



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

IIC2513 — Tecnologías y Aplicaciones Web

Proyecto Semestral - Sección 1

Actualización: 1 de octubre de 2024

1. Objetivos

- **Demostrar la comprensión** de conceptos de tecnologías web, como HTML, CSS, JavaScript y React, para desarrollar una aplicación web funcional y atractiva.
- **Trabajar de manera conjunta** en la planificación, diseño y codificación de una aplicación web, dividiendo tareas y responsabilidades de manera equitativa.

2. Introducción

El proyecto a desarrollar a lo largo del curso tiene como objetivo que los estudiantes, en un contexto de trabajo en equipo, enfrenten un desafío de diseño, programación, implementación y despliegue de una aplicación web, que les permita poner en práctica los contenidos del curso vistos durante el semestre.

Las entregas del proyecto están vinculadas entre sí, y el resultado final del proyecto será una aplicación totalmente funcional con un lado cliente y otro servidor. Además, todo el proyecto deberá estar respaldado por la documentación correspondiente en cuanto a diseño de la aplicación y el setup de cada una de las partes del proyecto.

Para lograr lo anterior, deberán cumplir con objetivos intermedios, que corresponderán a las “entregas parciales de proyecto”, de manera que el desarrollo sea incremental y guiado tanto por los contenidos como por el equipo de ayudantes.

Salvo excepciones, condiciones y restricciones, que se señalarán oportunamente en cada enunciado de entrega, cada equipo tendrá total libertad en cuanto al diseño de UX, UI y la implementación del proyecto.

3. Metodología de trabajo

Para el desarrollo del proyecto, los estudiantes deberán formar un grupo de 3 personas que se mantendrá para todas las entregas. Para inscribir los grupos se creará un formulario en *Google Forms* donde podrán registrarlos.

Importante: Quienes no tengan grupo al final del periodo de inscripción, serán asignados de manera aleatoria con otros estudiantes en similar situación.

4. Descripción general

4.1. Funcionalidades mínimas

El proyecto a desarrollar deberá cumplir **al menos** con las siguientes funcionalidades:

- Deberá tener registro y autenticación de usuarios (con *login* y *password*).
- Deberá usar una base de datos para almacenar toda información relevante de manera persistente.

- El lado del cliente (*front-end*) deberá tener una interfaz gráfica atractiva, completa e intuitiva que permita realizar múltiples acciones (tales como seleccionar, hacer *drag-and-drop*, u otras que el diseño permita). No tiene que soportar todas las acciones, pero sí se espera que sea una interfaz no-trivial.
- Se deben manejar diversos perfiles de usuarios y acciones para que cada perfil de usuario pueda realizar acciones acorde a su perfil. Por ejemplo, perfiles de jugador, administrador, "superusuario", "moderador", etc.
- El modelo E/R debe incluir una entidad específica que represente los estados de una acción, los cuales deben ser gestionados mediante solicitudes (*requests*) al *back-end*. Por ejemplo, esta entidad podría manejar los estados de un mensaje, tales como *enviado*, *recibido*, y *leído*; o los estados de una compra, como *en carrito*, *en envío*, y *despachado*.
- Debe tener páginas dinámicas, adicionales a las estáticas, que varíen su contenido dependiendo de las acciones que el usuario realice en tiempo real.
- Debe tener un panel que permita ver el resumen de acciones hechas por un usuario (un análogo sería una aplicación de banco que muestre las transacciones realizadas en un período fijo de tiempo).
- Debe manejar la información de manera segura (por medio de encriptación) cuando corresponda.
- La implementación y desarrollo debe proveer un set de pruebas automatizados.
- Deberá implementar al menos una de las siguientes tecnologías para mejorar la experiencia de usuario y la funcionalidad del proyecto:
 - **Progressive Web App (PWA)**: El proyecto deberá ser accesible como una aplicación web progresiva, permitiendo que los usuarios puedan instalar la aplicación en sus dispositivos y acceder a ella sin conexión a internet, utilizando técnicas como el uso de *service workers*.
 - **IndexedDB**: Se deberá implementar el uso de IndexedDB en el cliente para almacenar datos localmente, permitiendo el acceso y sincronización de la información con el servidor cuando la conexión a internet esté disponible.
 - **Websockets**: El proyecto deberá soportar comunicaciones bidireccionales en tiempo real entre el cliente y el servidor mediante el uso de Websockets, permitiendo la actualización dinámica de los datos sin necesidad de recargar la página.

