

Sprint #1

# **Merkit**

Taller de ingeniería de software  
21/10/2020

Tamara Badilla  
Sebastián Pacheco  
Álvaro Valverde

*“Merkit es una página web de administración para minimarket gratis, fácil y rápida de usar. Si bien la página no posee sistema de boleta, permite al usuario manejar fácilmente el stock, las ventas diarias realizadas, y acceso a la nube para que sus datos sean accesibles mediante cualquier dispositivo donde inicie sesión.”*

Dentro del primer Sprint del sistema Merkit, se realizó la obtención de requisitos junto con la arquitectura del sistema, por lo que en este documento se encuentran los requisitos funcionales, no funcionales y la arquitectura que se utilizará.

## *Requerimientos*

### Funcionales

Dentro de los requisitos funcionales de la aplicación, se encuentran:

- *Login*: La aplicación web cuenta con un inicio de sesión vinculado al correo, a través
- *Sistema de inventario*: El sistema debe poseer inventario, para así obtener una lista clara de productos los disponibles con su precio y stock correspondiente. Esto integrará las siguientes funciones:
  - ◆ Acceso a la base de datos.
  - ◆ Listar todos los productos en la base de datos junto con su nombre, stock (opcional), etc.
  - ◆ Agregar/quitar y editar productos de la base de datos.
- *Resumen de Ventas*: La página web permite mostrar el total de ventas, para hacer un análisis rápido, ofreciendo al usuario acceso a este cuando lo necesite. Esto integrará:
  - ◆ Acceso al historial de ventas realizadas.

- ◆ Acceso al almacenamiento local.
  - ◆ Filtro de productos según lo que se requiera (orden alfabético, mayor a, etc.)
- *Sistema de calculadora de vuelto*: La página debe entregar el valor del vuelto en monedas/billetes según lo requiera para facilitarle al cajero(a) este proceso.
- *Sistema de ventas*: el sistema debe mostrar los elementos que se puedan vender, junto con la capacidad de añadirlos a una lista para agregarlos a una venta. Esto integrará:
- ◆ Barra de búsqueda para acelerar el proceso.
  - ◆ Botón de finalizar venta que modifica la BD ventas y BD inventario.
  - ◆ Vista de elementos a agregar y elementos agregados.
- *Configuración de usuario*: el sistema debe permitir al usuario modificar sus datos, tales como su foto de perfil, su correo, su contraseña, etc.

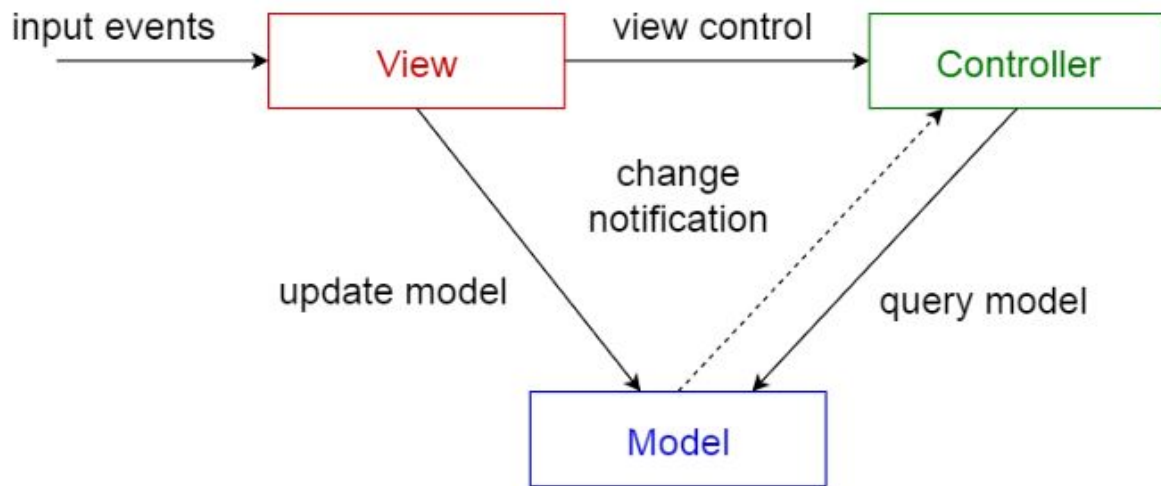
## No Funcionales

Por el contrario, los requisitos no funcionales que se detectaron para construir el sistema son:

- *Usabilidad*: Si bien es esperado que la página web esté disponible para usuarios de todo tipo, el cliente específico que se tiene en mente al construir el sistema es alguien que no posee mucha experiencia con la tecnología, es por esto que la usabilidad es muy importante. Se piensa llevar esto a cabo mediante un seguimiento cercano a los usuarios.
- *Backup*: El sistema se conectará a la nube de Google para así tener la posibilidad de acceder a sus datos desde cualquier dispositivo, en cualquier lugar (con conexión a internet).

