

9 de enero de  
2025

# INGENIERÍA DEL SOFTWARE II

**INGENIERÍA INFORMÁTICA**  
**JORGE LÓPEZ GÓMEZ**  
**MARCO MUÑOZ GARCÍA**  
**ISMAEL LÓPEZ MARÍN**  
**PABLO VERDÚGUEZ GERVASO**

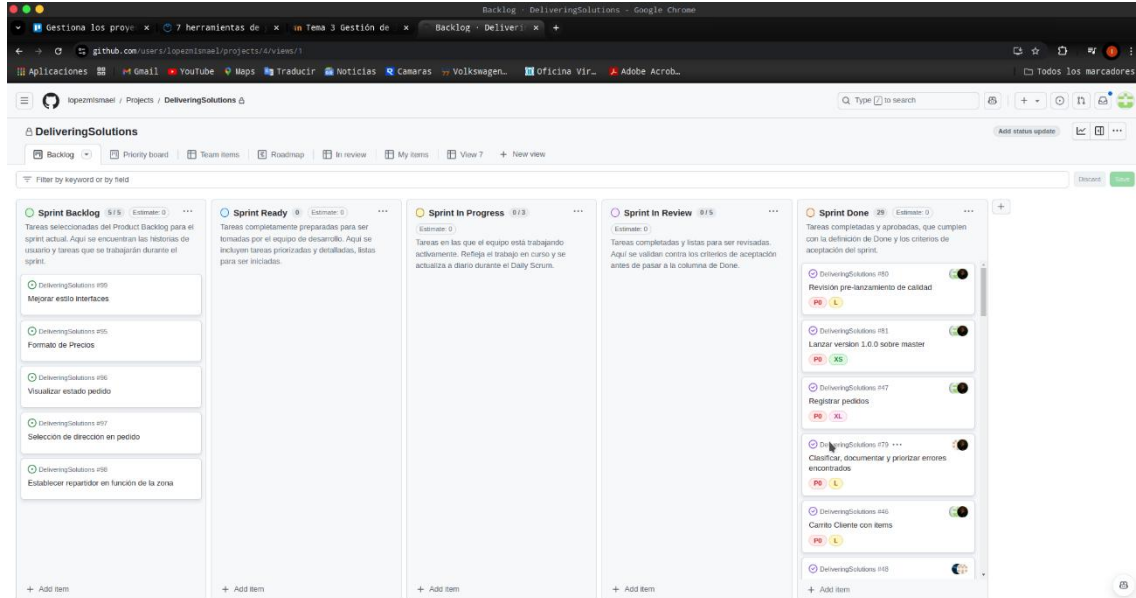
# ÍNDICE

DOCUMENTACIÓN SPRINT 5 .....	2
5.1. FORMATO DE PRECIOS .....	3
Evaluación del Cambio .....	3
Revisión y Cierre.....	4
5.1. VISUALIZAR ESTADO DEL PEDIDO .....	4
Evaluación del Cambio .....	5
Revisión y Cierre.....	5
5.2. SELECCIÓN DE DIRECCIÓN EN PEDIDO .....	6
Evaluación del Cambio .....	6
Revisión y Cierre.....	6
5.3. ESTABLECER REPARTIDOR EN FUNCIÓN DE ZONA.....	7
Evaluación del Cambio .....	7
Revisión y Cierre.....	7
5.4. MEJORAR ESTILO DE INTERFACES .....	8
Evaluación del Cambio .....	8
Revisión y Cierre.....	8
DOCUMENTACIÓN SPRINT 6 .....	9
6.1 Implementar SonarCloud .....	9
Evaluación del Cambio .....	9
6.2 Mejorar Estilo de Interfaces .....	10
6.3 Migración del Sistema a la Nube de AWS .....	11
Validaciones Finales.....	12
DOCUMENTACIÓN SPRINT 7 .....	13
7.1 Solucionar Issues SonarQube.....	13
7.2 Generar Test Iniciales para Todos los Gestores.....	14
Validaciones Finales.....	16
DOCUMENTACIÓN SPRINT 8 .....	17
8.1 Solucionar Issues SonarQube.....	17
Evaluación del Cambio .....	17
Revisión y Cierre.....	18
8.2 Generar Tests Finales para Todos los Gestores .....	19
Evaluación del Cambio .....	19
Revisión y Cierre.....	20
Validaciones Finales.....	21

# DOCUMENTACIÓN SPRINT 5

**Fecha:** 13 de noviembre de 2024

Durante este sprint nos hemos encargado de realizar los cambios solicitados por el cliente, así como mejoras del sistema que hemos considerado necesarias.



*Ilustración 1. Inicio del Sprint 5.*

La auditoría con el cliente se realizó el día 6 de noviembre. Se recibieron las siguientes solicitudes de cambio (CR):

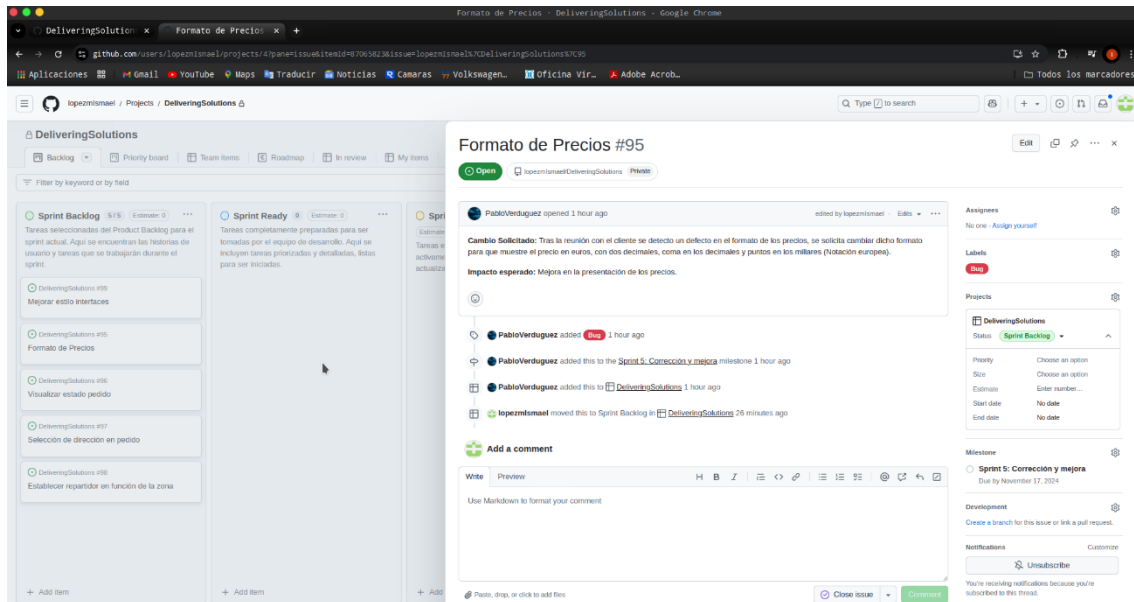
1. Formato de los precios con dos decimales y notación europea (puntos en los millares, comas en los decimales).
2. Selección de distintas direcciones del usuario a la hora de hacer el pedido, o registrar una nueva.
3. Visualizar el estado del pedido en el rol de cliente.
4. El repartidor debe asignarse en función de las zonas que tenga establecidas.