4.2. Temáticas

La propuesta de aplicación a desarrollar por cada grupo podrá ser elegida entre las siguientes opciones:

- **Chat en tiempo real**: este proyecto involucra el diseño y desarrollo de un sistema de mensajería instantánea que permita a los usuarios comunicarse en tiempo real. Este proyecto deberá incorporar tanto aspectos técnicos de la comunicación en tiempo real como una interfaz de usuario intuitiva y atractiva. Se espera:
 - Diseño de una interfaz de usuario limpia y amigable
 - Implementación de tecnologías de comunicación en tiempo real
 - El sistema debe permitir la comunicación simultánea entre varios usuarios, incluyendo la creación de salas de chat o conversaciones grupales.
 - Incorporar características como notificaciones de mensajes nuevos, la posibilidad de enviar emojis o archivos, y opciones de personalización de la interfaz.

- **E-commerce:** en esta opción, tendrán la oportunidad de diseñar y desarrollar una plataforma de comercio electrónico. Este proyecto debe permitir a los usuarios buscar, seleccionar y comprar productos en línea de manera eficiente y segura. Se espera:
 - Crear una interfaz que facilite la navegación, la búsqueda y la selección de productos, además de simplificar el proceso de compra.
 - Desarrollar un sistema que permita a los usuarios añadir productos a un carrito de compras, modificar cantidades y eliminar productos antes de realizar la compra.
 - Permitir que los usuarios se registren, inicien sesión y mantengan perfiles donde puedan seguir sus órdenes, gestionar métodos de pago y direcciones de envío.
 - Implementar herramientas administrativas para añadir, modificar o eliminar productos, así como para gestionar pedidos y disponibilidad de stock.
- **Juego de tablero:** podrán diseñar y desarrollar su propio juego de tablero en línea, inventando algo completamente nuevo o adaptando un juego clásico existente. Se espera:
 - Crear una interfaz de tablero que sea visualmente atractiva e interactiva.
 - Formular un conjunto de reglas que sean fáciles de entender pero que al mismo tiempo ofrezcan profundidad estratégica.
 - Utilizar gráficos y animaciones para mejorar la experiencia visual del juego y hacerla más dinámica.
 - Implementar capacidades de juego en línea que permitan a múltiples jugadores jugar desde diferentes ubicaciones de manera simultánea.
- **Planificador:** deberán crear un planificador en línea que tenga como función ayudar a sus usuarios a gestionar tareas, eventos, clases, metas etc. La idea es que sea una plataforma intuitiva que permita a los usuarios planificar lo que sea que quieran planificar (dependiendo del tema en el que esté basada su aplicación) de manera efectiva. Se espera:
 - Creación de un diseño limpio y funcional para la gestión de tareas y eventos.
 - Incorporación de funciones de recordatorio y notificación.
 - Posibilidad de establecer metas y realizar un seguimiento del progreso.
 - Personalización de la interfaz de usuario tal que esta pueda adaptarse a las preferencias de estos.
- **Red social:** esta última opción se basa en diseñar y desarrollar una red social con un enfoque temático único. Ya sea que elijas un tema artístico, deportivo, educativo o de otro tipo. Pueden basarse en redes sociales existentes pero la idea es que usen su creatividad para agregarle algo nuevo. Se espera:
 - Diseño de perfiles de usuario personalizables y atractivos.
 - Implementación de un sistema de publicación y comentarios.
 - Integración de características únicas relacionadas con el tema elegido.
 - Implementación de un chat donde los usuarios puedan comunicarse.

Adicionalmente, **se podrán proponer proyectos fuera de las temáticas listadas** siempre que sea conversado y justificado con el ayudante a cargo del grupo. Cualquier proyecto propuesto deberá incluir un conjunto de funcionalidades cuya implementación sea de una dificultad similar a la de los proyectos listados.

4.3. Entregas

El proyecto se desarrollará de manera incremental, dividido en las siguientes etapas:

- E0: Esta entrega está enfocada en la planificación y organización de la aplicación a desarrollar. Incluirá la creación de mockups para las principales vistas de la página, junto con la modelación de la base de datos. También deberán entregar una primera versión de las vistas con el uso de React.
- E1: Esta entrega se centra en el desarrollo visual de la aplicación web. Deberán diseñar e implementar las vistas y componentes a través de código adaptativo para diferentes tamaños de dispositivos. Además, la aplicación deberá estar desplegada en algún proveedor de alojamiento.
- E2: Esta entrega se enfoca en el desarrollo lógico de la aplicación web. Se espera que los estudiantes implementen correctamente las bases de datos, creen un sistema de autenticación y realicen las conexiones entre el *back-end* y el *front-end* para manejar datos dinámicos. La aplicación también deberá estar desplegada en algún proveedor de alojamiento.
- E3: En esta última entrega, se espera que la aplicación sea completamente funcional y que esté totalmente integrada, con una conexión completa entre el *back-end* y el *front-end*. Además de poder integrar el uso de tecnologías avanzadas a su aplicación.

