma empresa por culpa de un mal conteo y por último ganaría muchísima agilidad a la hora de trabajar y ese tiempo perdido en el stock se podría invertir en cualquier otra cosa.

## Metodología

Nuestra metodología de este proyecto consiste en a través de cada pedido nuevo y cada vez que se acabe algún producto del stock, que se vaya actualizando nuestra base de datos para tenerlo todo al día organizado y saber que es lo que falta con exactitud sin tener que llevar una gestión difícil de llevar.

## Resultados

Una vez hecho el primer prototipo hemos visto que nuestra base de datos tendrá que almacenar muchos datos e información a la vez que hacer cálculos.

## Análisis de resultados

La base de datos deberá de tener suficiente espacio y velocidad para poder manejar todos los datos de manera fluida, también deberá contener un menú inicial para poder acceder a las diferentes opciones de personalización o edición de los datos entre otras cosas.

## Conclusiones

Una vez visto su funcionamiento, llegamos a la conclusión de que es una buena manera para gestionar de manera eficiente los productos del stock sin tener que estar pendiente de cada tipo de producto ya que sería un tipo de gestión demasiado complejo con gran responsabilidad. Hemos llegado a la conclusión de que el modelo de base de datos es efectivo tanto a corto como a largo plazo para la gestión de stock.

## Prototipado del sistema

### 1. Persistencia de los datos:

Es fundamental que los datos sobre el stock y los pedidos se mantengan de manera duradera. Esto significa que, incluso después de que el sistema se apague o reinicie, la información debe estar disponible. Utilizar una base de datos relacional puede ser una buena opción, ya que ofrece mecanismos para asegurar la persistencia.

### 2. Concurrencia en la lectura y escritura de los datos:

En un restaurante, es probable que múltiples usuarios (como el personal de cocina y los meseros) accedan a la base de datos al mismo tiempo para realizar pedidos y actualizar el stock. Es importante que la base de datos maneje adecuadamente la concurrencia para evitar conflictos, como que se vendan más platos de los que hay en stock.

### **3. Independencia del medio de almacenamiento:**

La base de datos debería ser capaz de funcionar independientemente del tipo de almacenamiento que se utilice (por ejemplo, disco duro, SSD, almacenamiento en la nube). Esto permite flexibilidad en la infraestructura y facilita la migración a diferentes sistemas de almacenamiento si es necesario.

### **4. Independencia de las aplicaciones que consumirán los datos:**

Es ideal que la base de datos esté diseñada de tal manera que diferentes aplicaciones (como sistemas de punto de venta, aplicaciones móviles, etc.) puedan acceder a ella sin depender de su implementación interna. Esto se puede lograr mediante el uso de APIs o servicios web que actúen como intermediarios.

### **5. Integridad de los datos (consistencia y coherencia):**

La integridad es crucial para asegurar que los datos sean precisos y fiables. Esto incluye establecer restricciones como claves primarias y foráneas y utilizar transacciones para garantizar que las operaciones se completen correctamente. También es importante implementar validaciones para evitar errores en la entrada de datos.

## **¿Qué significa cada uno de estos criterios en el contexto de vuestro proyecto?**

Los criterios anteriores son esenciales para garantizar que el sistema de gestión funcione de manera eficiente y fiable, se debe asegurar que los datos del inventario y de los pedidos estén siempre al alcance y sean precisos. También que no dependan de ningún hardware específico o ligado a algo en específico y al mismo tiempo mantener siempre su coherencia.

## **¿Por qué son importantes?**

Son importantes porque garantizan el correcto funcionamiento, fiabilidad y eficiencia del sistema

## **En base a estos criterios, ¿qué mejoras podéis llevar a cabo en el diseño o implementación de vuestro sistema de información?**

Las mejoras que podríamos llevar a cabo son particionar las tablas por fecha por ejemplo según los pedidos de cada mes o semana actual.

También podríamos replicar esa base de datos para distribuir la carga de lectura y mejorar la disponibilidad del sistema, permitiendo que todo pueda realizarse en diferentes servidores.

**¿Se os ocurre algún otro aspecto que debéis tener en cuenta o que sea importante para vuestro proyecto?**

-Un aspecto que sería bastante considerable sería el control de desperdicios ya que es un aspecto importante en la gestión de un restaurante una eficiente reducción de productos

- Otro aspecto bastante importante sería un registro de costos de productos para mantenerse siempre actualizado de los costos de los ingredientes y ajustar los precios de los platos si los costos por alguna razón varían. Todo esto para mantener siempre una rentabilidad que se ajuste a nuestros objetivos del local.

-Un aspecto que no sería del todo importante es tener un registro implementado de los pedidos por plataformas de terceros (Uber Eats, Glovo). Para gestionar el numero de pedidos por entrega que se realizan al día y mes en el restaurante.