Adicionalmente, para mejorar la experiencia del usuario, se propuso el siguiente cambio:

5. Establecer un estilo más amigable y llamativo para las interfaces.

## 5.1. FORMATO DE PRECIOS

Esta tarea la hemos clasificado como de baja prioridad y de tamaño medio. Este cambio implica modificaciones en la presentación de datos y ajustes en la interfaz de usuario. La experiencia del usuario se incrementó significativamente, así como la satisfacción del cliente final.



*Ilustración 2. Captura de la tarea “Formato de Precios”.*

## Evaluación del Cambio

### Riesgos:

- Posibles inconsistencias en la presentación y manipulación de datos.
- Necesidad de pruebas para asegurar que no se almacenen datos incorrectos en la base de datos ni se muestren valores erróneos al cliente.

### Aprobación del Cambio:

- **Viable:** Sí.
- **Impacto:** Medio.
- **Prioridad:** Baja.
- **CCB determina:** Válido.

### Implementación

- Se corrigieron todas las interfaces para mostrar los precios en formato europeo.
- Se movió el archivo pom.xml de vuelta a la raíz del proyecto.
- Los precios fueron alineados a la derecha en todos los tickets, mejorando su visualización.

### Validación de los Requisitos Implementados

- Pruebas realizadas para garantizar que los precios se muestren correctamente en formato europeo.
  - Visualización de precios en tickets.
  - Pruebas con valores atípicos (muy altos, bajos, o con decimales).

**Resultado:** Los precios se muestran correctamente en todos los casos probados.

## Revisión y Cierre

### Validación de la Integridad de los Datos

- Confirmación de que las modificaciones no afecten la manipulación o el almacenamiento de datos.
  - **Base de datos:** Datos almacenados correctamente según el formato europeo.
  - **Operaciones internas:** Cálculos y sumas sin valores erróneos.

**Resultado:** Datos consistentes y sin errores.

### Revisión de Código

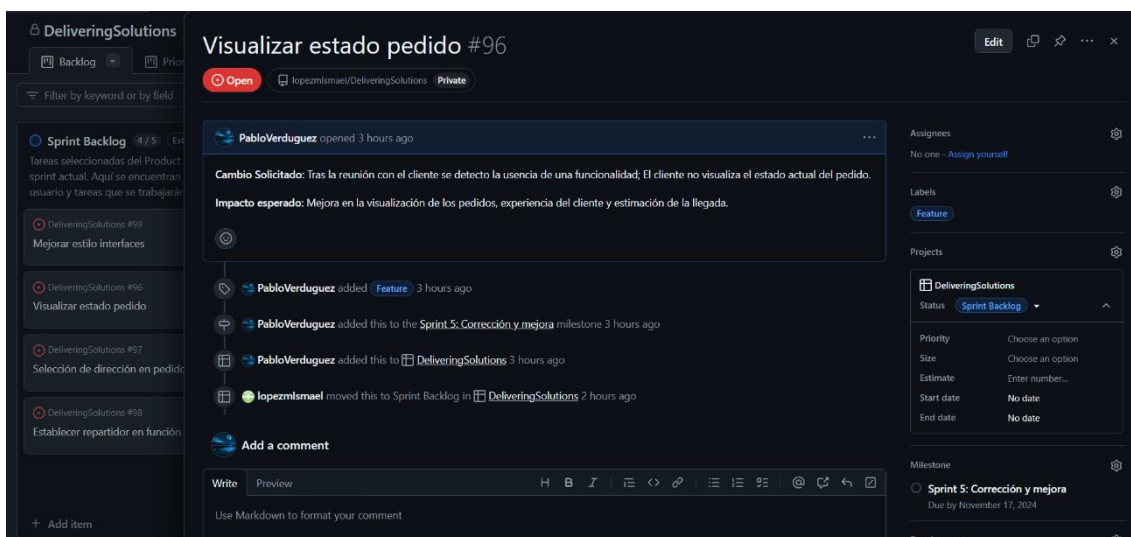
- Auditoría del código realizada para asegurar que el cambio esté aislado y siga las mejores prácticas.

**Resultado:** Código revisado, aprobado y puesto en producción.

## 5.1. VISUALIZAR ESTADO DEL PEDIDO

**Cambio Solicitado:** Tras la reunión con el cliente se detectó la usencia de una funcionalidad; El cliente no visualiza el estado actual del pedido.

**Impacto esperado:** Mejora en la visualización de los pedidos, experiencia del cliente y estimación de la llegada.



*Ilustración 3. Captura de la tarea “Visualizar estado Pedido”.*

## Evaluación del Cambio

### Análisis de Impacto:

- Ajustes necesarios en la presentación de datos y en la interfaz de usuario.
- Información del estado del pedido debe actualizarse en tiempo real o en intervalos adecuados.

### Riesgos:

- Posibles inconsistencias en los datos mostrados al cliente.
- Problemas de diseño o bugs en la interfaz que afecten la experiencia del usuario.

### Aprobación del Cambio:

- **Viable:** Sí.
- **Impacto:** Alto.
- **Prioridad:** Media.
- **CCB determina:** Válido.

### Implementación

- Se agregó un botón en el menú inicial del cliente para visualizar los pedidos realizados.
- Se incluyó la posibilidad de ver detalles del pedido y actualizaciones en tiempo real.

## Revisión y Cierre

### Validación de los Requisitos Implementados

- Pruebas realizadas para garantizar que el estado de los pedidos se visualice correctamente.
  - Acceso al botón en el menú inicial del cliente.
  - Verificación de detalles del pedido y actualizaciones consistentes.

**Resultado:** Los estados de los pedidos se visualizan correctamente, con actualizaciones consistentes.

### Validación de la Integridad de los Datos

- Confirmación de que los estados mostrados sean consistentes con los datos de la base de datos.

**Resultado:** Datos precisos y consistentes.

### Revisión de Código

- Auditoría del código realizada para garantizar que los cambios no afecten otras funcionalidades.

**Resultado:** Código revisado y aprobado.

## 5.2. SELECCIÓN DE DIRECCIÓN EN PEDIDO

**Cambio Solicitado:** Tras la reunión con el cliente se detectó que el cliente (ROL) solo hace pedidos a la primera dirección (Establecida por defecto), se debe corregir para que el cliente pueda seleccionar una de las direcciones registradas o registrar una nueva.

**Impacto esperado:** Mejorar la accesibilidad del uso en la aplicación a la hora de hacer un pedido.

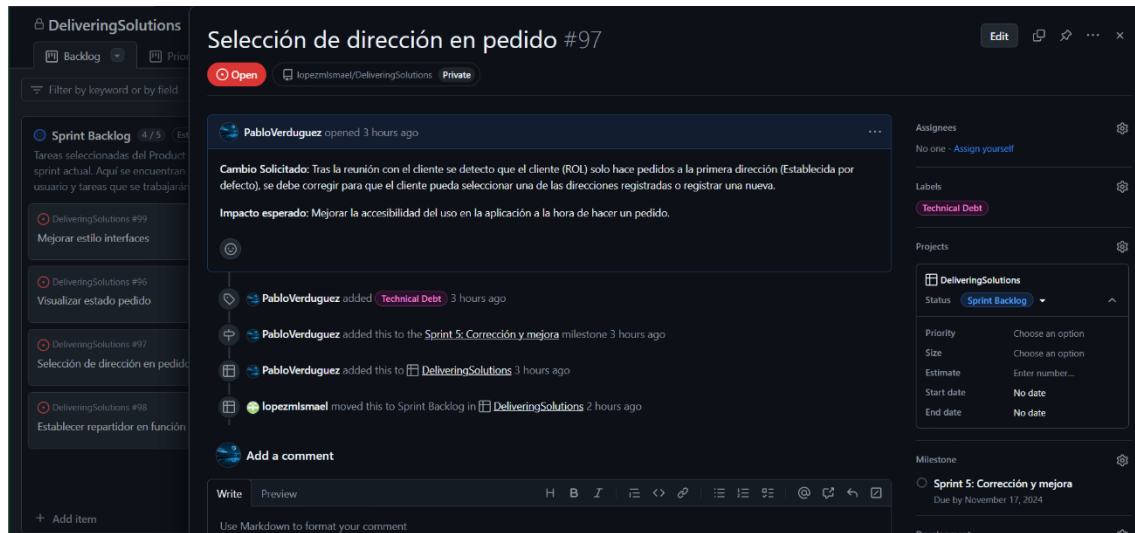


Ilustración 3. Tarea de “Selección de dirección en pedido”.