5. Indicaciones generales

- Para alojar sus aplicaciones en Internet, se han creado cápsulas para las plataformas:
 - **Render**: Recomendado para el *back-end*.
 - **Netlify**: Recomendado para el *front-end*.

Si desean utilizar otro proveedor al recomendado pueden hacerlo, pero deberán justificar y documentar su elección.

- El código deberá desarrollarse, alojarse y entregarse en GitHub, por lo que a cada grupo se le asignarán repositorios según sea necesario. Es importante que todos los integrantes de cada grupo utilicen el repositorio y GitHub de manera colaborativa. **Se revisará el uso de GitHub para evaluar que todo miembro haya trabajado.**
- El resultado final esperado de cada entrega (a excepción de la E_0) es que se logre realizar el despliegue de la aplicación web (en modo producción) en un servidor externo, de manera que esta quede disponible para ser utilizada por cualquier usuario de Internet. **No se revisarán trabajos de forma local** (a excepción de la E_0).
- Para evaluar cada entrega, todos los miembros del equipo deberán asistir a una reunión de corrección con el ayudante asignado al grupo. **Es responsabilidad del grupo el contactar y agendar la reunión con su ayudante.**
- El uso de inteligencias artificiales está permitido siempre y cuando sea correctamente referenciado su uso en el README.md.
- Es obligatorio que los repositorios del proyecto incluyan la información completa de los integrantes del equipo (nombre completo y correo UC de cada miembro), el enlace al deploy actualizado de la aplicación, y la documentación completa de la API.

6. Tecnologías y Herramientas Requeridas

- La aplicación deberá desarrollarse usando, al menos, las siguientes tecnologías: Node.js o [Next.js](#), [Koa](#), HTML, CSS, JavaScript o TypeScript y [React](#). Opcionalmente, se permite el uso de [jQuery](#) si se estima pertinente.
- Para la base de datos se deberá usar [PostgreSQL](#) y para conexión a base de datos se podrá utilizar [pg](#), [knex](#) o un ORM como [Sequelize](#).
- Se puede hacer uso de [SASS/SCSS](#), [Less](#), [Stylus](#), [EJS](#) u otros preprocesadores de CSS.
- Se puede hacer uso de lenguajes que compilen a JavaScript, tales como [TypeScript](#) o [CoffeeScript](#).
- Además, se puede hacer uso de bibliotecas de componentes HTML/CSS/JS como [Tailwind](#), [Bootstrap](#) o [ZURB Foundation](#).
- Cualquier paquete, *framework* o librería adicional que se quiera utilizar deberá ser aprobada por el equipo docente.

7. Fechas de entrega

El proyecto se divide en las siguientes entregas:

Evaluación	Publicación	Entrega	Ponderación
Entrega 0	martes 01 de octubre	viernes 11 de octubre	10 %
Entrega 1	viernes 11 de octubre	viernes 25 de octubre	25 %
Entrega 2	viernes 25 de octubre	viernes 08 de noviembre	25 %
Entrega 3	viernes 08 de noviembre	viernes 29 de noviembre	40 %

8. Evaluación y descuentos

Cada entrega de proyecto será evaluada con una nota de 1 a 7, donde la distribución de puntaje se indicará en el enunciado de cada entrega. Sin embargo, cada entrega (a excepción de la E_0) tendrá **ítems esenciales** que todos los equipos deberán implementar de manera completa. En caso de no estar realizados, el grupo podrá optar a una nota máxima 4,0.

Tal como se menciona en el programa del curso, la última entrega es de **carácter reprobatorio**. Es decir, el no obtener una calificación mayor o igual a 3.95 en dicha entrega significará que la nota de proyecto se calculará como el mínimo entre la nota obtenida y 3.90.

El espacio de evaluación con el ayudante guía debe ser gestionado por el grupo. En esta reunión se evaluará que todos los integrantes manejen lo implementado en la entrega correspondiente. Además, se les dará el *feedback* e indicaciones para la siguiente entrega.

En caso de no ponerse en contacto con el ayudante o no asistir (sin una justificación adecuada), el grupo o integrante tendrá un descuento del 50 % sobre el puntaje obtenido en la entrega.

La falta de enlaces de deploy actualizados y la ausencia de información completa y actualizada en el repositorio del proyecto resultará en una penalización del 50 % sobre la calificación obtenida en la entrega correspondiente.

