



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

IIC2513 — Tecnologías y Aplicaciones Web

# Proyecto Semestral - Entrega 3

## Entrega

- **Fecha y hora:** Viernes 30 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 completar la implementación del back-end, que da soporte a las funcionalidades del juego, y completar el front-end, que es la interfaz por medio de la que el juego es utilizable. Ambas piezas de software se deberán integrar para trabajar en conjunto.

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:** Este servicio deberá estar disponible para el front-end, para consultar y almacenar los datos relevantes al modelo del punto anterior. Esta API deberá ser implementada en [Koa](#). En caso de que existan elementos pendientes de desarrollar de la entrega anterior, estos deberán ser completados para esta entrega.
- **Front-end como UI para la interacción con usuarios:** Se debe implementar un front-end utilizando [React](#) para actuar como interfaz de usuarios (UI). Para su funcionamiento, deberá conectarse a la API (back-end). En caso de que existan elementos pendientes de desarrollar de la entrega anterior, estos deberán ser completados para esta entrega.

Al igual que en la entrega anterior, cada equipo deberá desarrollar documentación apropiada para cada pieza de software. Esta puede ser incluida en el README de cada repositorio utilizado. Este documento deberá incluir:

- **Back-end:**
  - Instrucciones para levantar la base de datos, indicando comandos y programas a utilizar, junto con una breve descripción.

- Instrucciones para instalar dependencias de la API, indicando comandos a ejecutar y una breve descripción de lo que hacen.
- Documentación de la API, indicando (como mínimo) para cada *endpoint*: 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.

#### ■ Front-end:

- **Nuevo!**. Instrucciones para construir entorno de desarrollo de front-end, indicando comandos a ejecutar y una breve descripción de lo que hacen.
- **Nuevo!**. Instrucciones para levantar la interfaz de la aplicación y conectarse a la API (back-end).

Además de esto, la aplicación deberá ser publicada y quedar disponible para su uso público. Para esto, se recomienda hacer *deployment* en alguna plataforma como [Netlify](#) o [Render](#).

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.
- **Nuevo!**. Implementación del front-end del juego, desarrollado utilizando React.
- Documentación para levantar la aplicación y montar la base de datos (back-end).
- **Nuevo!**. Documentación para levantar la interfaz de la aplicación y conectarse a la API (front-end).
- 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 C:** En back-end, se deberá completar la implementación de la lógica de juego. En front-end, deberá incluir la implementación de tablero básico funcional. Para que el tablero sea funcional, se deberá conectar la aplicación a los *endpoints* implementados en la entrega anterior.
- **Banda B:** En back-end deberá incluir manejo de sesión y permisos/autorización por medio de JWT (JSON Web Tokens), utilizando [koajs/jwt](#) y se deberá utilizar [bcrypt](#) para encriptar las contraseñas de los usuarios. Además, deberá hacer deploy tanto del front-end como del back-end. Se recomienda el uso de render y netlify respectivamente.
- **Banda A:** Uso de JWT por medio de *headers* en *requests* que llegan al servidor (o por medio de *http-only cookies*). Para completar la banda A, deberá completar una (o más) de las siguientes opciones (acordar con el ayudante):

- Completar la implementación del tablero. Todas las funcionalidades deberán estar disponibles, y la interfaz de usuario deberá ser limpia y fácil de utilizar.
- Incluir manejo de errores en el front-end. Aquellas acciones que lleven a algún error o que no sean exitosas deberán ser notificadas a los usuarios por medio de la interfaz: se podrán usar mensajes y realizar acciones en el front-end, priorizando la usabilidad de la aplicación.
- Deberán realizar validaciones en el back-end para evitar posibles errores por valores ingresados por medio de la API.

#### 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).