## Evaluación del Cambio

### Análisis de Impacto:

- Ajustes en la interfaz de usuario y en la lógica de backend para manejar la selección de múltiples direcciones.

### Riesgos:

- Posibles inconsistencias al seleccionar direcciones desactualizadas o inválidas.
- Necesidad de pruebas exhaustivas para asegurar el correcto funcionamiento.

### Aprobación del Cambio:

- **Viable:** Sí.
- **Impacto:** Medio.
- **Prioridad:** Alta.
- **CCB determina:** Válido.

## Revisión y Cierre

### Validación de los Requisitos Implementados

Pruebas realizadas para garantizar que los clientes puedan:

- Seleccionar direcciones registradas.
- Registrar nuevas direcciones durante el pedido.

**Resultado:** Funcionalidad implementada correctamente.

## Validación de la Integridad de los Datos

- Confirmación de que las direcciones seleccionadas o registradas se almacenan correctamente en la base de datos.

**Resultado:** Sin inconsistencias ni errores.

### 5.3. ESTABLECER REPARTIDOR EN FUNCIÓN DE ZONA

**Cambio Solicitado:** El repartidor(rol) actualmente puede elegir cualquier pedido, indistintamente de su ubicación. Puesto que al registrarse establece una zona con una serie de códigos postales, solo debe poder repartir en esas ubicaciones.

Mostrar pedidos a repartidores(rol) que tengan zonas con esos CP.

**Impacto esperado:** Evitar que el repartidor(rol) por error seleccione un pedido muy lejano.



*Ilustración 4. Tarea de “Establecer Repartidor en Función de Zona”*

## Evaluación del Cambio

### Análisis de Impacto:

- Ajustes en la lógica de asignación de pedidos, restringiendo su visualización según las zonas configuradas.

### Riesgos:

- Pedidos sin asignar si no coinciden con ninguna zona disponible.

### Aprobación del Cambio:

- **Viable:** Sí.
- **Impacto:** Alto.
- **Prioridad:** Alta.
- **CCB determina:** Válido.

## Revisión y Cierre

### Validación de los Requisitos Implementados



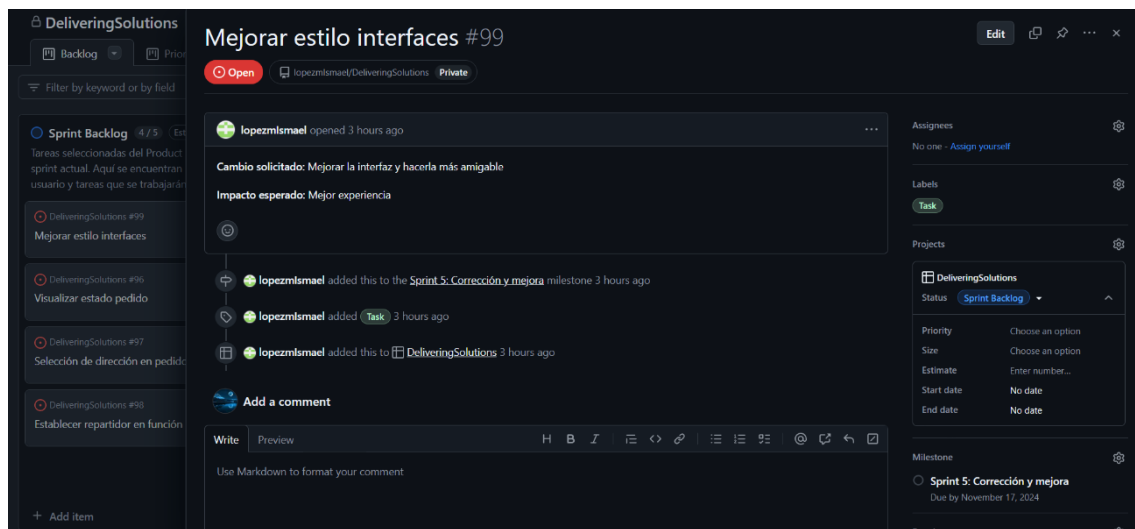
- Pruebas realizadas para garantizar que:
  - Solo se muestren pedidos correspondientes a las zonas del repartidor.
  - No se puedan seleccionar pedidos fuera de la zona asignada.

**Resultado:** Asignación correcta y funcionalidad implementada.

## 5.4. MEJORAR ESTILO DE INTERFACES

**Cambio solicitado:** Mejorar la interfaz y hacerla más amigable

**Impacto esperado:** Mejor experiencia



*Ilustración 6. Tarea de “Mejorar estilo Interfaces”*

## Evaluación del Cambio

### Análisis de Impacto:

- Ajustes en diseño y estilo para mejorar la consistencia visual y la experiencia del usuario.

### Riesgos:

- Incompatibilidades en dispositivos.
- Posible impacto en el rendimiento.

### Aprobación del Cambio:

- **Viable:** Sí.
- **Impacto:** Alto.
- **Prioridad:** Baja.
- **CCB determina:** No Válido.

## Revisión y Cierre

**Resultado:** Cierre sin cambios.

# DOCUMENTACIÓN SPRINT 6

**Fecha:** 18 de noviembre de 2024

Durante este sprint se llevó a cabo una semana de formación para la implementación de SonarCloud.

Las CR solicitadas en este sprint fueron:

1. **Implementar SonarCloud.**
2. **Mejorar estilo de interfaces.**
3. **Migración del sistema a la nube de AWS.**

## 6.1 Implementar SonarCloud

### Evaluación del Cambio

#### Análisis de Impacto:

- Este cambio implicó la implementación de SonarCloud para mejorar la gestión de calidad del código y permitir su análisis continuo.
- Se espera un aumento significativo en la capacidad del equipo para identificar y corregir errores de programación, problemas de seguridad y mantener altos estándares de calidad.

#### Riesgos:

- **Configuración inicial:** Posibles dificultades durante la integración.
- **Curva de aprendizaje:** El equipo debe familiarizarse con las herramientas y reportes que ofrece SonarCloud.
- **Incremento de tiempo:** La corrección de problemas reportados podría aumentar temporalmente el tiempo de desarrollo.

#### Aprobación del Cambio

- **Viable:** Sí.
- **Impacto:** Medio.
- **Prioridad:** Alta.
- **CCB determina:** Válido.

#### Implementación

- Se configuró SonarCloud de manera local.

