Sistema de Administración

**Gabriel Quinche, David Martínez, Karen Medina**

**No. de Equipo Trabajo: 8**

# INTRODUCCIÓN

El siguiente documento presenta un proyecto de software, con su propósito, requerimientos y posible alcance.

# DESCRIPCIÓN DEL PROBLEMA A RESOLVER

En los restaurantes generalmente se generan conflictos y complicaciones al tener que transmitir “voz a voz” la información de las ventas y el inventario diario restante en la cocina. Se pretende lograr que todos los responsables de manejar esa información tengan acceso a una base de datos actualizada en tiempo real a tener en cuenta a la hora de despachar los pedidos.

# USUARIOS DEL PRODUCTO DE SOFTWARE

Existirían dos tipos de usuarios clasificados de acuerdo a su interacción con el sistema: un administrador, con permisos para modificar cantidades de todos los elementos del sistema, y un usuario tipo “mesero” con accesibilidad restringida para modificar los valores del inventario mediante la generación de pedidos. Eventualmente podría incluirse un usuario tipo “cocina” que reciba pedidos y genere su cobro una vez despachado.

# REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

## Llenar inventario

## Descripción: El usuario administrador guardará los productos que se proyectan disponibles para el día.

*Acciones iniciadoras y comportamiento esperado*:

Se crean listas con los ingredientes disponibles que sólo el usuario administrador tiene la capacidad de modificar. Se espera ser este el paso inicial para las siguientes funcionalidades.

*Requerimientos funcionales*:

Por cada tipo de ingrediente existirá una lista. A esta lista se le agregan los ingredientes disponibles de su mismo tipo.

## Consultar disponibilidad

## Descripción: Los usuarios pueden hacer un recorrido por las listas para saber qué productos hay en el inventario

*Acciones iniciadoras y comportamiento esperado*:

El usuario elige el tipo de producto del que desea saber. El sistema debe responder con la cantidad de productos en lista.

*Requerimientos funcionales*:

Cada lista cuenta con un contador que indica la cantidad de productos, este se modifica cada vez que se agregan productos o se recibe un pedido. Al hacer la consulta se muestr

## Hacer pedido

## Descripción: El usuario “mesero” selecciona una combinación de ingredientes que se encuentren disponibles en las listas. Luego el pedido pasa a una cola junto con los demás.

*Acciones iniciadoras y comportamiento esperado*:

Se eligen una serie de ingredientes que deben estar disponibles en tiempo real. El contador debe restar estos ingredientes del inventario apenas se ordene.

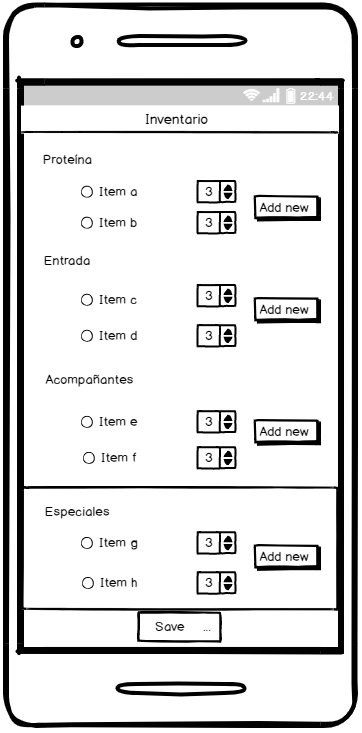
*Requerimientos funcionales*:

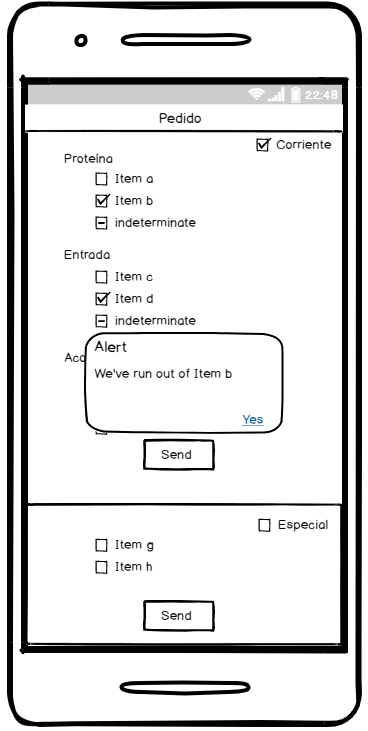
En caso de intentar pedir un ingrediente no disponible no se puede llevar a cabo el pedido. De modo que antes de enviar el pedido el sistema debe hacer una consulta al inventario.

# DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

Dependiendo del tipo de usuario, se presentan con lineamiento *material design* una interfaz minimalista que deja los meseros o distribuidores ingresar una cantidad de productos a vender. Existirá un botón de confirmar que mostrará el valor total de la cuenta.

Para un usuario administrador se le presenta a modo de tablas el conteo de los insumos del local, al igual que la opción de modificar este conteo.

  
Fig 1. Expectativa GUI usuario administrador.

  
Fig 2. Expectativa GUI usuario “mesero”.

# Entornos de desarrollo y de operación

Se espera montar la lógica principal del programa mediante mockup sencillos usando Java y Swing en un repositorio compartido.

# PROTOTIPO DE SOFTWARE INICIAL

Se encuentra alojado en el github del proyecto, en el link:

<https://github.com/Damamapi/sAres>

Adicionalmente, para mantener una versión gráfica de desarrollo del repositorio, se podrá apoyar mediante una herramienta como Sourcetree, disponible en el siguiente URL:

<https://www.sourcetreeapp.com/>

Segundo, en este prototipo se deben implementar por lo menos dos instancias de la estructura de datos tipo **Lista implementada como arreglos**. Para aclarar, aquí Lista se refiere a una estructura de datos lineal, o contenedor de datos en la cual los datos se organizan de una manera lineal (como en una hilera) de tal manera que cada dato puede tener a lo sumo un antecesor y a lo sumo un sucesor.

Tercero, para estas listas implementadas como arreglos se deben soportar por lo menos las siguientes operaciones funcionales:

* Creación de la lista
* Inserción de un solo dato
* Actualización de un solo dato
* Eliminación de un solo datos
* Búsqueda de un solo dato
* Consulta de todos los datos
* Almacenamiento de los datos

# PRUEBAS DEL PROTOTIPO

* Si bien la actualización y consulta en una base de datos desordenada podría ser eficiente con el uso de hash maps y tree maps para conformarse con los requisitos, las listas con arrays sencillos terminan siendo prácticas pues para un restaurante común una lista insumos no pasará de los 1000, donde linealmente se puede recorrer con facilidad, pero que por su condición bastante constante (pocos restaurantes suelen agregar nuevos ingredientes y platos incluso en el transcurso de un mes) el corrimiento de incluso los n elementos para acomodar ingredientes de manera ordenada es un costo insignificante para una computadora o incluso un móvil de condiciones actuales. de igual forma los pedidos que suelen ser actualizados constantemente no pasarán de los 100 en un hora, por lo que una lista con un contador circular, podrá simular sin problemas una cola, para dar prioridad a los pedidos segun su llegada, pero igual poder acceder a datos específicos directamente.