9. Evaluación de pares

Al final de cada entrega, se abrirá un formulario de evaluación de pares, donde los integrantes del grupo tendrán que evaluar a sus compañeros según los siguientes criterios:

- **Comunicación:** Busca evaluar la eficacia con la que el miembro se comunica con el resto del equipo. Esto incluye la claridad en la transmisión de ideas, la disponibilidad para discutir problemas y la contribución en reuniones. Se valorará la capacidad de escuchar a los demás y responder de manera constructiva.
- **Calidad del código / trabajo:** Este criterio evalúa la precisión técnica, la limpieza y la organización del código desarrollado por el miembro del equipo. También considera si el trabajo cumple con los estándares de desarrollo establecidos y si es eficiente en términos de funcionalidad y rendimiento.
- **Puntualidad:** Evalúa la consistencia del miembro del equipo en cumplir con los plazos acordados para la entrega de tareas y contribuciones. Este criterio busca determinar la fiabilidad del integrante del equipo.
- **Contribución / Participación:** Este criterio mide el grado de pro actividad del miembro en el proyecto. Incluye la frecuencia y el valor de sus aportaciones al trabajo del grupo, su participación activa en el desarrollo del proyecto y su disposición para asumir tareas.
- **Cumplimiento:** Este punto evalúa cómo el miembro del equipo sigue las directrices y requisitos del proyecto. Incluye la adherencia a las normativas del proyecto y la consistencia en seguir los procesos establecidos para el desarrollo y la documentación.

Dentro de cada ítem, se deberá evaluar según el grado de cumplimiento, siendo estos:

- Malo (1 punto)
- Mediano (2 puntos)
- Decente (3 puntos)
- Bueno (4 puntos)
- Excelente (5 puntos)
- Destacable (6 puntos) (*)
- Sobresaliente (7 puntos) (*)

Para que se puedan asignar los puntos marcados con (*)—Destacable (6 puntos) y Sobresaliente (7 puntos)—es necesario que un integrante haya contribuido más que los demás al trabajo del equipo. Esto implica que, mientras un miembro del grupo se destaca por su esfuerzo, otros miembros debieron haber contribuido en menor medida.

Cada integrante deberá repartir:

$$\text{puntos totales} = n \cdot p \cdot i$$

donde:

- n representa la cantidad de participantes.
- p es el máximo de puntos que cada participante puede obtener por grado de cumplimiento (5).
- i es el número de criterios que cada participante debe evaluar (5).

Considere el siguiente método de asignación de puntos en un grupo de tres personas: si un integrante trabaja más que los otros, puede recibir hasta 2 puntos adicionales en cada criterio de evaluación. Por ejemplo, si un integrante cumple con los plazos de entrega puntualmente y sus compañeros no, puede recibir hasta 7 puntos en ese criterio. Sin embargo, cada integrante no puede recibir más de 4 puntos de los otros dos, respetando el límite máximo de 15 puntos totales por grupo por criterio.

En situaciones donde todos los miembros del equipo trabajen por igual, cada uno debe recibir 5 puntos por criterio.

Para calcular el puntaje final de participación de cada integrante, se tomará el promedio de los puntos obtenidos en cada criterio. La suma de estos promedios constituirá la base para determinar la nota final de cada miembro del equipo.

La nota luego se calcula como:

$$NE = 1 + ((NE_{\text{grupal}} - 1) \cdot F_g)$$

Donde F_g es un factor calculado a partir de las evaluaciones de los compañeros. Puede moverse entre 0,2 y 1,4. **No responder esta coevaluación, quiere decir que le asigna el factor normal (1.0x) a todos los integrantes del equipo.**

Si no se cumple la suma de puntos, no se considerará la evaluación como válida y se restará 5 décimas de tu nota personal.

10. Política para entregas atrasadas

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).

Si un grupo no utiliza sus cupones durante todo el semestre, se le bonificara con 2 décimas por cada cupón no usado, sobre la nota final del proyecto. Esto solo si está aprobando el curso.

11. Recomendaciones

- Diseñar bien la lógica de cada parte de la aplicación, el entorno, los estados, la información requerida y su modelo de almacenamiento.
- Planificar el trabajo para que permita y facilite la colaboración entre integrantes del equipo.
- Preguntar y consultar, usando los canales dispuestos por el equipo docente, colaborando entre pares (sin copiar) y apoyándose en los ayudantes.
- Trabajar con tiempo, planificando cómo abordar una entrega en cuanto se publica el enunciado.
- Comenzar con una interfaz de usuario (UI) simple y minimalista, e ir evolucionando el diseño a medida que se avanza en la implementación.
- Es importante conocer los detalles de diseño y decisiones tomadas en la construcción de la aplicación. El conocimiento en detalle de lo que se está realizando es parte de los aspectos que serán evaluados.