#### Revisión y Cierre

#### Validación de los Requisitos Implementados

- Se realizaron pruebas para confirmar que SonarCloud genera reportes precisos de calidad del código y detecta problemas.

**Resultado:** SonarCloud está plenamente operativo y cumple con los requisitos establecidos.

## Validación de la Integridad de los Datos

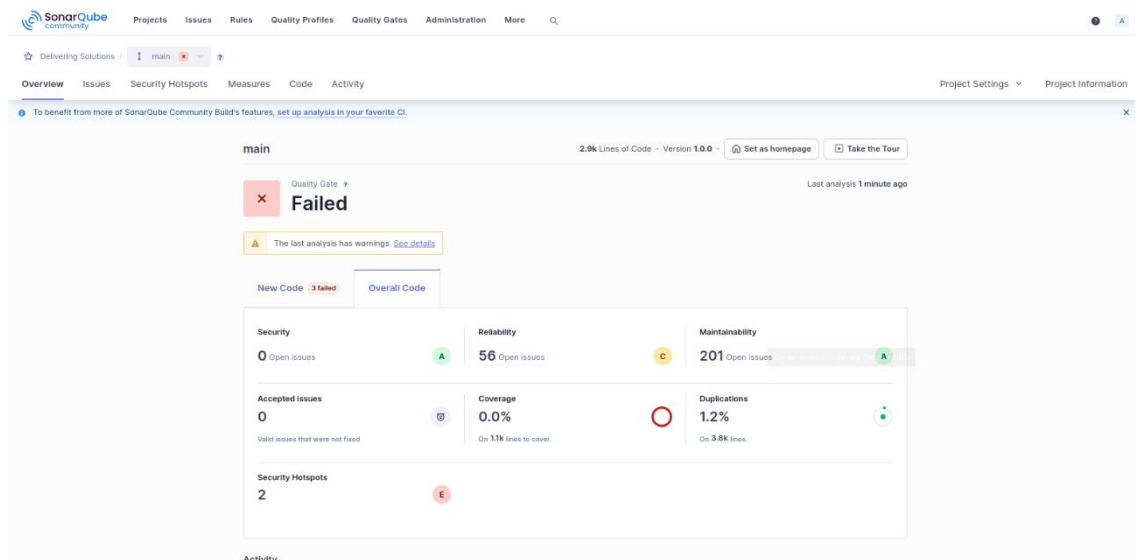
- Se verificó que los reportes sean consistentes y no generen falsos positivos o negativos.

**Resultado:** Los datos son precisos y confiables.

## Revisión de Código

- Auditoría del código realizada para identificar problemas detectados por SonarCloud y priorizar su resolución.

**Resultado:** Código revisado y actualizado según los estándares de calidad.



*Ilustración 7. Tarea de “Implementar SonarCloud”*

## 6.2 Mejorar Estilo de Interfaces

### Evaluación del Cambio

#### Análisis de Impacto:

- Este cambio implicó ajustes en diseño y estilo para mejorar la consistencia visual y la experiencia del usuario.
- Incluyó cambios en colores, tipografías y componentes de la interfaz, con el objetivo de hacerla más amigable y atractiva.

#### Aprobación del Cambio:

- **Viable:** Sí.
- **Impacto:** Alto.
- **Prioridad:** Baja.
- **CCB determina:** Válido, ya que, al haberse completado todas las tareas previas, se empezó a implementar la mejora de la interfaz.

### Implementación

- Se rediseñaron las pantallas principales para mejorar la experiencia del usuario.
- Se crearon nuevos botones para mejorar la navegabilidad.
- Se validó la consistencia visual en dispositivos de diferentes tamaños y resoluciones.

### Revisión y Cierre

#### Validación de los Requisitos Implementados

- Se realizaron pruebas para garantizar que los cambios visuales mejoraran la experiencia del usuario.

**Resultado:** Los cambios implementados son visualmente consistentes y funcionales en todas las plataformas probadas.

#### Validación de la Integridad de los Datos

- Confirmación de que los cambios en la interfaz no afectaron la integridad de los datos ni el rendimiento del sistema.

**Resultado:** No se encontraron inconsistencias ni errores.

#### Revisión de Código

- Se validó que las modificaciones cumplieran con las mejores prácticas de desarrollo y que el código fuera escalable.

**Resultado:** Código revisado y aprobado.

## 6.3 Migración del Sistema a la Nube de AWS

### Evaluación del Cambio

#### Análisis de Impacto:

- Este cambio implicó la migración de la aplicación, la base de datos y SonarCloud a la nube de AWS.
- Permite la compartición de recursos entre los stakeholders y la liberación del sistema en producción para su uso global.

#### Riesgos:

- **Configuración inicial:** Posibles dificultades durante la migración completa del sistema.
- **Cambio de la BBDD:** Fue necesario cambiar de SQLite a MySQL, utilizando una instancia RDS.

#### Aprobación del Cambio

- **Viable:** Sí.
- **Impacto:** Alto.
- **Prioridad:** Alta.
- **CCB determina:** Válido.

### Implementación

- Se migró la base de datos de SQLite a MySQL, configurando una instancia RDS en AWS.
- Se desplegó la aplicación en instancias EC2 de AWS.
- Se configuró SonarCloud para operar de manera conjunta con los recursos en la nube.

## Revisión y Cierre

### Validación de los Requisitos Implementados

- Pruebas realizadas para garantizar que la migración del sistema funcione correctamente en la nube.
  - Validación de acceso a la base de datos.
  - Pruebas de rendimiento de la aplicación en instancias de AWS.

**Resultado:** Migración exitosa con rendimiento optimizado.

### Validación de la Integridad de los Datos

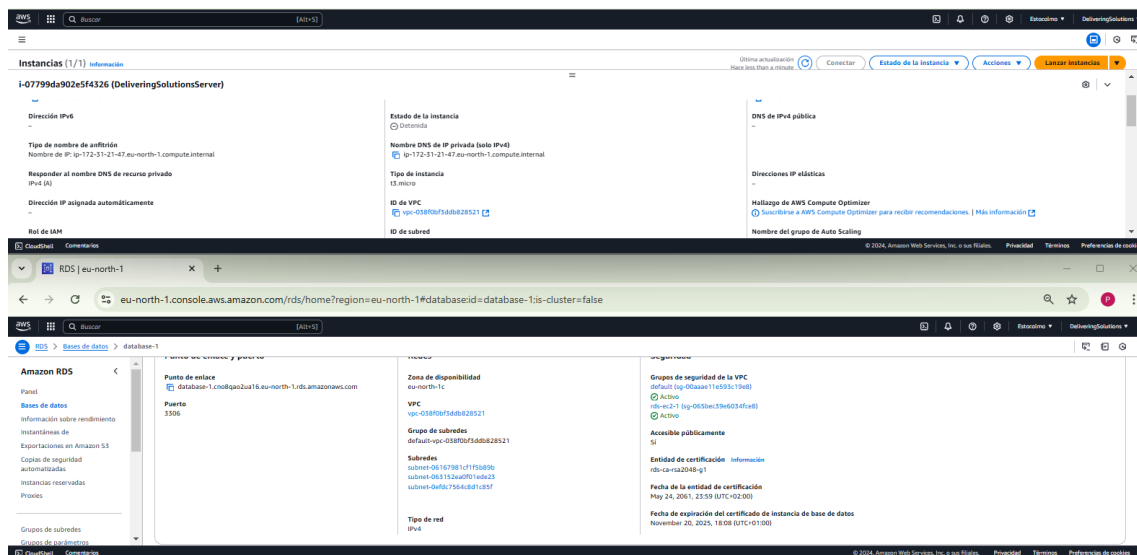
- Verificación de que los datos migrados sean consistentes y precisos.

