



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

IIC2513 — Tecnologías y Aplicaciones Web

Proyecto: Entrega 3

Actualización: 3 de noviembre de 2023

Entrega

- **Fecha y hora:** Viernes 17 de noviembre del 2023, a las 22:00
- **Lugar:** Repositorio grupal en la organizacion del curso en GitHub

Objetivos

- **Construir** sitios con contenido dinámico.
- **Construir** aplicaciones usando las tecnologías y herramientas disponibles.
- **Integrar** técnicas de desarrollo de software para construir aplicaciones web de alta calidad.

Descripción

Para esta entrega, cada equipo de estudiantes deberá en completar lo que han comenzado en las entregas anteriores, tanto en términos de *frontend* como de *backend*. Esto implica la finalización de la principal interfaz de usuario de su aplicación web, que fue avanzada en la primera entrega. Esta no solo debe ser atractiva e sencilla de utilizar, sino que también debe ser funcional, manteniendo una adecuada conexión con el *backend*. Para ello,

- **Manejo de sesión:** se debe implementar un sistema que permita a los usuarios autenticarse de manera segura en la aplicación web. Esto incluye la creación de un proceso de inicio de sesión que valide las credenciales del usuario, genere y administre tokens de sesión y permita el cierre de sesión de manera eficiente. Para ello, deberán hacer uso de [Jason Web Tokens](#), que pueden manejar utilizando la librería de Koa [jsonwebtoken](#).

Además, se debe asegurar que las rutas y *endpoints* relevantes estén protegidos de manera adecuada para garantizar la seguridad de los datos y la privacidad de los usuarios. El sistema de manejo de sesión debe ser robusto y confiable, utilizando la librería [bcrypt](#) para hashear las contraseñas de los usuarios.

- **Realización de CRUDs:** cada grupo debe terminar la implementación de las operaciones CRUD (Crear, Leer, Actualizar y Borrar) para las entidades que aún no han sido abordadas en el *backend*. Esto implica la creación de *endpoints* que permitan a los usuarios realizar una diversidad de acciones como agregar nuevos registros, consultar y actualizar datos existentes y eliminar recursos correctamente.
- **Interfaz principal de usuario:** cada grupo debe completar la vista principal del *frontend* de la aplicación web. Esto incluye la finalización de la interfaz de usuario de la página principal de la aplicación de acuerdo con las especificaciones previamente definidas, asegurando que esté visualmente atractiva y funcional. Es sumamente esencial garantizar una comunicación efectiva con el *backend* para que los datos se muestren de manera coherente y se actualicen en tiempo real cuando sea necesario.

- **Documentación de la API:** cada grupo deberá proporcionar una documentación detallada y actualizada, indicando (como mínimo) para cada *endpoint*:
 - Método HTTP a utilizar.
 - Ruta del *endpoint*.
 - Argumentos que recibe (y su formato).
 - Lo que retorna este *endpoint* (y su formato).

Cada grupo podrá utilizar el mismo documento creado para la entrega anterior y modificarlo tal que se reflejen los cambios realizados en su aplicación.

- **Uso de Linter:** al igual que en la entrega anterior, se espera que sigan con la utilización de una herramienta de *linting* como [ESLint](#) para mantener un código limpio y coherente en todo el proyecto, mejorando así la legibilidad y la calidad del código. Este debe ser utilizado en el *backend* de su proyecto. Es importante destacar que el uso del linter debe ser efectivo y no se limita a simplemente desactivar todas las reglas que este verifica. Si tienen dudas sobre qué guías deben mantener y cuáles pueden eliminar, no duden en hablar con sus ayudantes guías.

Además de lo descrito anteriormente, cada grupo deberá seguir con el uso apropiado de *GitHub*, incluyendo ramas, PRs y commits convencionales.

Bonus

Además de los requisitos mínimos mencionados anteriormente, se les otorgará la oportunidad de mejorar aún más su trabajo. Para incentivar las prácticas de desarrollo de alta calidad, se otorgará 5 décimas de bono a aquellos estudiantes que opten por implementar pruebas unitarias en su aplicación utilizando la librería [Jest](#). Las pruebas unitarias son una herramienta esencial para garantizar la estabilidad y el funcionamiento correcto de su aplicación, y demuestran un compromiso con la excelencia en el desarrollo.

