

# Aseguramiento de Calidad del Software



Docente: Silvia Alejandra Andrade Rodríguez  
Equipo: QSC (Quality Software and Consulting)  
Proyecto Line Clothes

# 1. Descripción del problema

- Realizar un sistema que pueda gestionar los productos, ventas, cambios, devoluciones y envíos de una tienda de ropa.

# LineClothes



Editor de Defectos

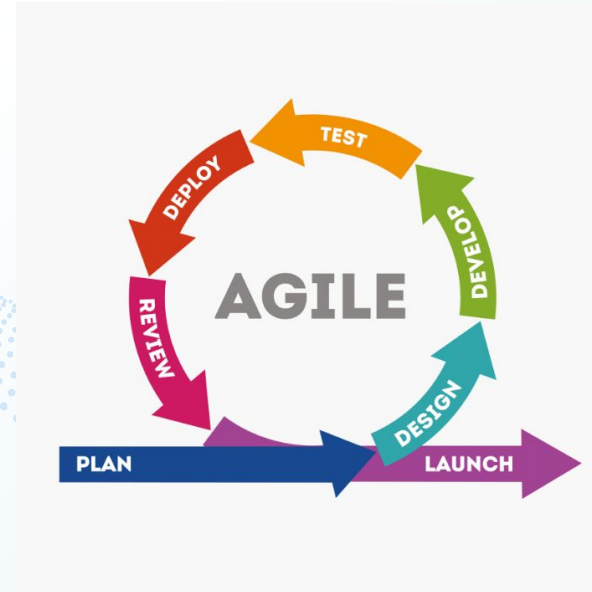
Proyecto	ID	Tipo	Inyectado	Elimina
/Proyecto/Line Clothes PSP0	1	Syntax	Code	Code
/Proyecto/Line Clothes PSP0	2	Documentation	Planning	Plannir

## 2. Modelos y Estándares de Calidad

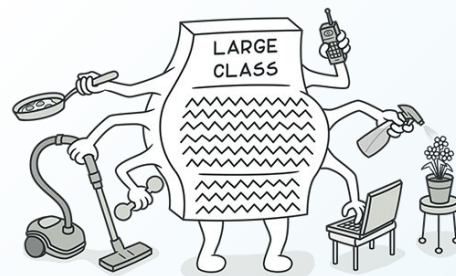
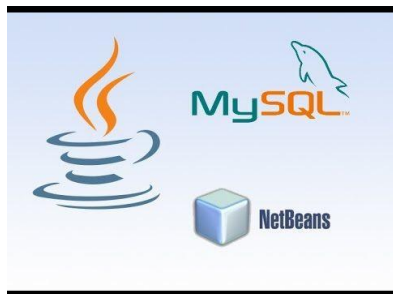
### Modelo en el Proceso



### Modelo en el Producto



# Estándares de Calidad



# 3. Calidad en el Proceso

PSP

Personal Software Process

Proceso de Mejora, de forma disciplinada.

Encontrar los defectos en fases tempranas del Proceso.

Medir su trabajo, analizar sus resultados.