**Resultado:** Datos almacenados y accesibles sin inconsistencias.

### Revisión de Código

- Revisión de las configuraciones implementadas para garantizar escalabilidad y seguridad.

**Resultado:** Configuraciones aprobadas y funcionales.



*Ilustración 8. Tarea de “Migrar Sistema”*

## Validaciones Finales

- **Implementación exitosa de SonarCloud:** Los reportes y análisis continuos ya están operativos.

- **Migración completada:** La aplicación y la base de datos están funcionando correctamente en AWS, con una mejora significativa en la escalabilidad.
- **Estilo de interfaces:** Los cambios implementados mejoraron la experiencia del usuario y fueron validados por el equipo de diseño.

## DOCUMENTACIÓN SPRINT 7

**Fecha:** 1 de diciembre de 2024

Durante este sprint se solucionaron los issues que mostraba SonarQube y se establecieron los test iniciales.

Las CR solicitadas en este sprint fueron:

1. **Solucionar Issues SonarQube.**
2. **Generar test iniciales para todos los gestores.**

### 7.1 Solucionar Issues SonarQube

#### Evaluación del Cambio

##### Análisis de Impacto:

- Este cambio implicó revisar y corregir los issues detectados por SonarQube en la calidad del código.
- Las correcciones incluyeron problemas de mantenibilidad, duplicación de código y otros errores menores reportados.

##### Riesgos:

- **Interrupción en desarrollo:** La corrección de los problemas identificados podría interferir temporalmente con el desarrollo de nuevas funcionalidades.
- **Falsos positivos:** Algunos issues podrían no representar errores reales y requerirían revisiones manuales adicionales.

#### Aprobación del Cambio

- **Viable:** Sí.
- **Impacto:** Alto.
- **Prioridad:** Alta.
- **CCB determina:** Válido.

#### Implementación

- Se realizaron revisiones exhaustivas del código basándose en el reporte de SonarQube.
- Se corrigieron 201 issues relacionados con mantenibilidad, 56 con fiabilidad, 2 issues de seguridad y duplicación de código.
- Se redujo la duplicación del código hasta el 1,1%.

## Revisión y Cierre

### Validación de los Requisitos Implementados

- Pruebas realizadas para garantizar que los cambios no afectaran otras funcionalidades.
  - Reanálisis del código en SonarQube tras las correcciones.
  - Confirmación de la resolución de todos los issues detectados inicialmente.

**Resultado:** Los issues se solucionaron correctamente y SonarQube reportó un incremento en la calidad general del código.

### Validación de la Integridad de los Datos

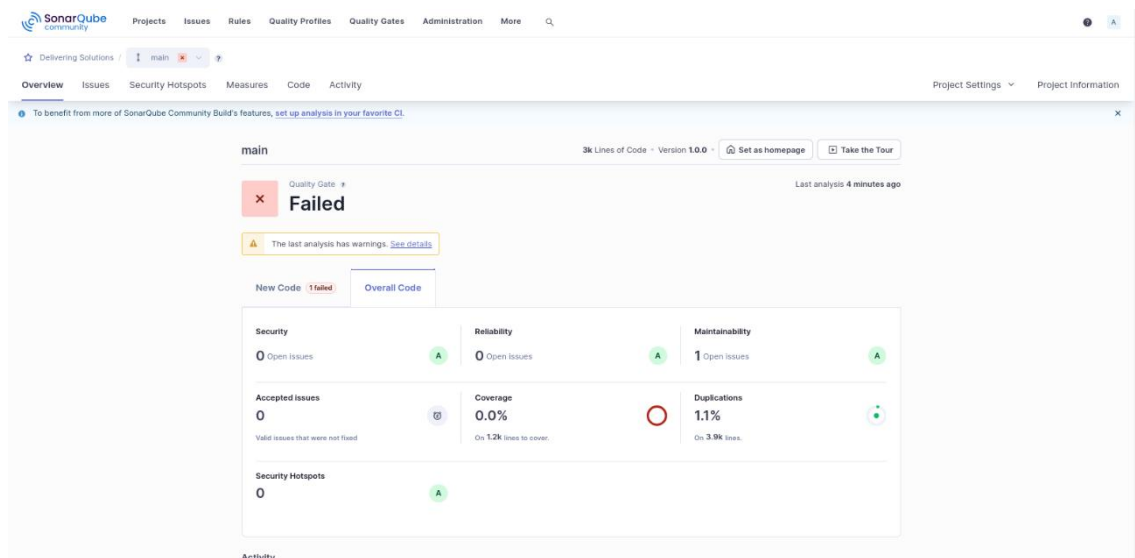
- Verificación de que las correcciones realizadas no introdujeran errores adicionales ni inconsistencias en el sistema.

**Resultado:** Datos y funcionalidades consistentes.

### Revisión de Código

- Auditoría final para garantizar que se siguieron las mejores prácticas de desarrollo.

**Resultado:** Código revisado y aprobado con 1 issue leve.



*Ilustración 9. Corrección Sonar*

## 7.2 Generar Test Iniciales para Todos los Gestores

### Evaluación del Cambio

#### Análisis de Impacto:

- Este cambio implicó crear tests unitarios y de integración para validar el correcto funcionamiento de los gestores principales.
- Los tests iniciales cubrirán casos básicos para garantizar estabilidad en futuras implementaciones.

### Riesgos:

- **Cobertura incompleta:** Al tratarse de tests iniciales, algunos casos extremos podrían no estar contemplados.
- **Errores en los tests:** La implementación de tests incorrectos podría generar falsos positivos.

### Aprobación del Cambio

- **Viable:** Sí.
- **Impacto:** Medio.
- **Prioridad:** Alta.
- **CCB determina:** Válido.

### Implementación

- Se desarrollaron tests iniciales para los gestores de usuarios, pedidos y zonas.
- Se configuró un pipeline para ejecutar los tests automáticamente en cada build.
- Se logró una cobertura inicial del 53,6% del código.

### Revisión y Cierre

#### Validación de los Requisitos Implementados

- Pruebas realizadas para confirmar que los tests se ejecutan correctamente y cubren los casos esperados.

**Resultado:** Los tests iniciales se ejecutaron satisfactoriamente, detectando y corrigiendo errores en escenarios previamente no cubiertos, como por ejemplo realizar un pedido sin items y con coste 0.

#### Validación de la Integridad de los Datos

- Verificación de que los tests no generaron conflictos en el flujo de desarrollo.

**Resultado:** Sistema estable con tests funcionales.

#### Revisión de Código

- Auditoría del código de los tests para confirmar su calidad y efectividad.

