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

Sprint 1.

Fecha de inicio del Sprint: Enero 14 de 2020

Capacidad en tiempo de los integrantes durante el Sprint:

- Semana 9: 7.5h.
- Semana 10: 7.5h.
- Semana 11: 7.5h.
- Total Sprint 1 para cada equipo: 22.5h X Número de integrantes.

Requerimientos de entrega para el Sprint Review (Febrero 8):

- El proyecto debe seguir los elementos de gestión de SCRUM. Una vez se haya definido qué historias del BackLog del producto irán al BackLog del Sprint, el equipo debe hacer una planificación, y mantener un control de tiempos que permitan generar continuamente un Burndown Chart. Para esto, cada equipo debe crear un equipo en Teams – ARSW-2-2020-2-NOMBRE_PROYECTO-, y compartida (con su equipo y el usuario 'sebastian.henao@escuelaing.edu.co'). En la misma agregar una copia de la siguiente plantilla: https://docs.google.com/spreadsheets/d/1yZoKknLEC3vKegFZItBNNNkV1pOoZ5jh9ZS2aeMgghc/copy Puede aprender a usar la plantilla leyendo el siguiente tutorial.

- 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). Por ahora, como capa de persistencia se debe inyectar una implementación basada en el concepto de Stub (https://es.wikipedia.org/wiki/Stub), manteniendo los datos en memoria. De la misma manera, el 'front-end' debe estar correctamente modularizado y con un diseño por capas.

- Se debe entregar un diseño básico de la arquitectura que incluya:
  - Diagramas de casos de uso.
  - Diagrama entidad relación de la base de datos.
  - Diagrama de clases.
  - Diagramas de secuencia de alto nivel de las funcionalidades más importantes.
  - Diseño de la interfaz gráfica usando Wireframes, para ello revise la herramienta mockflow, aunque puede usar la herramienta que prefiera.

- Dado lo anterior, la planificación debería plantear una distribución de tareas a partir de dichas capas/módulos. Para la estimación de dichas tareas, tenga en cuenta el límite indicado anteriormente (22.5h x Número de integrantes). Si se considera que las historias de usuario seleccionadas lo superan, se puede replantear el alcance sugerido para el sprint.

- Aunque no es posible realizar la dinámica del '*daily meeting*' de SCRUM, se espera que la misma se realice como mínimo al inicio de las sesiones de clase. Por lo tanto, **las fallas** no sólo implicarán un registro en el reporte de asistencia, sino **una falla ante la dinámica del grupo de trabajo**.

- 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.
  - Mantener un sistema de control de versiones GIT, en una Organización GitHUB, en la que el equipo haga integración continua de sus avances.
  - Registrar el repositorio en el sistema de análisis estático de calidad de código CODACY: https://www.codacy.com
  - Configurar un sistema de Integración Continua (Circle.CI) que, ante los cambios, ejecute la fase de pruebas del proyecto y haga un despliegue automático en Heroku. Documentación: https://github.com/ARSW-ECI/SpringBoot_CircleCI_Heroku_CI

- Al finalizar el Sprint, junto con el avance del producto se deben entregar un avance del documento de la arquitectura del software: Arquitectura lógica, arquitectura física, Comportamiento dinámico. Para esto, agregue en la carpeta compartida de su proyecto una copia de: https://docs.google.com/document/d/1xYc4JC6x12csR2Jm-vO8vCxZ3xTYzV6KZqbHm-UbiE4/copy

- A más tardar el 22 de Febrero se debe haber diligenciado el siguiente formulario, incluyendo el repositorio GITHUB oficial, e integrantes del equipos.
  https://forms.gle/8vMALwQFwX8jJALd6

- 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.
  - Indicaciones básicas de uso de la aplicación (usuarios disponibles por defecto, instrucciones de manejo, etc).
  - Enlace al entorno Circle.CI del proyecto como 'status badge': https://circleci.com/docs/1.0/status-badges/
  - Enlace al entorno Codacy del proyecto como 'status badge': https://support.codacy.com/hc/en-us/articles/212799365-Badges

Evaluación:

No se aceptarán proyectos que no tengan como base unas especificaciones claramente definidas, y unas historias de usuario con criterios de aceptación concretos (sin ambiguedades), medibles y verificables. Tampoco se aceptarán proyectos que no vengan respaldados por un sistema de control de versiones (historial GIT) que muestre el trabajo realizado por cada integrante. Aquellos que cumplan con este requisito, tendrán como evaluación:

- Criterios de aceptación de las historias de usuario planteadas para el Sprint: Definidos vs Cumplidos: 60%.
- Calidad del código (diseño, manejo de la concurrencia, etc): 10%
- Proceso de desarrollo: 15%,
- Contribución individual al proyecto. Este ítem afectará la evaluación de los dos anteriores (85%). Es decir, si hubo poca o ninguna contribución, la evaluación individual para el proyecto será escalada en la misma proporción.
- Calidad del avance en el documento de arquitectura: 15%.