



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

IIC2513 — Tecnologías y Aplicaciones Web

Proyecto: Entrega 1

Actualización: 7 de mayo de 2024

Entrega

- **Fecha y hora:** Viernes 31 de mayo 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, deberán diseñar la interfaz de una aplicación web. Esta aplicación deberá actuar como un sitio web navegable (es decir, que permita moverse entre las distintas páginas por medio de la UI), que incluya contenido estático y dinámico, en la que se pueda explorar información básica de la aplicación y que incluya avances de lo que serán los futuros componentes usados en esta.

Se entiende como contenido dinámico al contenido generado en base a la interacción del usuario, lo que permite generar o modificar partes de la página desplegada en base a sus acciones o sesiones. Por otra parte, el contenido estático es aquel en el que las páginas disponibles permanecen constantes y no cambian. **Notar que el uso de efectos CSS no se considera contenido dinámico.**

Para esta entrega, cada equipo deberá desarrollar como mínimo:

- **Página de inicio (*Landing page*):** Esta página será la carta de presentación de su aplicación y deberá incluir el logo, una imagen representativa del objetivo de la página y una frase que resuma lo que buscan implementar. Se espera que el diseño de la página de inicio sea no solo funcional sino también estéticamente atractivo y moderno, reflejando las últimas tendencias en diseño web. Se otorga libertad a cada equipo para añadir elementos adicionales que consideren relevantes para la experiencia del usuario, como por ejemplo un formulario de *log in* o *sign in* estático, que se desarrollará para funcionar en entregas futuras. Se valorará especialmente la originalidad y la innovación en el diseño, evitando soluciones de diseño triviales o genéricas.
- **Página del equipo (*About Us*):** Esta página deberá presentar a los miembros del equipo. Cada miembro deberá ser representado con una imagen y una breve descripción personal. Se espera que implementen componentes animados con CSS para enriquecer la presentación, tales como tarjetas animadas de presentación. Pueden inspirarse en ejemplos disponibles en [Uiverse.io](https://uiverse.io). **Recuerden citar en el *README.md* los elementos que utilicen.**

- **Página principal (*Main page*):** Esta página representará el objetivo de su aplicación y deberá mostrar un avance significativo que permita visualizar cómo funcionará la aplicación en el futuro. Aunque no es necesario que todos los elementos estén completamente funcionales en esta etapa, deben estar suficientemente desarrollados para ofrecer una clara comprensión de sus funciones. Es aceptable que algunos componentes estén “hardcodeados”, siempre que ayuden a ilustrar el propósito y la mecánica de la aplicación. Se espera que esta página cuente con al menos un componente dinámico.
- **Página de instrucciones (*Docs Page*):** Esta página será necesaria para orientar a los usuarios sobre cómo navegar y utilizar la aplicación. Deberá incluir explicaciones sobre el funcionamiento y los objetivos de cada funcionalidad de la aplicación. Cada sección de la página debe clarificar el propósito de los componentes implementados y proporcionar guías de cómo interactuar con ellos.
- **Barra de navegación (*Navbar*):** Esta componente esencial deberá facilitar la movilidad entre las diversas secciones de la página mediante botones o enlaces de redirección claramente identificables. La barra de navegación debe estar diseñada para conectar todas las vistas de la aplicación de manera intuitiva, asegurando una experiencia de usuario fluida y eficiente.
- **Archivo GitHub Actions:** Implementar un flujo de trabajo de despliegue automático utilizando GitHub Actions. Cada vez que se realice un *push* a la rama principal, el sistema debe actualizar la aplicación automáticamente en la plataforma de alojamiento. Esto facilitará la gestión continua del proyecto y garantizará que los cambios más recientes estén siempre disponibles en línea.

Dado que para esta entrega (y para el proyecto en general) será necesario utilizar React, se exige que incorporen como mínimo **4 elementos dinámicos** en el menos **2 vistas distintas** y **2 elementos con efectos CSS en todas las vista**. Por ejemplo, un componente dinámico en las instrucciones y otra en la página principal. Algunos ejemplos de este tipo de página son:

- **Simulador MacOS:** Esta página utiliza efectos CSS, destacando el efecto *hover* que aplica un zoom a los elementos cuando el usuario pasa el cursor sobre ellos, mejorando así la interactividad visual.
- **Contador:** En este sitio, se emplea el hook *useState* de React para crear un contador dinámico que actualiza su valor en respuesta a las interacciones del usuario, demostrando un uso efectivo de los componentes dinámicos en React.
- **React day and night:** Esta página implementa hooks y animaciones CSS. Utiliza el hook *useState* para gestionar el estado del *DarkMode*, y una animación de rotación en el logo de React para mejorar la experiencia visual del usuario.
- **4 colors:** Antes de iniciar el juego, este sitio muestra un componente dinámico que cambia para presentar las reglas del juego. Este enfoque podría ser aplicado para mostrar instrucciones o elementos de un juego de manera interactiva.
- **Mubasic:** Al llegar a la landing page de este sitio, se presentan componentes que simulan teclas de piano y permiten la interacción del usuario. Este ejemplo muestra cómo crear una página de inicio dinámica e interactiva.

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 presentar la implementación completa de la aplicación web desarrollada en React, que incluya todas las vistas solicitadas. Es esencial que los documentos de diseño estén actualizados de acuerdo con el estilo de las vistas implementadas. Además, un requisito **esencial** es que la aplicación debe estar desplegada en línea, accesible a través de un enlace específico y disponible para cualquier usuario; no se aceptarán entregas de proyectos que solo funcionen en un entorno local.

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.

Nota: Los ítems marcados como **ESENCIAL** son requisitos mínimos indispensables para la entrega. El incumplimiento de estos requisitos limitará la **nota máxima posible a 4,0**. Es decir que, bajo esta condición, un 4,0 se convierte en el puntaje máximo posible y la escala de calificación se ajustará en base a esto. Por ende, si la nota obtenida inicialmente supera el 4,0, esta se ajustará proporcionalmente de acuerdo con la nueva escala máxima establecida.

- **Documento de diseño [1 punto] [ESENCIAL]:** 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.
- **Deploy [1 punto] [ESENCIAL]:** 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.**
- **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.
- **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. **Los componentes creados deben estar en una carpeta destinada únicamente a componentes.**
- **Diseño [8 puntos]:** Cada vista de la aplicación debe incorporar al menos dos elementos con efectos CSS. Además, se deben cumplir los siguientes criterios para asegurar la coherencia y eficacia del diseño:
 - Los colores utilizados en la página web deben alinearse con los definidos en el documento de diseño.
 - Se deben respetar los principios de diseño UI/UX, incluyendo la jerarquía visual, la alineación y el diseño responsivo.
 - Los elementos CSS deben ser empleados de manera interactiva, agregando efectos visuales que mejoren la experiencia del usuario.
 - La navegación por la página debe ser fácil e intuitiva, facilitando el acceso y la interacción con el contenido de manera efectiva.
- **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. Además deberá tener el direccionamiento a todas las paginas diseñandas. **No se permitirá llegar a una pagina modificando manualmente el link de la url.**

- **Uso de GitHub Actions [3 puntos]:** Deben implementar un archivo de GitHub Actions configurado para realizar el despliegue automático de la aplicación. Cada vez que se realice un *push* a la rama principal (*main*), el sistema debe actualizar automáticamente la aplicación en el servidor de alojamiento.
- **Landing page [1.5 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.
- **Main page [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. **Debe contar con al menos un componente dinámico.**
- **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. **Debe contar con al menos un componente dinámico.**
- **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, significará 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).