



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

IIC2513 — Tecnologías y Aplicaciones Web

Proyecto: Entrega 3

Actualización: 11 de noviembre de 2024

Entrega

- **Fecha y hora:** Viernes 29 de noviembre del 2024, a las 22:00
- **Lugar:** Repositorio grupal en la organización del curso en GitHub, rama `main`.

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 completar lo que han comenzado en las entregas anteriores, tanto en términos de *front-end* como de *back-end*. Esto implica que, al final de esta entrega, deberán tener un producto mínimo viable completamente funcional. El *front-end* y el *back-end* de su aplicación deben estar conectados, tal que su página web se puede utilizar. En específico, para esta entrega deberán desarrollar:

- **Finalización de CRUDs:** en caso de que no se hayan completado los *endpoints* solicitados en la Entrega 2, deberán terminarlos.
- **Finalización de vistas:** en esta entrega, deberán terminar todas las vistas mínimas para el funcionamiento de su aplicación. Esto incluye la interfaz principal, la página de inicio, las vistas de registro, de inicio de sesión, de instrucciones, etc.
- **Conexión entre cliente y servidor:** deben conectar todos los *endpoints* a su *front-end*, tal que no haya nada “hardcodeado” al final de su entrega.
- **Usuario administrador:** al manejo de sesión que ya existe deben agregar un usuario administrador, utilizando **JWT** para realizar el proceso de autorización en este. Se espera que este usuario administrador pueda acceder a rutas protegidas que el resto de los usuarios no pueden utilizar.
- Deberán integrar como mínimo una de las siguientes tecnologías:
 - **Uso de WebSockets:** deberán integrar WebSockets en su aplicación para permitir la actualización en tiempo real de todos los usuarios conectados. Esta funcionalidad asegurará que cualquier cambio en la página se refleje instantáneamente para todos los usuarios sin necesidad de recargar la página.

- **Progressive Web App (PWA)** : deberán implementar una PWA que permita instalar la aplicación como una aplicación de escritorio. Para esto, deberán utilizar service workers, los cuales gestionarán el caché de los recursos esenciales de la aplicación, asegurando su disponibilidad offline.
- **IndexedDB**: Esta base de datos deberá almacenar información de la aplicación para permitir su uso sin conexión a internet, y sincronizarse periódicamente con la API desplegada.
- **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).

Cualquier cambio que hayan realizado, ya sea la haber creado *endpoints* nuevos o editado antiguos, debe verse reflejado en la documentación. En caso de que no se realice ningún cambio desde la Entrega 2, deben entregar el documento en esta.

- **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 *back-end* 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 premiará por tomar la oportunidad de mejorar aún más su trabajo. Para incentivar las prácticas de desarrollo de alta calidad, se otorgará una bonificación de 5 décimas a aquellos grupos que opten por implementar pruebas en el *front-end* de su aplicación utilizando la librería **Selenium**. Las pruebas en el *front-end* son esenciales 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, se establecerá un criterio mínimo que los alumnos deben alcanzar para ganarse el bono. Deberán cubrir al menos el flujo de registro e inicio de sesión, junto con probar una de las funcionalidades claves de su aplicación (si es un juego, realizar una jugada; si es un planificador, agregar un evento nuevo; etc). En caso de que no logren alcanzar esta meta, se dará puntaje parcial por probar de forma correcta dos de estas tres funcionalidades (registrarse, iniciar sesión, funcionalidad clave de su aplicación).

Competencia

Además de cumplir con los requisitos mínimos y bonus establecidos en esta entrega, se llevarán a cabo dos competencias adicionales para reconocer el esfuerzo y la excelencia en diferentes áreas del proyecto. Los ganadores de estas serán decididos a través de una votación llevada a cabo por el equipo docente.

Mejor aplicación

En esta categoría, se evaluarán las aplicaciones en temas de su funcionalidad, usabilidad y la calidad general del proyecto. Se otorgarán los siguientes premios:

- **Primer lugar:** el grupo con la mejor aplicación ganará un premio de 5 décimas al promedio final del proyecto.
- **Segundo lugar:** el segundo mejor grupo ganará un premio de 4 décimas al promedio final del proyecto.
- **Tercer lugar:** el tercer mejor grupo ganará un premio de 3 décimas al promedio final del proyecto.

Mejor diseño de *front-end*

En esta categoría, se reconocerá a los grupos con los mejores diseños de las interfaces de usuario de su *front-end*. Se espera que, con esto, se fomente la creatividad, la innovación y la atención al detalle en el desarrollo de aplicaciones web. Se otorgarán premios a los cinco mejores grupos en esta categoría, cada uno ganando un premio de 2 décimas al promedio final del proyecto.