**Resultado:** Tests revisados, aprobados e integrados. 82 nuevos issues.



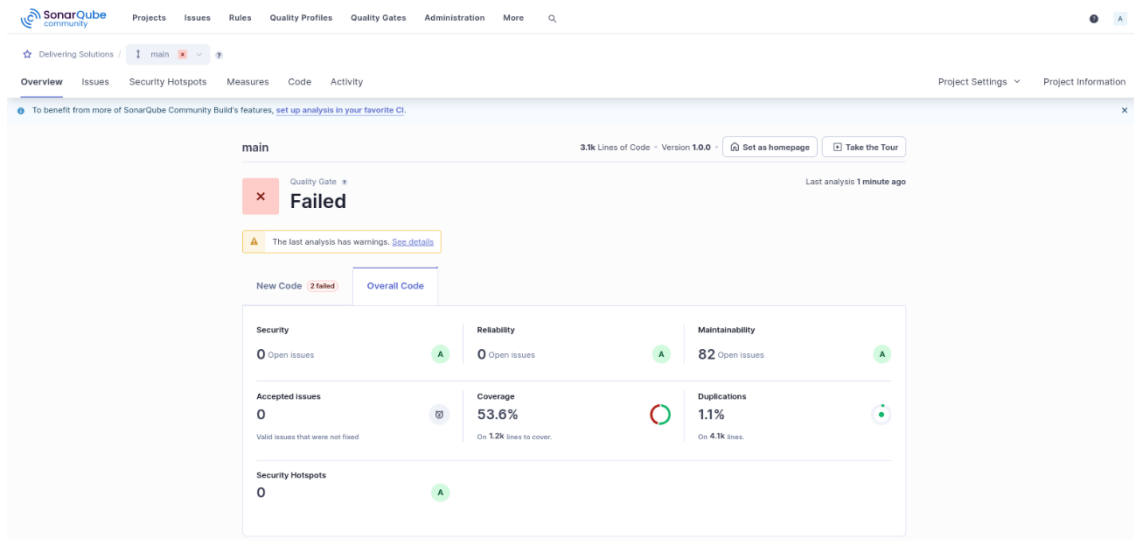


Ilustración 10. Primeros Test

## Validaciones Finales

- **Corrección de issues SonarQube:** Se alcanzó un nivel satisfactorio de calidad del código.
- **Tests iniciales implementados:** Se logró una base de pruebas funcional para futuros desarrollos.

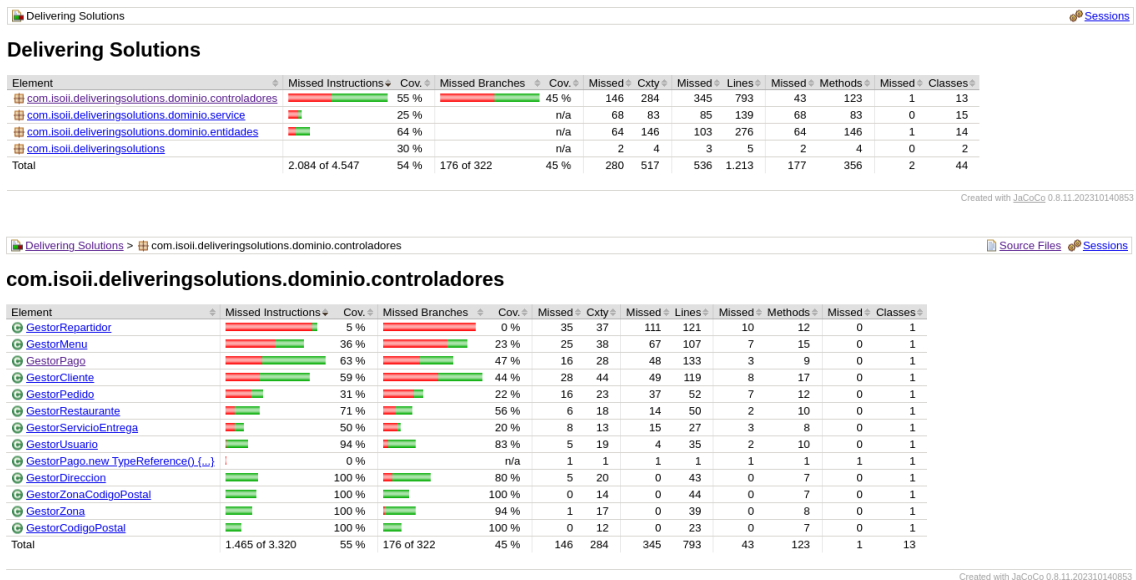


Ilustración 11. Cobertura con JaCoCo

# DOCUMENTACIÓN SPRINT 8

**Fecha:** 30 de diciembre de 2024

Durante este sprint se solucionaron los issues que mostraba SonarQube y se generaron los tests finales.

Las CR solicitadas en este sprint fueron:

1. **Solucionar Issues SonarQube.**
2. **Generar test finales para todos los gestores.**

## 8.1 Solucionar Issues SonarQube

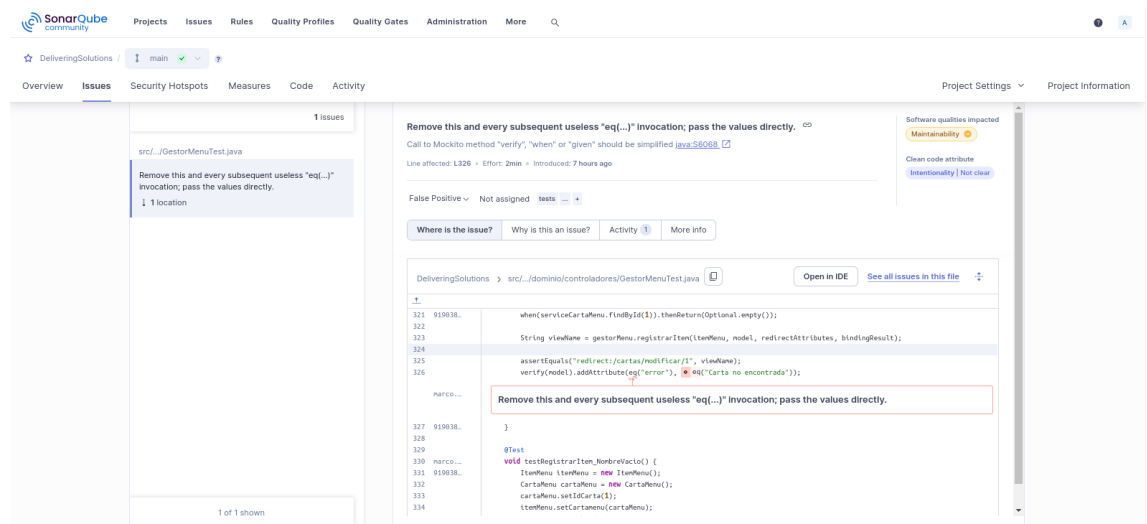
### Evaluación del Cambio

#### Análisis de Impacto:

- Este cambio implicó una revisión exhaustiva del código para solucionar los problemas reportados por SonarQube.
- Los ajustes realizados mejoraron significativamente la mantenibilidad y duplicación del código.

#### Riesgos:

- **Falsos positivos:** Algunos problemas identificados, como los 22 relacionados con el uso de mockito matchers, no representaban errores reales y fueron marcados como aceptados.
- **Impacto en la integración:** Cambios extensivos podrían afectar otras funcionalidades.



*Ilustración 12. Falso positivo*

### Aprobación del Cambio

- **Viable:** Sí.
- **Impacto:** Alto.
- **Prioridad:** Alta.
- **CCB determina:** Válido.

### Implementación

- Se solucionaron los issues identificados en el SonarQube inicial y se marcó como "aceptados" los 22 issues relacionados con Mosquitto al ser falsos positivos.
- Se optimizó el código, logrando una cobertura del 88.1% y reduciendo la duplicación al 1.0%.
- Se aseguró la confiabilidad y la mantenibilidad con un puntaje "A" en todas las categorías.