# Productos Obtenidos

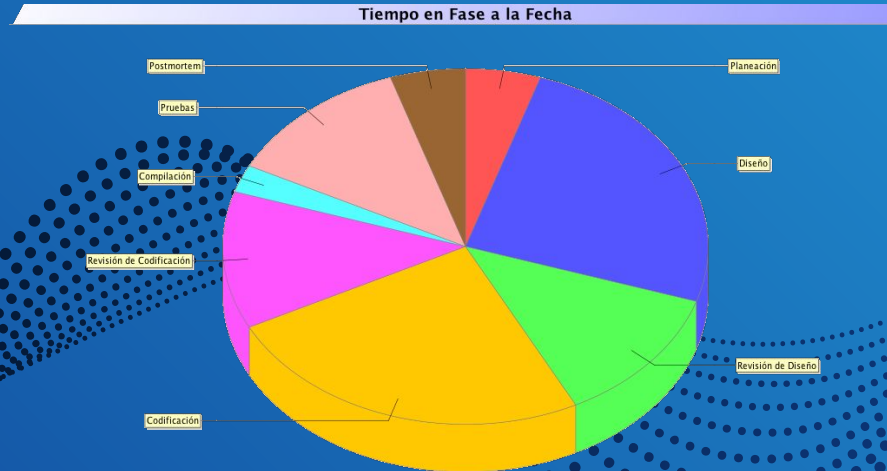
## Análisis de Defectos





# Productos Obtenidos

## Análisis de Planeación



# Herramientas Utilizadas



**Process  
Dashboard**



**balsamiq®**



**StarUML**

The Open Source UML/MDA Platform



# 4. Calidad en el producto

## Modelos de calidad de Productos

Boehm (1976) **McCall (1977)** Arthur (1985) **FURPS (187)** Gilb (1988)

Deutch (1988) **Schulmeyer (1990)** Gillies (1992) **Drome (1995)** ISO-9126 (2001)

A decorative graphic at the bottom of the slide consisting of a series of overlapping, wavy lines of small dots in white and light blue, creating a sense of motion and depth.

## Modelo de McCall

- Revisión del producto (capacidad de cambio)
- Transición del producto (capacidad de adaptación)
- Operaciones del producto (características básicas del producto).

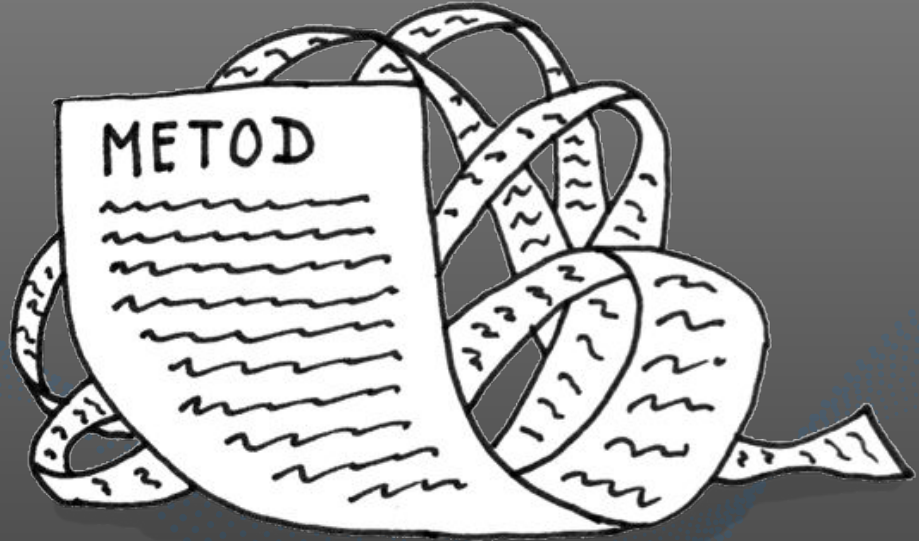
## Modelo de Boehm

- Realice lo que desea el usuario
- Utilice recursos informáticos de manera correcta y eficiente
- Sea fácil de utilizar y aprender
- Sea bien diseñado, codificado, probado y mantenido.

## ISO IEC 9126

- Define seis factores de calidad, que se consideran tanto internos como externos (funcionalidad, fiabilidad, usabilidad, eficiencia, facilidad de mantenimiento y portabilidad)

## 5. Calidad del código



## EXCELENCIA:

Un código que más allá de programar las funcionalidades requeridas, esté libre de errores, sea claro, simple y orientado a objetos.



## CALIDAD:

Conjunto de propiedades que le confieren la capacidad para satisfacer las necesidades implícitas o explícitas para las que se ha desarrollado.

“Desarrollar el código que permita proveer al cliente de un producto que satisfaga sus necesidades”

## Código duplicado:

- Este término se utiliza cuando hablamos de un código fuente que aparece más de una vez, ya sea dentro de uno o diferentes programas, de propiedad o mantenido, por la misma entidad.

## Código muerto

- Es el código que se encuentra en nuestra aplicación, pero nunca es utilizado. Normalmente aparece después de hacer refactor en nuestro código.

