



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

IIC2513 — Tecnologías y Aplicaciones Web

Proyecto: Entrega 1

Actualización: 31 de agosto de 2023

Entrega

- **Fecha y hora:** Viernes 15 de septiembre del 2023, a las 22:00
- **Lugar:** Repositorio grupal en la organización 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, deberán desarrollar un sitio web navegable (que permita moverse entre las distintas páginas por medio de la interfaz) que mezcle contenido estático y dinámico, en el que se pueda explorar información básica de la aplicación y que incluya avances en lo que serán los futuros componentes usados en esta. Como mínimo, cada equipo deberá desarrollar:

- Una **landing page** que incluya: un logo, una frase que resuma la aplicación y una imagen que tenga relación con esta. Cada equipo tendrá libertad de agregar más cosas si creen que aporta a la página (por ejemplo, mostrar un formulario de log in/sign in que se vuelva funcional en futuras entregas).
- Una **navbar** o barra de navegación que sirva para que los usuarios puedan moverse entre páginas.
- Una **página principal**, que será donde se desarrolle la mayor parte de su aplicación. Esta puede no estar completa, pero debe estar avanzada, de forma que se pueda entender cómo funcionará la aplicación a futuro y con qué elementos. Estos **no necesariamente** deben ser funcionales aún, sino que pueden estar "hardcodeados".
- Una **página de instrucciones**, donde se especifique exactamente cómo funcionará la aplicación y cuál es el objetivo de cada una de sus funcionalidades.

Dado que para esta entrega (y para el proyecto en general) será necesario utilizar React, se exige que incorporen a lo menos **(2) elementos dinámicos** en (2) vistas distintas. Por ejemplo, un componente dinámico en las instrucciones y otra en la página principal. Algunos ejemplos de este tipo de página son:

- **4 colors**: en esta página, previo a empezar el juego, se muestra un componente que dinámicamente va cambiando para mostrar las reglas. Esto es un ejemplo de lo que podrían hacer para mostrar las instrucciones de su juego o los elementos de este.

- **Mubasic:** en esta página, llegando a la landing page, se muestran componentes que simulan teclas de piano, con las que el usuario puede interactuar. Este es un ejemplo de como se puede hacer una página de inicio dinámica.

Por último, para esta entrega, será necesario que apliquen las buenas prácticas de *Github*, que incluyen el uso de *Pull Requests*, *conventional commits* y *branches*. En el repositorio del curso podrán encontrar una cápsula con una explicación detallada de lo que se espera en este punto.

Entregables

Para esta entrega, deberán entregar la implementación de la aplicación web en React, con todas las vistas solicitadas.

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 16 puntos.

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

- **Uso de React [1 puntos] [ESENCIAL]:** debe haber un uso de la librería ReactJS a lo largo de la entrega, utilizando Vite para crear el proyecto.
- **Landing page [2 puntos]:** esta vista debe ser atractiva y aportar una introducción clara a la aplicación. Debe incluir un logo representativo, una frase que resuma la aplicación de manera concisa y una imagen relevante. Pueden agregar elementos adicionales que enriquezcan la página, como un formulario de inicio de sesión o registro, mostrando su comprensión de componentes y estilos en CSS.
- **Navbar [2 puntos]:** la barra de navegación debe permitir a los usuarios moverse de manera intuitiva entre las distintas páginas del sitio web. Debe ser visible en todas las páginas y reflejar el diseño coherente de la aplicación.
- **Página principal [3 puntos]:** debe proporcionar una visión general de cómo funcionará la aplicación en el futuro, incluyendo los elementos principales que se utilizarán. Aunque estos elementos pueden estar "hard-codeados", se espera que muestren una comprensión sólida de cómo los componentes interactuarán y se presentarán al usuario.
- **Página de instrucciones [2 puntos]:** esta vista debe proporcionar una guía detallada sobre cómo funcionará la aplicación web y cuál es el propósito de cada una de sus *features*. Debe ser clara y fácil de entender para los usuarios.
- **Componentes dinámicos [3 puntos] [ESENCIAL]:** se incorporan al menos dos elementos dinámicos en dos de las vistas que programen. Estos componentes deben demostrar la capacidad de manejar cambios de estado y actualizaciones en tiempo real. Los elementos dinámicos podrían incluir áreas que muestran información actualizada, interacciones de usuario u otras funcionalidades que requieran cambios en la interfaz, tal como se muestra en los ejemplos dados.
- **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.

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).
- Para esta entrega, queda **expresamente prohibido** el uso de bibliotecas de componentes HTML/CSS/JS como Bootstrap o Zurb Foundation y de preprocesadores de CSS como EJS, SASS/SCSS, Less y Stylus.