## Revisión y Cierre

### Validación de los Requisitos Implementados

- Pruebas realizadas para confirmar la resolución de los problemas y la mejora general de calidad.
  - Validación del nuevo análisis de SonarQube.
  - Confirmación de la disminución de issues abiertos y duplicación de código.

**Resultado:** Los problemas se resolvieron satisfactoriamente, marcándose los falsos positivos como aceptados, y se mejoró la calidad del código.

### Validación de la Integridad de los Datos

- Verificación de que los cambios realizados no introdujeran nuevos problemas.

**Resultado:** Integridad del código garantizada.

### Revisión de Código

- Auditoría del código completada para garantizar el cumplimiento de estándares.

**Resultado:** Código optimizado, revisado y aprobado.

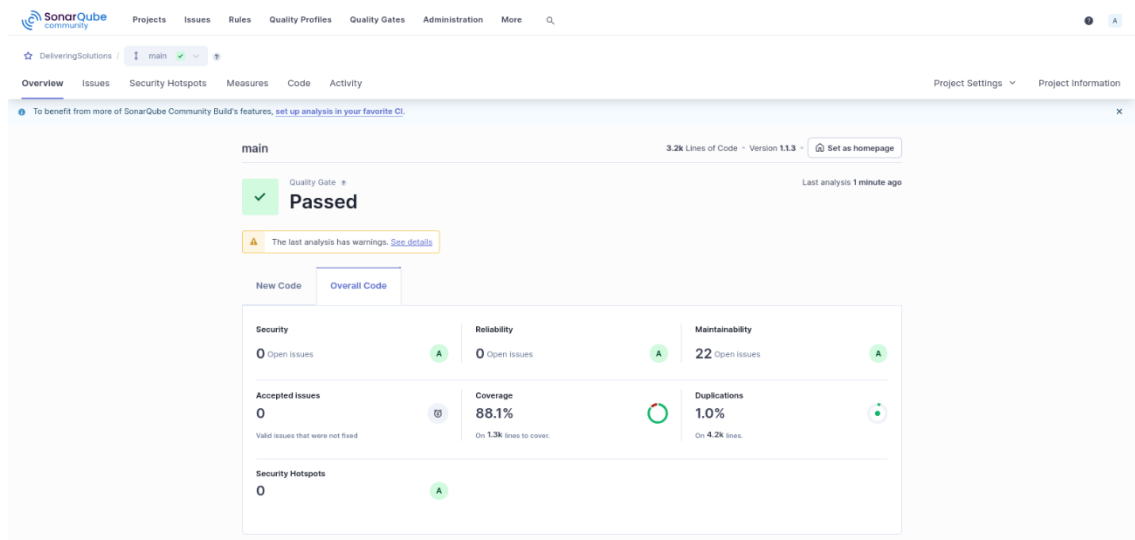


Ilustración 13. Segunda ejecución de SonarQube

## 8.2 Generar Tests Finales para Todos los Gestores

### Evaluación del Cambio

#### Análisis de Impacto:

- Este cambio implicó la generación de tests unitarios y de integración para cubrir casos extremos y asegurar la estabilidad del sistema.
- La implementación de los tests finales incrementó la cobertura de código a un nivel casi completo.

#### Riesgos:

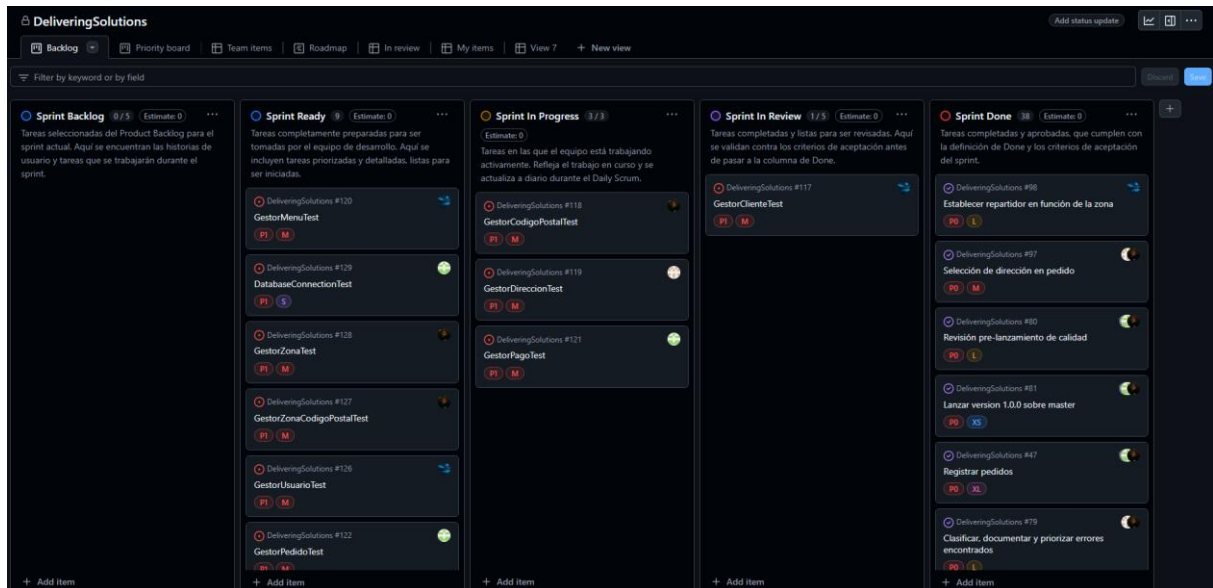
- **Casos no cubiertos:** Existe la posibilidad de que algunos escenarios específicos no hayan sido incluidos.
- **Errores en los tests:** Tests mal diseñados podrían generar falsos positivos.

#### Aprobación del Cambio

- **Viable:** Sí.
- **Impacto:** Medio.
- **Prioridad:** Media.
- **CCB determina:** Válido.

#### Implementación

- Se desarrollaron tests finales para los gestores de usuarios, pedidos, zonas y productos.
- Se logró una cobertura del 88.1% del código, con pruebas diseñadas para cubrir los escenarios críticos.
- Se configuró el pipeline de CI/CD para ejecutar los tests automáticamente.



*Ilustración 14. Inicio Sprint 8*

## Revisión y Cierre

### Validación de los Requisitos Implementados

- Pruebas realizadas para asegurar que los tests cubran todos los escenarios necesarios.
  - Ejecución exitosa de los tests finales.

**Resultado:** Los tests finales cumplen con los requisitos establecidos y detectaron errores previos que fueron corregidos.