## Arquitectura del sistema

Se decidió construir el sistema usando el tipo de arquitectura de Modelo-Vista-Controlador, ya que por el requisito funcional de Usabilidad, se espera separar lo más posible el lado de interacción de usuario de toda la logística que hay detrás.



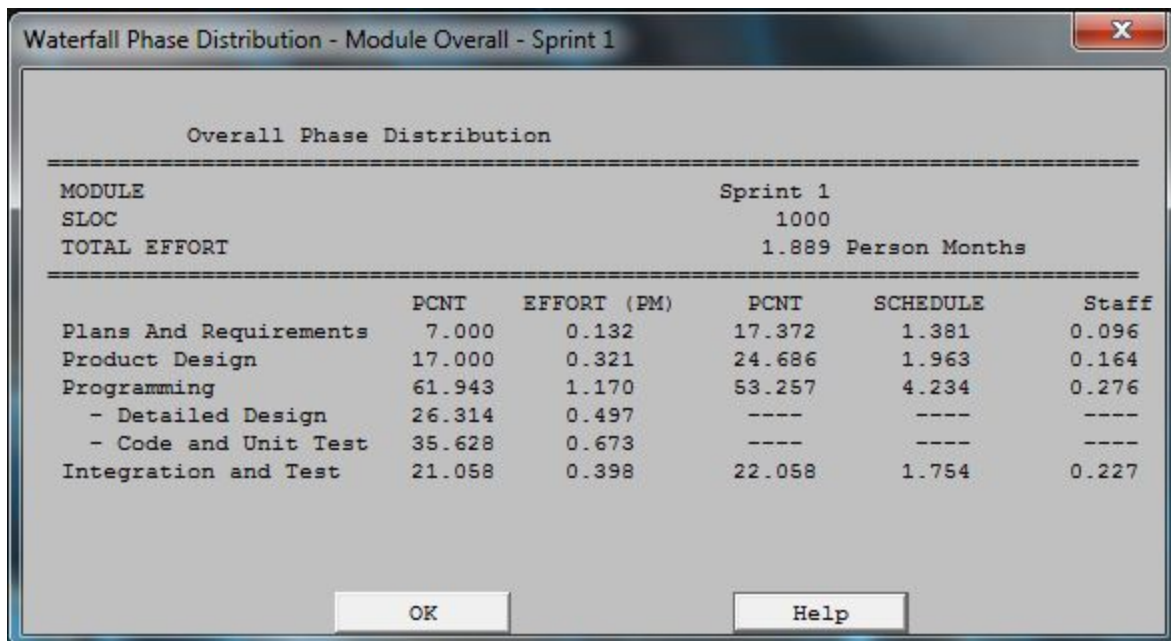
(Diagrama MVC, *Patrones comunes de arquitectura*)

→ **Controller:** Corresponde a un componente centralizado que maneja la información de la aplicación. En particular, será el encargado de administrar los objetos tipo Producto que han sido creados con la información recibida del Modelo:

- ◆ **Clase Producto:** Contendrá la información “cruda” de los productos, tal cual llega desde el Modelo, pero encapsulada. También contará con varias funciones disponibles que facilitarán su despliegue en la Vista de la aplicación.

- **View:** Corresponde al componente de interfaz gráfica, que dispondrá de todas las funcionalidades del software, clasificado en pestañas. Cada pestaña tendrá sus propios botones y barras de búsqueda correspondientes, por lo que el Controller estará escuchando en todas las pestañas.
- **Model:** Corresponderá al componente que administrará la conexión con la Base de Datos, y entregará la información requerida al Controller. Tendrá una copia local del inventario de la empresa, además de un respaldo en la nube (Google Drive), y tendrá la capacidad de requerir los datos a conveniencia.

## Planificación



Waterfall Phase Distribution - Module Overall - Sprint 1

Overall Phase Distribution

MODULE	Sprint 1				
SLOC	1000				
TOTAL EFFORT	1.889 Person Months				
	PCNT	EFFORT (PM)	PCNT	SCHEDULE	Staff
Plans And Requirements	7.000	0.132	17.372	1.381	0.096
Product Design	17.000	0.321	24.686	1.963	0.164
Programming	61.943	1.170	53.257	4.234	0.276
- Detailed Design	26.314	0.497	----	----	----
- Code and Unit Test	35.628	0.673	----	----	----
Integration and Test	21.058	0.398	22.058	1.754	0.227

OK Help