Para conseguir esta bonificación adicional, las pruebas que realicen deberán cubrir un 50% del código que han realizado. En caso de que no logren alcanzar esta meta, se dará puntaje parcial por llegar a, por lo menos, 30%.

Entregables

Como equipo, deberán entregar:

- Implementación del *backend* con los CRUDs y el manejo de sesión realizado. Debe incluir el uso de linter.
- Implementación del *frontend* de la aplicación con la vista principal finalizada, al menos 50% de los *endpoints* conectados y el manejo de sesión funcional.
- Documentación para levantar la aplicación y montar la base de datos.
- Documentación de uso de los *endpoints* implementados.

Rúbrica

A continuación, se describe el criterio de cada uno de los ítems de la rúbrica con la que se evaluará esta entrega. La nota se calcula al 50 % considerando un total de 28 puntos.

Nota: Los ítems marcados como ESENCIAL son requisitos mínimos indispensables para la entrega. No cumplirlo, reducirá la **nota máxima a 4,0**.

- **Manejo de sesión en *backend* [6 puntos] (ESENCIAL):** se espera que se implemente una forma de manejar la sesión de usuarios en el lado del servidor. Esto incluye la validación de credenciales, el uso y almacenamiento de JWT y la encriptación de contraseñas. Es de suma relevancia que tengan rutas protegidas para que la funcionalidad del manejo de sesión pueda ser evidenciada.
- **Manejo de sesión en *frontend* [5 puntos] (ESENCIAL):** se espera que se implemente una forma de manejar la sesión de usuarios en el lado del cliente. Esto implica permitir que los usuarios puedan, como mínimo, registrarse e iniciar sesión, además de utilizar el JWT obtenido al hacer *login* para acceder a rutas protegidas.
- **Operaciones CRUD [4 puntos]:** se espera que se creen todos los *endpoints* necesarios para que su aplicación funcione, utilizando las entidades de su modelo E/R que fueron programadas en *Sequelize* en la entrega anterior. No es necesario que todos los modelos tengan la opción de crear, leer, eliminar y editar, sino que pueden programar lo que estimen necesario. Sin embargo, es de suma importancia que su *backend* ya esté completo y funcional.
- **Interfaz de usuario principal [4 puntos]:** deben completar la interfaz de usuario principal de su aplicación web. Esta debe estar casi del todo funcional, tal que ya se pueda vislumbrar como será el funcionamiento de su proyecto.
- **Conexión *frontend-backend* [2 puntos]:** se espera que tengan a lo menos el 50% de los *endpoints* creados conectados con el *frontend*. Es decir, si tienen 10 *endpoints*, mínimo 5 deberán estar conectados para conseguir el total del puntaje en este ítem.
- **Documentación API [2 puntos]:** se debe proporcionar una documentación exhaustiva y de calidad para la API de la aplicación. Esta debe incluir información detallada sobre cada uno de los *endpoints* disponibles en la API, lo que incluye el método HTTP utilizado, la ruta del endpoint, los argumentos que se deben proporcionar (junto con su formato), y lo que se puede esperar como respuesta del endpoint (junto con su formato). Además, se recomienda el uso de herramientas como *Swagger* o *Postman* y la inclusión de ejemplos para ayudar a los ayudantes a comprender cómo interactuar con la API de manera efectiva.
- **Documentación instalación BDD y API [2 puntos]:** se espera que se entregue documentación para guiar a los usuarios en la instalación y configuración del backend de la aplicación. Para la base de datos, se espera que la documentación incluya información sobre cómo configurar y levantar la base de datos, así como los comandos específicos que deben ejecutarse. En el caso de la API, la documentación debe indicar cómo instalar las dependencias necesarias para ejecutarla, proporcionando comandos específicos para ello.
- **Gitflow [2 puntos]:** se espera que cada grupo siga buenas prácticas de Git y GitHub en su desarrollo, utilizando *Pull Requests* para la revisión de código y la integración de cambios; aplicando *Conventional Commits* para mensajes descriptivos y significativos; y organizando su trabajo en ramas de manera lógica y coherente.
- **Linter [1 puntos]:** se evalúa si se utiliza una herramienta de linting, como *ESLint*, para mantener un código limpio y legible.

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, significara un descuento del 50% sobre la nota obtenida por él o los individuos.
- Durante el proyecto completo, cada grupo puede utilizar (3) cupones de atraso. En caso de no usarse en esta entrega, pueden ser utilizados durante el resto del proyecto. 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).