### Validación de la Integridad de los Datos

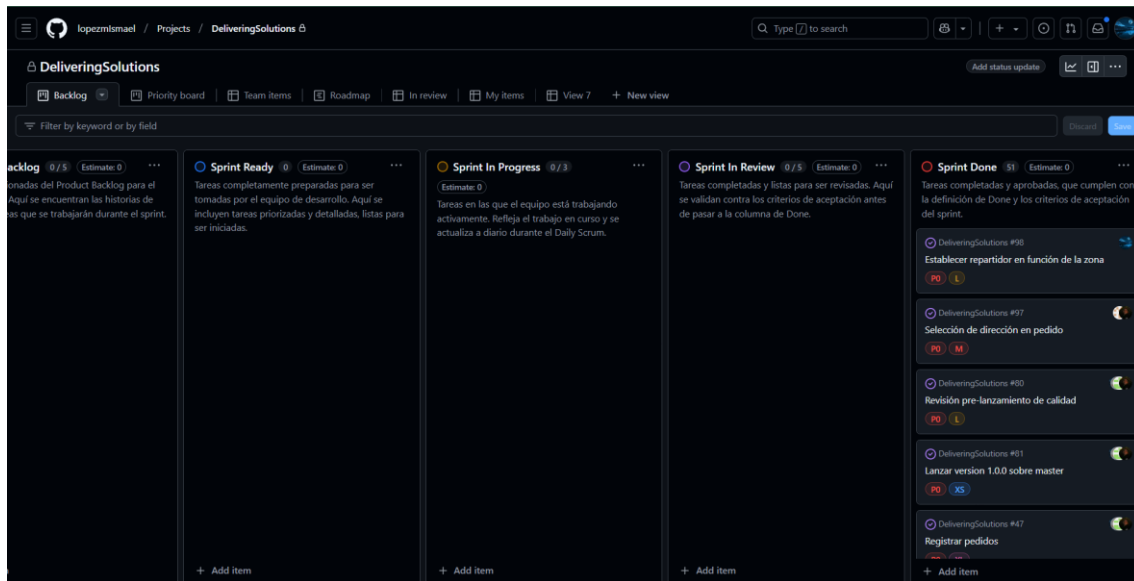
- Confirmación de que los tests no interfirieron con el funcionamiento general del sistema.

**Resultado:** Sistema estable y validado.

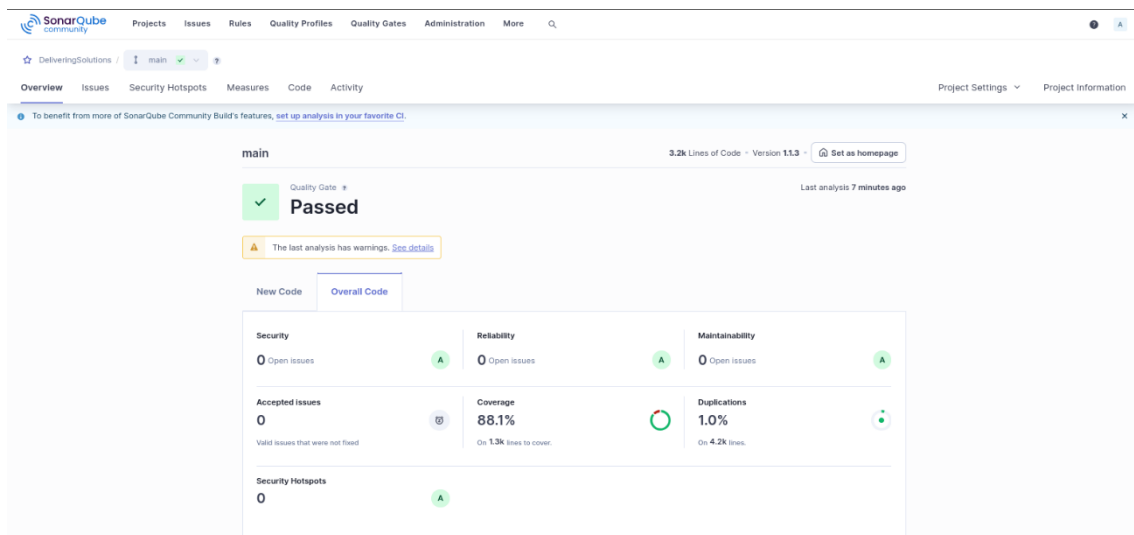
### Revisión de Código

- Revisión de los tests generados para garantizar su efectividad y adherencia a las mejores prácticas.

**Resultado:** Tests revisados y aprobados.



*Ilustración 15. Sprint 8 Final*



*Ilustración 16. Ejecución final SonarQube*

## Validaciones Finales

- **Resolución de issues SonarQube:** Calidad del código mejorada significativamente, con falsos positivos gestionados correctamente.
- **Generación de tests finales:** Cobertura casi completa y sistema estable.

## Delivering Solutions

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes	
<a href="#">com.isoii.deliveringsolutions.dominio.controladores</a>	<div><div></div></div>	55 %	<div><div></div></div>	45 %	146	284	345	793	43	123	1	13	
<a href="#">com.isoii.deliveringsolutions.dominio.service</a>	<div><div></div></div>	25 %	<div><div></div></div>	n/a	68	83	85	139	68	83	0	15	
<a href="#">com.isoii.deliveringsolutions.dominio.entidades</a>	<div><div></div></div>	64 %	<div><div></div></div>	n/a	64	146	103	276	64	146	1	14	
<a href="#">com.isoii.deliveringsolutions</a>	<div><div></div></div>	30 %	<div><div></div></div>	n/a	2	4	3	5	2	4	0	2	
Total		2.084 of 4.547	54 %	176 of 322	45 %	280	517	536	1.213	177	356	2	44

Created with JaCoCo 0.8.11.202310140853

[Delivering Solutions](#) > [com.isoii.deliveringsolutions.dominio.controladores](#)

[Source Files](#) [Sessions](#)

## com.isoii.deliveringsolutions.dominio.controladores

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes	
<a href="#">GestorRepartidor</a>	<div><div></div></div>	5 %	<div><div></div></div>	0 %	35	37	111	121	10	12	0	1	
<a href="#">GestorMenu</a>	<div><div></div></div>	36 %	<div><div></div></div>	23 %	25	38	67	107	7	15	0	1	
<a href="#">GestorPago</a>	<div><div></div></div>	63 %	<div><div></div></div>	47 %	16	28	48	133	3	9	0	1	
<a href="#">GestorCliente</a>	<div><div></div></div>	59 %	<div><div></div></div>	44 %	28	44	49	119	8	17	0	1	
<a href="#">GestorPedido</a>	<div><div></div></div>	31 %	<div><div></div></div>	22 %	16	23	37	52	7	12	0	1	
<a href="#">GestorRestaurante</a>	<div><div></div></div>	71 %	<div><div></div></div>	56 %	6	18	14	50	2	10	0	1	
<a href="#">GestorServicioEntrega</a>	<div><div></div></div>	50 %	<div><div></div></div>	20 %	8	13	15	27	3	8	0	1	
<a href="#">GestorUsuario</a>	<div><div></div></div>	94 %	<div><div></div></div>	83 %	5	19	4	35	2	10	0	1	
<a href="#">GestorPago.new_TypeReference().{...}</a>	<div><div></div></div>	0 %	<div><div></div></div>	n/a	1	1	1	1	1	1	1	1	
<a href="#">GestorDireccion</a>	<div><div></div></div>	100 %	<div><div></div></div>	80 %	5	20	0	43	0	7	0	1	
<a href="#">GestorZonaCodigoPostal</a>	<div><div></div></div>	100 %	<div><div></div></div>	100 %	0	14	0	44	0	7	0	1	
<a href="#">GestorZona</a>	<div><div></div></div>	100 %	<div><div></div></div>	94 %	1	17	0	39	0	8	0	1	
<a href="#">GestorCodigoPostal</a>	<div><div></div></div>	100 %	<div><div></div></div>	100 %	0	12	0	23	0	7	0	1	
Total		1.465 of 3.320	55 %	176 of 322	45 %	146	284	345	793	43	123	1	13

Created with JaCoCo 0.8.11.202310140853

Ilustración 17. Cobertura final JaCoCo