



Resumen uso de Entornos de Desarrollo en el Proyecto

Olga Sotero Escobar

1ºDAM A

Herramientas usadas en el ciclo de desarrollo

Durante el desarrollo del proyecto utilicé varias herramientas clave. El IDE principal fue Visual Studio Code, donde pude programar en Java, gestionar carpetas y trabajar con extensiones como Copilot, que me ayudó a completar código y a recibir sugerencias inteligentes.

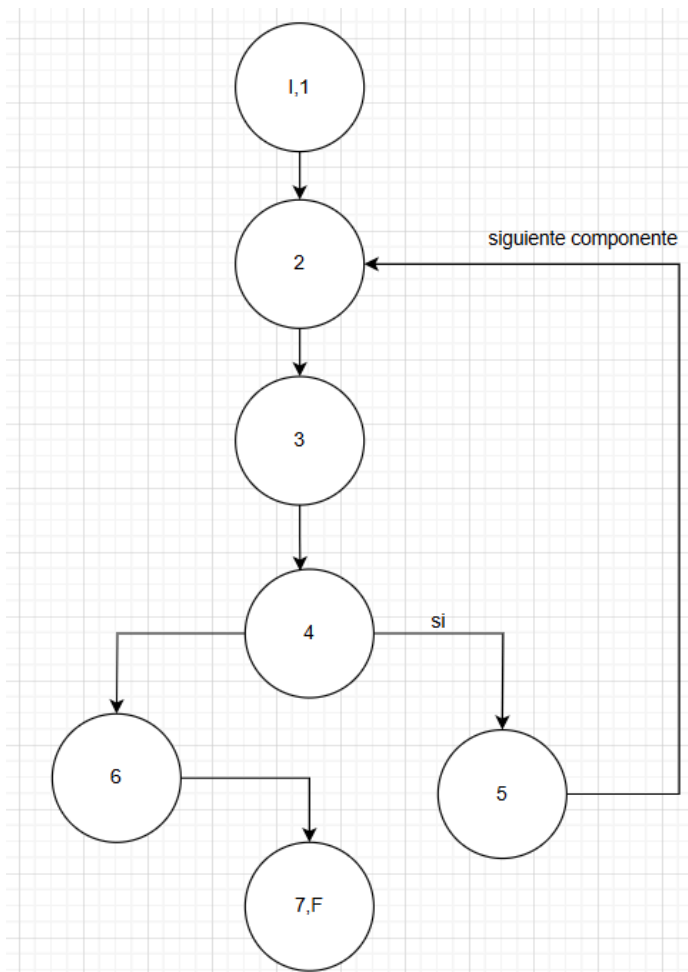
Especificación de requisitos

| Tipo | Descripción |
|------------------------|--|
| <u>Funcionalidades</u> | <ul style="list-style-type: none">- Gestión de componentes- Registro de usuarios- Registro de compras- Actualización de stock en la base de datos |
| <u>Restricciones</u> | <ul style="list-style-type: none">- El stock no puede ser negativo- Los datos deben guardarse correctamente en MySQL- Solo usuarios válidos pueden acceder |
| <u>Prioridades</u> | <ul style="list-style-type: none">- Conexión segura a la base de datos- Correcta actualización del stock tras compra- Interfaz gráfica clara y sencilla |

Grafo de flujo

- En este apartado voy a realizar el grafo de flujo del método calcularPrecioTotal();

```
public double calcularPrecioTotal(Connection cn, String[] componentes) {    1
double total = 0.0;
try {
    for (String nombre : componentes) {    2
        3 Statement st = cn.createStatement();
        3 String consulta = "SELECT precio FROM componentes WHERE nombre = '" + nombre + "'";
        ResultSet rs = st.executeQuery(consulta);    3
        4 if (rs.next()) {
            5 total += rs.getDouble(columnLabel:"precio");
        }
    }
    6 } catch (SQLException e) {
        System.out.println(x:"Error al calcular precio total");
        e.printStackTrace();
    }
    7 return total;
}
```



- Complejidad ciclomática:

$$V(G) = \text{aristas} - \text{nodos} + 2 = 9 - 7 + 2 = 4$$

$$V(G) = \text{nodos predicados} + 1 = 3 + 1 = 4$$

$$V(G) = \text{regiones} = 4$$

- Caminos:

1C: 1,2,3,4,6,7,F

Ejemplo: no encuentra resultados (rs.next() falso).

2C: 1,2,3,4,5,siguiente componente, 2...

Ejemplo: encuentra resultado (rs.next() verdadero) y suma precio.