## Estándares de codificación

- Se refiere a convenciones para escribir código fuente, las cuales frecuentemente son dependientes del lenguaje de programación.

## Bugs

- Un bug es un error o un defecto en el software que hace que un programa funcione de forma incorrecta.

## Complejidad ciclomática

- Es una métrica de calidad software basada en el cálculo del número de caminos independientes que tiene nuestro código.



```
public class cls_factura {  
  
    private final String select = "SELECT * FROM TICKET";  
    private PreparedStatement PS;  
    private DefaultTableModel DT,DT1;  
    private ResultSet RS;  
    private final cls_conexion CN;  
  
    public cls_factura(){  
        PS=null;  
        CN= new cls_conexion();  
  
    }  
}
```

```
public DefaultTableModel getDatos(){  
    try{  
        setTitulos();  
        PS=CN.getConnection().prepareStatement(select);  
        RS=PS.executeQuery();  
        Object[] fila = new Object[8];  
        while(RS.next()){  
            fila[0]=RS.getInt(1);  
            fila[1]=RS.getString(2);  
            fila[2]=RS.getString(3);  
            fila[3]=RS.getString(4);  
        }  
    }  
}
```

## 6. Metodologías Ágiles

¿Que es una metodología ágil?

- Por definición, las metodologías ágiles son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno.



# ¿Por qué se usan las metodologías ágiles?

- **Mejora de la calidad del producto:** Estas metodologías fomentan el enfoque proactivo de los miembros del equipo en la búsqueda de la excelencia del producto. Además, la integración, comprobación y mejora continúa de las propiedades del producto mejora considerablemente el resultado final.
- **Mayor satisfacción del cliente:** El cliente está más satisfecho al verse involucrado y comprometido a lo largo de todo el proceso de desarrollo. Mediante varias demostraciones y entregas, el cliente vive a tiempo real las mejoras introducidas en el proceso
- **Mayor motivación de los trabajadores:** Los equipos de trabajo autogestionados, facilitan el desarrollo de la capacidad creativa y de innovación entre sus miembros.

- **Trabajo colaborativo:** La división del trabajo por distintos equipos y roles junto al desarrollo de reuniones frecuentes, permite una mejor organización del trabajo.
- **Uso de métricas más relevantes:** Las métricas utilizadas para estimar parámetros como tiempo, coste, rendimiento, etc. son normalmente más reales en proyectos ágiles que en los tradicionales. Gracias a la división en pequeños equipos y fases podemos ser más conscientes de lo que está sucediendo.
- **Mayor control y capacidad de predicción:** La oportunidad de revisar y adaptar el producto a lo largo del proceso ágil, permite a todos los miembros del proyecto ejercer un mayor control sobre su trabajo, cosa que permite mejorar la capacidad de predicción en tiempo y costes.

# TABLEROS KANBAN Y TRELLO







## Los tableros de su equipo

**Desarrollo App  
LineClothes**

**Modificaciion de la  
Factura**

Crear un tablero nuevo  
5 restantes



## Tableros del equipo

**Administrar los Envios**

**Devoluciones y  
Cambios**

**Inventario**



## Información

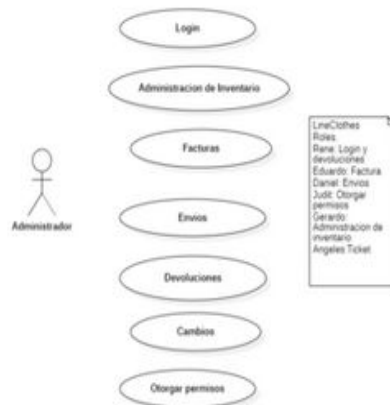
# LineClothes

Información general del proyecto



+ Añada otra tarjeta

## Punto de partida



Modelo básico de casos de uso del punto de partida.



+ Añada otra tarjeta

## Herramientas



Java 8 SE



# NetBeans

NetBeans IDE 8.2



+ Añada otra tarjeta

## Desarrollo



Desarrollo.



Generar Devoluciones

🕒 11 de dic. de 2020

RM

+ Añada otra tarjeta