

Universidad ORT Uruguay
Facultad de Ingeniería

Ingeniería de Software Ágil 2

Análisis de Issues

Clavijo, Tomás (235426)

Barreto, Sofía (258216)

Facello, Bruno (260694)

Docente: Alvaro Ortas

2023

Índice

Introducción.....	3
Análisis de deuda técnica.....	4
Análisis de errores, patrones y principios.....	4
NDepend: Calidad del diseño.....	4
Cobertura de pruebas unitarias.....	5
Testing exploratorio: Resumen de Issues.....	5

Introducción

Este documento reúne la información correspondiente al análisis de deuda técnica, consistente en identificar issues que se ingresarán posteriormente en GitHub con su descripción y su respectiva clasificación (bugs - severidad -, mejoras, problemas de estándares, etc). Así como también las estrategias que se utilizaron para hallarlos. Es decir, posterior a la revisión de la solución provista, indicaremos cuáles son, dónde están, y qué solución ameritan los aspectos a mejorar.

Análisis de deuda técnica

Análisis de errores, patrones y principios

En la documentación brindada por quienes llevaron adelante el proyecto manifiestan que solucionaron los errores que encontraron en la primera etapa, y pudieron implementar todas las funcionalidades sin inconveniente alguno.

Aseguran haber utilizado patrones como Strategy con el objetivo de definir una familia de algoritmos, encapsularlos y hacerlos intercambiables en tiempos de ejecución. También mencionan el uso de patrones GRASPs, asignándole responsabilidades a las clases que tienen la información necesaria para cumplir con ellas; y buscando siempre la alta cohesión.

NDepend: Calidad del diseño

Basándonos en las métricas reportadas en la documentación y en la duración del proyecto inicial, podemos concluir que las métricas de diseño muestran una perspectiva positiva.

En el diagrama de “abstracción vs inestabilidad” existe un único paquete en zona de dolor, siendo este el del dominio. Debido a lo que representa no modificaría en gran medida el proyecto para realizar nuevas abstracciones.

En cuanto al análisis realizado acerca de la dependencia entre paquetes, se observa que hay dos paquetes comprometidos, los cuales no estarían cumpliendo con SDP (Principio de dependencias estables). El primero sería ArenaGestor.Domain, depende del paquete ArenaGestor.BusinessHelpers y el otro sería la ArenaGestor.API que tiene dos dependencias con paquetes más inestables como son ArenaGestor.BusinessFactory y ArenaGestor.DataAccessFactory.

Análisis de código estático

Se realizó un análisis del sistema y el código y se llegó a la conclusión de que el sistema fue realizado en español pero las excepciones y errores son lanzadas en inglés, agregando dificultad al usuario para entender el problema y agregando una incoherencia en el sistema.

Cobertura de pruebas unitarias

El front-end no está probado unitariamente, sólo funcionalmente. Esto puede significar una deuda técnica, ya que podrían existir múltiples bugs que requieren de horas extra para poder solucionarlos.

En cuanto al back-end se observa que cumplen con un muy buen porcentaje de cobertura, siendo de un 99,02% (superior a 95%). El único paquete que no cumple con pruebas unitarias es ArenaGestor.BusinessHelper; quienes realizaron el proyecto justifican que ArenaGestor.BusinessLogic utiliza sus métodos y tiene una cobertura de 99,04%; lo cual les da confianza en que funciona correctamente.

Testing exploratorio: Resumen de Issues

Como se mencionó en la guía de manejo del repositorio, para reportar todos los bugs y errores encontrados usaremos la sección de Issues de GitHub, habiendo ingresado los mismos:

1. Creación de banda : Como administrador quiero poder seleccionar un rol de artista perteneciente a una banda, para poder crear la misma.

i. Al seleccionar un rol, la opción se ve como el Id

Severidad : Leve

Prioridad : Baja

Tipo : Bug visual

2. Importación y exportación de conciertos: Como administrador quiero poder importar y exportar conciertos, para poder ver los conciertos registrados en sistemas externos.

Severidad : Mayor

Prioridad : Alta

Tipo : Bug

3. Registro: Como usuario quiero poder registrarme para tener acceso a las funcionalidades de mi rol.

Severidad : Mayor

Prioridad : Alta

Tipo : Bug

4. Eliminar género creado: Como administrador quiero poder eliminar un género, para que no sea utilizado en nuevos eventos.

