PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2513 — Tecnologías y Aplicaciones Web

Proyecto Semestral - Entrega 2

Entrega

- **Fecha y hora:** Viernes 2 de junio de 2023, 20:00
- **Lugar:** Repositorio grupal en la organización del curso en GitHub

Objetivos

- **Construir** aplicaciones usando las tecnologías y herramientas disponibles.
- **Aplicar** principios y estándares para facilitar la interacción de software en la web.
- **Integrar** técnicas de desarrollo de software para construir aplicaciones web de alta calidad.

1. Descripción

Para esta entrega, cada equipo de estudiantes deberá trabajar en la definición e implementación del back-end que da soporte a las funcionalidades del juego en desarrollo. Esto implica modelar e implementar la base de datos, implementar la API que permite la interacción con la capa de persistencia, y desarrollar las funcionalidades del juego utilizando los elementos anteriores.

Las componentes de software a implementar en esta entrega son:

- **Base de datos en PostgreSQL**: Se debe diseñar un modelo apropiado para el contexto, además de documentar cómo levantar la base de datos para probar la aplicación. Para interactuar con la base de datos, se deberá utilizar Sequelize.
- **API RESTful para interactuar con la base de datos**: El front-end de la aplicación deberá conectarse a este servicio para consultar y almacenar los datos relevantes al modelo del punto anterior. Esta API deberá ser implementada en Koa.
- **Completar primera versión de front-end**: En caso de que existan elementos pendientes de desarrollar de la entrega anterior, estos deberán ser completados para esta entrega.

Además, durante el desarrollo del back-end de su juego, cada equipo deberá desarrollar documentación apropiada para cada parte de la aplicación. Esta puede ser incluída en el README del repositorio que contiene el código del back-end. Este documento deberá incluir:

- Instrucciones para instalar dependencias de la API, indicando comandos a ejecutar y una breve descripción de lo que hacen.
- Instrucciones para levantar la base de datos, indicando comandos y programas a utilizar, junto con una breve descripción.

- Documentación de la API, indicando (como mínimo) para cada *enpoint*: método HTTP a utilizar, ruta del *endpoint*, argumentos que recibe (y su formato) y lo que retorna este *endpoint* (y su formato). Se recomienda el uso de Swagger para documentar la API (ejemplo relevante a Koa) e incluir ejemplos en la documentación.

Para realizar preguntas técnicas, sobre requisitos o sobre el proyecto o entrega en general se deberá utilizar el repositorio oficial del curso en GitHub: **IIC2513/Syllabus**.

2. Entregables

Como equipo deberán entregar:

- Implementación del back-end del juego, incluyendo la base de datos en PostgreSQL y la API en Koa.
- Documentación para levantar la aplicación y montar la base de datos.
- Documentación y/o ejemplos de uso de los *endpoints* implementados.

3. Bandas

A continuación, se describe el criterio para entrar a cada banda. **Tengan en cuenta que los criterios son acumulativos y que para caer dentro de una banda es necesario tener todo lo de las bandas anteriores**. La nota correspondiente a cada una de estas bandas se encuentra descrita en el enunciado general del proyecto.

- **Banda D**: Implementación del modelo en la base de datos, incluyendo todas las entidades y sus relaciones. Se recomienda documentar el modelo en el README.
- **Banda C**: Implementación de lógica de juego, que puede contener algunos errores. Debe tener al menos un `GET` funcional para obtener el estado de una partida/juego y un `POST` funcional para enviar una de las jugadas de la partida.
- **Banda B**: Implementación de la lógica de juego sin errores. Algunos de los `POST` que el juego requiera para probar las jugadas que se implementaron pueden tener errores. Debe entregar documentación o un archivo JSON de prueba. Esta parte será evaluada con postman, por lo que eno deben tener aún la conexión back y front.
- **Banda A**: Uso de ESLint y Gitflow durante el desarrollo del proyecto. Además, no hay errores en los *endpoints* implementados.
- **Banda A+**: Solo para esta entrega se asignará una nota de 7,5 en caso de que haya un avance en la vista del tablero de juego. Los detalles sobre que debe incluir deben ser discutidos con el ayudante

4. Notas

- Se recomienda que cada equipo se ponga en contacto con su ayudante en caso de que requiera aclarar lo que se espera del proyecto en general y de la entrega en particular, además de aclarar detalles de la entrega que no estén claros.
- Posterior a la entrega, cada equipo deberá coordinar una reunión con su ayudante para recibir *feedback* de la entrega. Además, en esa misma instancia, se le pedirá a cada integrante que hable de detalles funcionales y técnicos de lo que entregaron. Si bien puede pasar que se dividan ciertas responsabilidades, se espera que manejen de manera general lo implementado. Es responsabilidad del grupo ponerse en contacto con el ayudante y gestionar la reunión. **El no reunirse con su ayudante antes de que inicie la siguiente entrega, significará un descuento del 50 % sobre la nota obtenida por el o los individuos**.

- Esta entrega tiene asociado **un (1) cupón de atraso**. En caso de no usarse, puede ser utilizado en futuras entregas. En esa línea, si no se ocupó el cupón de atraso para la anterior entrega, ahora se tienen dos (2). Los días de atraso incluyen fines de semana (utilizar un cupón de atraso un viernes, mueve la entrega al sábado, no al siguiente día hábil).