Escuela Colombiana de Ingeniería
Arquitecturas de Software – ARSW
Proyecto de curso – Sprint 2
2020 - 2

Las siguientes son las condiciones de evaluación del Sprint 2 para el proyecto de ARSW.

Arquitectura Backend

La arquitectura del *backend* del proyecto debe separar claramente las capas de control, lógica, caché, persistencia, y garantizar un bajo acoplamiento entre las mismas. La capa de 'caché', a diferencia de la de persistencia, es responsable de mantener sólo información volátil requerida por la aplicación (por ejemplo, el estado de una partida, si se tratara de un juego).

Arquitectura Frontend

De la misma manera, el *frontend* debe estar correctamente modularizado y con un diseño por capas.

Continuidad de desarrollo en GitHub

Mantener un sistema de control de versiones GIT, en GitHub, en la cual el equipo pueda evidenciar el continuo trabajo en el proyecto así como su aporte individual.

Despliegue en Heroku y Manual de Uso

Como primero se hará una revisión -sin participación del equipo de desarrollo- de lo que esté desplegado en la nube, se debe tener cuidado de incluir en el archivo README.md del repositorio, usando formato Markdown:

- Nombre, asignatura, descripción, autores.
- Enlace a la aplicación en Heroku.
- Arquitectura de la aplicación especificando:
    - Diagrama de clases
    - Diagrama de Entidad relación
    - Diagrama de despliegue
    - Diagrama de casos de uso
    - Diagrama de Componentes
- Indicaciones básicas de uso de la aplicación (usuarios disponibles por defecto, instrucciones de manejo, etc).

Análisis estático de código y Pruebas

- Registrar el repositorio en el sistema de análisis estático de calidad de código CODACY: https://www.codacy.com
- El proyecto debe realizarse sobre una línea de producción que contemple TDD, Integración, y Entrega Continua. Esto implica:
    - Hacer un desarrollo dirigido por pruebas (por ahora, de la capa lógica), en donde se harán especificaciones de lo que se implementará, y sobre éstas se hará un diseño basado en clases de equivalencia y condiciones de frontera para las

pruebas. Las pruebas son indispensables principalmente si hay modelos que contengan reglas de la aplicación de lado del servidor.

Completitud Funcional

Se evaluara sobre el total de las historias de usuario no colaborativas propuestas y documentadas en Taiga o en la herramienta que hayan escogido para administrar el proyecto.