Entregables

Como equipo, deberán entregar:

- Implementación del *back-end* completo, con todos los *endpoints* correspondientes y el manejo de sesión con autorización y autenticación. Debe incluir el uso de *linter*.
- Implementación del *front-end* de la aplicación con las vistas listas y conectadas al *back-end*. Además de restringir al menos 1 vista para el usuario administrador.
- Documentación para levantar la aplicación y montar la base de datos en local.
- Documentación de uso de los *endpoints* implementados.
- URLs de la aplicación (tanto *front-end* como *back-end*) desplegada para uso público.
- Documento de diseño actualizado.

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 % del puntaje total.

- **Corrección de comentarios del ayudante [1 punto] [ESENCIAL]:** Es obligatorio que el equipo haya considerado y corregido los comentarios realizados por el ayudante en la entrega anterior. Se espera que las observaciones hayan sido tomada en cuenta y que los aspectos indicados hayan sido corregidos o mejorados.

- **Finalización de CRUDs [4 puntos]:** se espera que completen todos los *endpoints* necesarios para realizar las operaciones de Crear, Leer, Actualizar y Eliminar (CRUD) en la base de datos. Cada entidad principal de la aplicación debe cuenta con sus respectivos *endpoints* para garantizar la funcionalidad de su aplicación.
- **Finalización de vistas [4 puntos]:** se deben finalizar todas las vistas necesarias para el funcionamiento básico de la aplicación. Esto incluye la interfaz principal, página de inicio, registro, inicio de sesión, instrucciones, entre otras.
- **Conexión cliente-servidor [5 puntos]:** se espera que logren una conexión exitosa entre el *front-end* y el *back-end*. Todos los *endpoints* tienen que estar integrados y funcionando adecuadamente en el *front-end*, eliminando cualquier referencia “hardcodeada” y asegurando un flujo de datos dinámico.
- Deberán integrar como mínimo una de las siguientes tecnologías:
 - **Integración de WebSockets [5 puntos]:** Se espera que logren implementar el uso de WebSockets para que los cambios realizados en la página se reflejen en tiempo real, sin necesidad de que el usuario actualice manualmente la página.
 - **Progressive Web App (PWA) [5 puntos]:** Se espera que logren implementar una PWA que permita instalar la aplicación como una aplicación de escritorio. La PWA debe funcionar de manera consistente tanto en su versión web como en la versión instalada, ofreciendo una experiencia offline gracias al uso de *service workers*.
 - **IndexedDB [5 puntos]:** Se espera que logren integrar IndexedDB en el cliente para almacenar la base de datos localmente en caché. Además, la información almacenada deberá sincronizarse periódicamente con la API desplegada, garantizando que los datos estén actualizados incluso en situaciones de conexión intermitente.

El integrar más de una funcionalidad es equivalente a una bonificación a su nota final del proyecto.

- **Usuario administrador [3 puntos]:** se espera que implementen la funcionalidad de usuario administrador utilizando JWT para la autorización. El usuario administrador tiene acceso a rutas protegidas que no están disponibles para otros usuarios tanto para el *back-end* como *front-end*. Se debe contar con al menos una vista y un *endpoint* protegida para este usuario.
- **Deploy [1 punto]:** La aplicación debe estar desplegada en línea y accesible para cualquier usuario a través de un enlace específico. **No se aceptarán revisiones de proyectos que solo funcionen localmente.**
- **Documentación API [3 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.
- **Documento de diseño [1 punto]:** Es importante que mantengan su repositorio constantemente actualizado con las últimas versiones de las vistas diseñadas. Esto asegurará que el avance y la documentación del proyecto estén siempre al día.
- **Gitflow [3 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 punto]:** se espera que utilicen una herramienta de *linting*, como *ESLint*, para mantener un código limpio y legible.

Notas

- Las bonificaciones solo la podrán recibir si cumplen con los requisitos mínimos para aprobar el proyecto.
- Esta entrega es de **carácter reprobatorio**. Esto implica que, obtener una nota menor a 4.0 implica la reprobación del proyecto entero. Se les recomienda revisar la rúbrica a continuación para asegurar la cantidad de puntos mínimos para obtener la nota mínima necesaria.
- 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).
- Se descontarán 0.5 décimas si no se establece un *timeout* de 1 minuto en el *workflow* de *GitHub Actions*. Se permitirá un máximo de 20 minutos de ejecución y 15 *runs* por entrega; el incumplimiento de estos límites resultará en un descuento.