Severidad : Menor

Prioridad : Media

Tipo : Mejora

5. Creación de usuario: Como administrador, quiero poder dar de alta a un usuario, para que el usuario pueda ingresar al sistema.

a. En el caso de poner el mail incorrecto

Severidad : Mayor

Prioridad : Media

Tipo : Bug

6. Filtrado por fechas: Como espectador, quiero poder filtrar bandas y artistas por nombres, para ver el detalle completo incluyendo el listado de sus conciertos.

Severidad : Crítico

Prioridad : Inmediata

Tipo : Bug

7. Filtrado de Tour: Como espectador quiero filtrar bandas y artistas por nombre para ver el detalle completo incluyendo el listado de sus conciertos

Severidad : Mayor

Prioridad : Alta

Tipo : Bug

8. Datos de Usuarios: Como usuario no espectador quiero poder editar mis datos para dejarlos actualizados

Severidad : Menor

Prioridad : Media

Tipo : Mejora

9. Compra de Tickets: Como espectador quiero comprar: Como usuario no espectador quiero poder comprar tickets utilizando el botón de "comprar tickets" para ir a un concierto

Severidad : Mayor

Prioridad : Alta

Tipo : Mejora

10. Baja de Música: Como usuario administrador quiero poder leer el mensaje correcto de error para entender que está ocurriendo.

Severidad : Menor

Prioridad : Media

Tipo : Bug

11. Mensaje de error en editar banda : Como usuario administrador quiero poder leer el mensaje correcto de error para entender que esta ocurriendo

- a. Cuando no se selecciona una artista ocurre un error de que el identificador debe ser mayor a cero, lo que no le informa del error al usuario.

Severidad : Menor

Prioridad : Baja

Tipo : Bug

12. Como espectador quiero comprar: Como usuario no espectador quiero poder comprar tickets utilizando el botón de "comprar tickets" para ir a un concierto

Severidad : Normal

Prioridad : Alta

Tipo : Mejora

13. Corrección estándares de código

- a. Sustituir "new Ejemplo()" por "new ()"
b. Crear una excepción que herede de Exception en vez de utilizar NullPointerException
c. Sustituir new Guid() por Guid.Empty
d. Atributos en clase que no son utilizados

- e. Métodos vacíos sin implementación y sin comentario de explicación de porqué fue realizado así

14. Titulo en comprar tickets: Como usuario quiero leer el titulo correcto para entender que estoy haciendo

Severidad : Menor

Prioridad : Baja

Tipo : Bug visual

15. No se ve la cantidad de tickets disponibles: Como usuario quiero saber cuantos tickets hay disponibles para no pasar por el proceso de compra en el caso de que esté agotado

Severidad : Menor

Prioridad : Baja

Tipo : Bug visual

16. Error en editar nombre de artista: Como usuario administrador quiero leer el error correcto para entender qué está pasando

Severidad : Mayor

Prioridad : Alta

Tipo : Bug

17. Error al cambiar nombre de artista, concierto o solista: Como usuario (artista, concierto o banda) quiero poder leer el mensaje de error correcto para entender qué está pasando

Severidad : Menor

Prioridad : Baja

Tipo : Bug Visual

18. Error al editar concierto y no modificar la fecha: Como administrador quiero editar un concierto sin la necesidad de cambiar la fecha para poder cambiar el nombre del mismo.

Severidad : Mayor

Prioridad : Media

Tipo : Bug

19. Editar usuario permite hacer que un usuario no tenga rol: Como administrador quiero poder editar un usuario y que tenga al menos un rol para que la aplicación pueda seguir su flujo

Severidad : Mayor

Prioridad : Inmediata

Tipo : Bug

20. Editar el mail de un usuario: Como usuario administrador quiero poder editar el main de un usuario para que el mismo pueda acceder con el nuevo mail

Severidad : Menor

Prioridad : Baja

Tipo : Bug

21. Eliminar tu propio usuario: Como usuario administrador quiero no poder eliminar mi propio usuario para dejar al menos un usuario y que pueda seguir el flujo de la aplicación

Severidad : Mayor

Prioridad : Media

Tipo : Bug

22. Vender tickets a un usuario que no existe: Como usuario vendedor quiero no poder vender tickets a un mail inexistente para no desperdiciar los mismos.

Severidad : Mayor

Prioridad : Alta

Tipo : Bug

23. Logout desde artista: Como usuario artista quiero poder cerrar sesión para finalizar el flujo de la aplicación

Severidad : Crítico

Prioridad : Inmediata

Tipo : Bug