

Introducción a las Metodologías de Desarrollo de Software

Trabajo Práctico Especial

ScrumGame

Descripción Detallada del Problema

Los juegos serios han surgido con el objetivo de mejorar el compromiso y el desempeño de los jugadores, ya que realizan una misión exacta, integrándose con la mecánica del juego, creando así una misión muy atractiva. Se ha investigado con mucho interés el uso de juegos serios en Ingeniería de Software para aumentar el compromiso de los desarrolladores. Por otro lado, el uso de juegos serios resulta de gran interés para capacitar y entrenar a futuros profesionales para experimentar situaciones a las que se podrían enfrentar en el desarrollo de software. Siguiendo esta línea, y apoyados en el uso de los celulares como una constante fuente de aprendizaje, se presenta *ScrumGame*, un juego serio desarrollado como una aplicación móvil para enseñar los principales conceptos de Scrum. Si bien existe una versión inicial del juego estable, el objetivo es extender la funcionalidad de *ScrumGame* para motivar y atraer a muchos más usuarios.

En primer lugar, el juego deberá contar con un módulo para loguear usuarios. En este sentido, además es importante que haya un acceso para un usuario administrador que gestione usuarios y contraseñas a través de una interfaz web.

Actualmente, el juego consta de niveles y cada nivel consta de módulos. Para superar un nivel se deberá completar los módulos de ese nivel. Cada módulo tiene diferentes actividades. Es importante que todo lo que hace un jugador tiene que quedar logueado junto con el tiempo que tardó en cada actividad. Esos datos tienen que poder ser accedidos por el administrador desde la interfaz web.

Para mejorar la jugabilidad del juego, se decidió agregar características que potencien la "gamificación" del juego. Para ello, la idea es ofrecer estadísticas sobre los logros obtenidos por los jugadores. Asimismo, el juego deberá contar con una tabla de posiciones con los puntajes obtenidos en una semana, donde cada jugador podrá ver el puntaje del resto de los jugadores. Ese puntaje se reiniciaría por semana. En esta línea, es importante contar con *insignias* para incentivar a los jugadores. Las *insignias* se pueden conseguir por semana completa de juego, por completar un módulo, entre otros criterios dependientes de cada insignia.

Con el objetivo de mantener a los usuarios "enganchados" con el juego, se debe enviar notificaciones para incentivar su uso y las interrelaciones entre los usuarios. Para llegar a mayor público, se deberá ofrecer la posibilidad de multilenguaje en *ScrumGame*. Asimismo, para mejorar la estética del juego, y la comprensión de las actividades, se debe poder agregar imágenes a las distintas actividades.

Finalmente, el equipo de desarrollo deberá explotar su capacidad creativa y proponer juegos nuevos para enriquecer *ScrumGame* y que sea mucho más atractivo.

Desarrollo del Proyecto

El proceso de desarrollo del TPE será organizado siguiendo Scrum. Se realizarán 1 warm-up y 3 sprints y las reviews están organizadas en el siguiente cronograma:

- **23 de Septiembre:** entrega warm-up con todas las user stories, definición del *criterio de done* y prototipo (sketch).
- **16 de Octubre:** entrega Sprint #1.
- **04 de Noviembre:** entrega Sprint #2 con diagrama de clases de lo implementado hasta el momento.
- **27 de Noviembre:** entrega Sprint #3 con diagrama de clases actualizado.

La asistencia al Sprint review de todos los miembros del equipo será condición necesaria (pero no suficiente) para la aprobación del TPE.

La propuesta es formar 2 equipos de Scrum, los cuales tendrán que interactuar para integrar la funcionalidad al final de cada Sprint. Si en una clase se hace la Sprint Review, en la siguiente clase se hará la Sprint Planning del siguiente sprint.

Los docentes oficiarán de Scrum Master y Product Owner. Todo el proyecto será gestionado a través de un proyecto Jira y un repositorio Git asociado con dicho proyecto. Deberán hacer un fork de dicho repositorio. Los commits al repositorio deberán referenciar el ID de la tarea correspondiente de Jira. Además del código fuente y la información del Jira, los artefactos a entregar en cada sprint deberán incluir, cuando corresponda, el resumen de la sprint planning y la sprint retrospective. Para la realización de los diagramas de UML se sugiere el uso de la herramienta Google Draw.io.

Repositorio GitHub

Link: <https://github.com/grodriguez86/ScrumIoS>