3C: 1,8,7,F

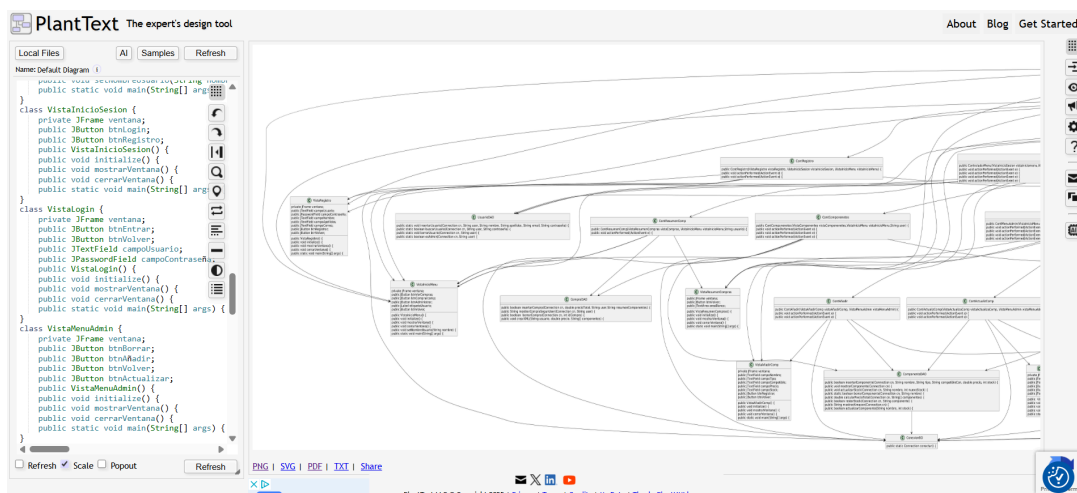
Ejemplo: ocurre excepción, salta al bloque catch.

4C: 1,2,3,4,5,7,F

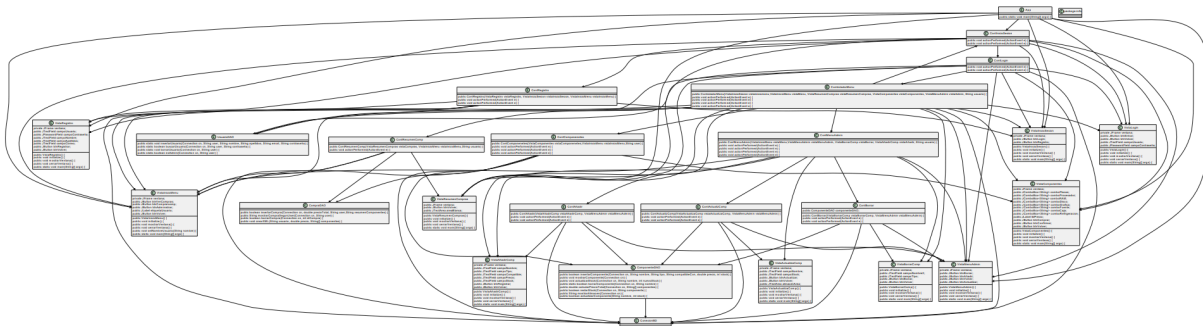
Ejemplo: termina después del último componente.

UML

- Para realizar el UML he usado esta pagina (<https://www.planttext.com/>), en la cual solo hay que subir un archivo puml en el que se describa las clases y relaciones del proyecto. He subido el archivo el cual se encuentra en la carpeta (PROYECTO_COMPLETO_RELACIONES.puml), y al pulsar refresh se crea el UML:



- Así queda el UML que también se encuentra en la carpeta(UML_PROYECTO.pdf):



