

Universidad ORT Uruguay

Facultad de Ingeniería

Obligatorio Ingeniería de Software Ágil 2

Informe Segunda entrega

[Repositorio](#)

Francisco Cagnoli – 238266

Martin Radovitzky - 246579

Hernán Yáñez – 242618

2022

Álvaro Ortas

Entregado como requisito para la obtención del título de Ingeniero en Sistemas

Bugs seleccionados

Para seleccionar los bugs a corregir para esta entrega se seleccionaron 2 de los de prioridad roja (los mas vitales). El primero fue el bug de issue numero #4 y título 'no redirecciona al hacer login' que consta que cuando se ingresaba como administrador no es redireccionado a la página de Admin, sino que se quedaba en la pagina de cliente. Este era uno de los issues más importantes ya que permitía a un administrador acceder a paginas y vistas para los clientes, permitiéndole hacer reservas y reseñas.

El segundo issue con numero #2 y título 'Permite publicación de reseña previo a la fecha de reserva' consta de que un usuario puede dejar una reserva sin haber ido a la residencia y antes a la fecha de su reserva. Esto es gravísimo ya que un usuario con fines maliciosos o enojo con cierta residencia puede dejar reseñas negativas sin haber ido al lugar y bajarle la puntuación a este.

Resultados obtenidos en el período

En este segundo periodo fuimos capaces de hacer la reparación de los bugs seleccionados, también se pudieron realizar las dos funcionalidades y sus respectivas pruebas en SpecFlow, siguiendo los criterios de aceptación creados en las cartas BDD. Finalmente, se hizo el cross-review tal que lo que uno programaba los otros validaban como product owner.

Desarrollo/mantenimiento del nuevo tablero usando BDD

En el tablero se tienen las columnas Backlog, To do, Requirement Definition, TestCase implementation, COD, Testing, Refactor y Done. Para comenzar se agrego la nueva columna de Backlog, que contiene toda las cards que no serán trabajadas en el periodo y no estén terminadas, luego pasan a TestCase implementation donde se definen los criterios de aceptación y se pasan a spec.

Las tarjetas que debían ser trabajadas en este periodo fueron sacadas del backlog y puestas en la columna To Do (columna que contiene lo que se debe hacer por el periodo), luego cuando se comienza a trabajar en la tarjeta definiendo sus requerimientos se pasa a la columna del Requirement Definition una vez terminada la ingeniería de requerimientos (criterios de aceptación, requerimientos) se pasa a TestCase implementation donde se crean los criterios y se pasan a spec, después pasa a COD donde se codifica y se crean los tests pertenecientes al TDD teniendo en cuenta que cumpla con los requerimientos y criterios de aceptación. Luego se pasa a la columna de testing donde se hace la revisión por parte del product owner y, si pasa la revisión y los test pasa a la columna Done, cerrando el issue y mergenado el pull request a develop.

Configuración/mantenimiento del pipeline y su vínculo con el tablero.

El principio del pipeline, comienza cuando se crea un nuevo issue y se le asigna al proyecto del board, lo cual hace que se agregue automáticamente a la columna de backlog. Luego la siguiente parte del

pipeline es cuando se crea una rama a partir de la card en el momento del requirement definition, después de esto se pasa por el tablero hasta llegar al testing y crear el pull request donde a través de las github actions se hace el build y corren las pruebas del proyecto, esta columna también recibe las cards que tengan un pull request abierto. Luego, en la columna de refactor las cards que se le reabra un pull request se asignan a esta y finalmente cuando se cierra el pull request van a la columna de done.

Dificultades encontradas y formas de solución

Las principales dificultades fueron en cuanto a tecnologías, dado que ninguno tenía experiencia con la automatización de tests ni con GitHub Actions. A pesar de esta brecha, no fue un desafío mayor al que implica aprender a utilizar una nueva tecnología, y como integrar todo. El único percance en particular fue que no logramos probar, a través de los SpecFlow tests, que esté funcionando correctamente la validación del rol del usuario. Asimismo, los criterios de aceptación los tuvimos que modificar constantemente para que funcionen los SpecFlow tests. Esto llevó a que las cards en la Kanban board se muevan mas y regresen a columnas que ya habían pasado.

Lecciones aprendidas trabajar y mejoras en el proceso

Esto lo discutimos en la retrospectiva, la cual grabamos, y dejamos un punteo siguiendo el método DAKI de las cosas a agregar, dejar de hacer, mejorar o seguir haciendo.

Link a retrospectiva de la entrega 2: <https://youtu.be/2xH2zSNAtPY>

Drop

- Nada

Add

- Grupo de WhatsApp para mejorar comunicación
- Build github action Angular
- Empezar a calcular las métricas
 - Se uso la información pero no se crearon las métricas.

Keep

- Mantener cross review product owner
- Test spec
- Github actions para los tests y el build del backend.
- Tablero con columnar backlog y test case implementation

Improve

- Decidir un lenguaje tipo para cada cosa (inglés o español)

- Commits, pull request, código inglés
 - Tablero, cards español
 - Nombre de las ramas son en español ya que se sacan de los issues.
- Comunicación:
 - Grupo de WhatsApp
- División de tareas.