Uso de git/github

- Durante el desarrollo, utilicé Git para gestionar versiones del código. Cada cambio importante fue guardado en commits, permitiendo volver atrás si surgían errores. Además, usé GitHub para subir el proyecto online, mantenerlo seguro y compartirlo fácilmente.
- Este fue el primer commit que hice al principio del proyecto, estos son los pasos que hacía en git cuando quería subir cambios a github:

```
User@DESKTOP-CABPFFI MINGW64 ~
$ cd PROGRAMACION/

User@DESKTOP-CABPFFI MINGW64 ~/PROGRAMACION
$ cd PROYECTO/

User@DESKTOP-CABPFFI MINGW64 ~/PROGRAMACION/PROYECTO (master)
$ git remote remove origin

User@DESKTOP-CABPFFI MINGW64 ~/PROGRAMACION/PROYECTO (master)
$ rm -rf .git
$ git init
Initialized empty Git repository in C:/Users/User/PROGRAMACION/PROYECTO/.git/

User@DESKTOP-CABPFFI MINGW64 ~/PROGRAMACION/PROYECTO (master)
$ git config --global user.name "olsenabi32"
$ git config --global user.email "olgasteroescobar@gmail.com"

User@DESKTOP-CABPFFI MINGW64 ~/PROGRAMACION/PROYECTO (master)
$ git remote add origin https://github.com/olsenabi32/PROYECTO.git

User@DESKTOP-CABPFFI MINGW64 ~/PROGRAMACION/PROYECTO (master)
$ git add .

User@DESKTOP-CABPFFI MINGW64 ~/PROGRAMACION/PROYECTO (master)
$ git commit -m "Primer commit subida de carpetas"
[master (root-commit) 5d0c6e7] Primer commit subida de carpetas
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Controlador/Controlador.java
create mode 100644 Modelo/Modelo.java
create mode 100644 Vista/VistaPrimera.java

User@DESKTOP-CABPFFI MINGW64 ~/PROGRAMACION/PROYECTO (master)
$ git push -u origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 449 bytes | 449.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/olsenabi32/PROYECTO.git
* [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

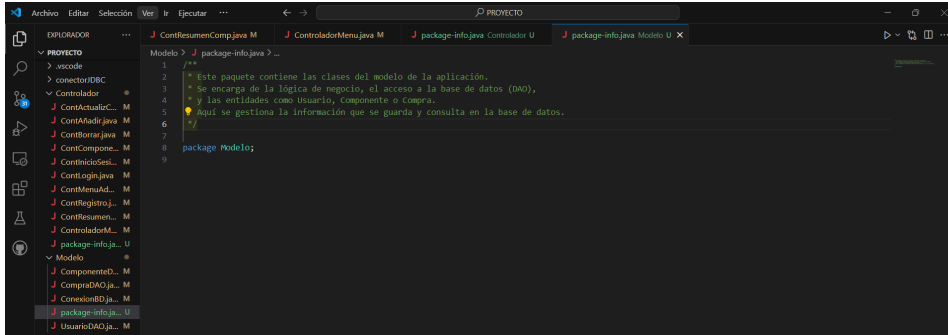
User@DESKTOP-CABPFFI MINGW64 ~/PROGRAMACION/PROYECTO (master)
$
```

- A través de este enlace puede visitar mi repositorio de github donde puede ver todo el proyecto subido correctamente:

[ENLACE](#)

JAVADOC

- Para crear el javadoc, he añadido los comentarios a todos los archivos y luego he creado un archivo muy similar a éste en cada carpeta:



- En los comentarios he usado @param: para explicar los parámetros de cada método. @return: para detallar qué devuelve cada método.

```
/**
 * Método restarStock.
 * @return resultado del método o acción realizada.
 */
```

- Tuve que meter todo el código en una carpeta en src y después de hacer los comentarios en el código poner esta línea en la terminal de visual studio y se creo directamente:

```
PS C:\Users\User\PROGRAMACION\PROYECTO> & "C:\Program Files\Java\jdk-24\bin\javadoc.exe" -d doc -sourcepath src -subpackages Controlador:Modelo:Vista
Loading source files for package Controlador...
Loading source files for package Modelo...
Loading source files for package Vista...
Constructing Javadoc information...
Creating destination directory: "doc\"
Building index for all the packages and classes...
Standard Doclet version 24+36-3646
Building tree for all the packages and classes...
Generating doc\Modelo\ComponentedDAO.html...
src\Modelo\ComponentedDAO.java:19: warning: no @param for cn
    public boolean isInsertarComponente(Connection cn, String nombre, String tipo, String compatibleCon, double precio, int stock) {

```

- Puede consultar el Javadoc en la página web en el apartado de lenguajes de marcas.

OVERVIEW
TREE
INDEX
SEARCH
HELP

Packages

| Package | Description |
|-------------|--|
| Controlador | Este paquete contiene todos los controladores del patrón MVC. |
| Modelo | Este paquete contiene las clases del modelo de la aplicación. |
| Vista | Este paquete contiene todas las clases relacionadas con la interfaz gráfica. |

PÁGINA WEB

- Para esta página web he creado dentro de la carpeta html 4 archivos html dentro de estos está la parte de css y javascript.
- Desde la parte de css con :hover hago que se cambie de color ciertas partes de la pagina.

```
main h1:hover {  
  color: #e532ec;  
}
```

- Cuando se pulsa un botón desde la parte del código de javascript hago que lo escuche y envíe un mensaje por pantalla.

```
<script>  
document.getElementById('botonJavadoc').addEventListener('click', function() {  
  alert('¡Gracias por consultar mi JavaDoc!');  
});  
document.getElementById('botonXML').addEventListener('click', function() {  
  alert('¡Gracias por revisar mi XML!');  
});  
</script>
```

- También he añadido una animación para que los títulos se muevan un poco al pulsarlos, cuando se pulsan los títulos desde el script se llama al método que está definido en la parte de style y hace que se muevan.

```
document.querySelectorAll('h1, h2').forEach(t => {  
  t.addEventListener('click', function() {  
    t.classList.add('animar');  
    setTimeout(() => { t.classList.remove('animar'); }, 300);  
  });  
});  
</script>
```

```
.animar {  
  transform: translate(10px, -10px);  
  transition: transform 0.3s ease;  
}  
</style>
```