



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

IIC2513 — Tecnologías y Aplicaciones Web

# Proyecto Semestral

## 1. Introducción

El proyecto, a desarrollar a lo largo del curso, tiene como objetivo que los estudiantes enfrenten un desafío de diseño, programación, implementación y despliegue en ambiente web, que les permita poner en práctica los conocimientos adquiridos durante este semestre.

Las entregas del proyecto están vinculadas entre sí y la última de las entregas tendrá como resultado una aplicación totalmente funcional con un lado cliente y otro servidor (¡en la nube!).

Para lograr lo anterior, usted deberá cumplir con objetivos intermedios, mejor conocidos como “entregas parciales de proyecto”, de manera de llevar un diseño inicial a un “producto” totalmente funcional. Lo anterior revela la importancia de no atrasarse ni saltarse una entrega y procurar mantenerse al día.

Los enunciados que guiarán cada entrega parcial darán la línea general de la funcionalidad que deberá implementar, pero sin entrar en mayores detalles ni puntos específicos de tal manera que usted, junto a su grupo, demuestren su capacidad de trabajo en equipo y resolución de problemas.

Salvo excepciones, condiciones y restricciones, que se señalarán oportunamente en cada enunciado de entrega, usted tendrá total libertad en cuanto a la interfaz y la implementación de su trabajo.

## 2. Metodología de Trabajo

Para el desarrollo del proyecto, deberá formar un grupo (que se mantendrá para todas las entregas) de sólo 2 personas. Para inscribir los grupos se creará un formulario en Google Forms donde podrán registrarlos.

Quienes no tengan grupo, al final del periodo de inscripción, serán asignados aleatoriamente a quienes no tengan grupo. Éstos grupos serán de 2 personas.

## 3. Marco General de las entregas

El resumen del proyecto en una línea: Deberán construir una aplicación Web. La aplicación web a construir podrá ser un juego por turnos con al menos tres participantes.

## 4. Reglas y Características Generales del juego

El juego a realizar será propuesto por cada grupo. Las propuestas serán revisadas por el equipo docente y se acordará entre ambas partes (grupo y ayudante) sus características finales. Esto no será evaluado, sin embargo, se debe acordar una reunión antes de la publicación de la Entrega 1 (viernes 14 de abril) con el ayudante guía para discutir la propuesta. **En caso de no concretarse esta reunión, el grupo no podrá acceder a los cupones de atraso.**

El juego, en la entrega final al menos:

- Deberá tener registro y autenticación de usuarios (login y password)

- Deberá usar una base de datos para almacenar información relevante
- Deberá ser un juego basado en turnos, es decir, una persona realiza todas sus jugadas y luego esperará a que el otro jugador la realice (por ejemplo como en el juego Ataque). Es posible también que todos los jugadores hagan sus jugadas y las entreguen al servidor el cual, una vez que reciba TODAS las jugadas (o se cumpla un tiempo de envío), para que el servidor resuelva combinando simultáneamente todas las jugadas.
- El lado cliente debe tener una interfaz gráfica atractiva, completa y que permita realizar múltiples acciones (como seleccionar, drag and drop, u otras que el diseño permita. No tiene que tener todas las acciones, pero si se espera que sea una interfaz no-trivial)
- Se deben manejar diversos perfiles de usuarios y acciones para que estos puedan realizar acciones acorde a su perfil. Por ejemplo, perfil jugador, perfil administrador, perfil superusuario, perfil “moderador” de juego, etc.
- Debe tener un proceso (estados) que se manejen por base de datos. Por ejemplo, estado de los turnos, turnos pasados por jugador, indicar qué jugador posee el turno actual, etc.
- Debe tener páginas dinámicas, adicionales a las estáticas, que varíe su contenido dependiendo de las acciones que el usuario realice
- Debe tener un panel que permita ver resumen de acciones hechas por usuario (un análogo sería una aplicación de banco que muestre transacciones realizadas en un periodo fijo de tiempo)
- Debe manejar seguridad (encriptación) de la información, cuando corresponda
- La implementación y desarrollo debe proveer un set de pruebas automatizados
- Debe tener un panel de administración que permita auditar las acciones realizadas por los distintos usuarios y otras acciones acordes a este perfil, como por ejemplo dar de baja a usuarios, suprimir juegos u otras

## 5. Indicaciones Generales

- Para alojar sus aplicaciones en Internet, Se recomendarán plataformas a su debido tiempo. Si desean utilizar otro proveedor al recomendado, pueden hacerlo, pero deberán justificar y documentar su elección. Noten que las cápsulas, guías y apoyo de los ayudantes será para las plataformas que se señalarán.
- El código deberá entregarse en GitHub. Les indicaremos, una vez oficializados los grupos, cuáles son los repositorios asignados por grupo. Es importante que todos los integrantes utilicen el GitHub de manera colaborativa. SE REVISARÁ EL USO DE GITHUB PARA EVALUAR SI ALGUIEN TRABAJÓ O NO.
- El resultado final esperado es que ustedes realicen el despliegue real de su aplicación web en modo producción en la nube y que ésta quede disponible para ser utilizada por cualquier usuario de Internet.
- Para evaluar cada entrega, todos los miembros del equipo tendrán una reunión de corrección con el ayudante que se les haya asignado. Hay que aclarar que es RESPONSABILIDAD DEL GRUPO el contactar y agendar la reunión con el ayudante.
- El avance logrado en cada entrega se evaluará en base a lo mostrado, cuando corresponda, en el repositorio Github en la fecha límite de la entrega. Para entregas más avanzadas se mirará lo entregado en las plataformas respectivas y se harán pruebas funcionales, es decir evaluación si la aplicación hace o no lo que debiese estar haciendo. Recordar que Github se utilizará como una medida de trabajo de cada integrante del grupo. Si el ayudante detecta que uno o más integrantes del grupo tienen baja interacción en su repositorio de Github, puede descontar a esos alumnos un porcentaje de la nota del grupo, todo esto se detalla más adelante.
- En el caso de que un grupo no entregue nada, la nota será de un 1.0.

## 6. Condiciones y Restricciones

- La aplicación debe desarrollarse usando al menos: Node.js, Koa, HTML, CSS, Javascript y React.
- Para la construcción de la aplicación pueden usar JQuery si lo estiman pertinente.
- Para base de datos deberán usar Postgres y para conexión a la base de datos pueden utilizar pg, knex o un ORM como sequelize
- Pueden usar también EJS, SASS/SCSS, less, stylus u otros preprocesadores de CSS
- No está permitido el uso de lenguajes que compilen a JavaScript, como CoffeeScript o TypeScript. Tampoco podrán usar bibliotecas de componentes HTML/CSS/JS como Bootstrap o Zurb Foundation.
- Cualquier package, framework o biblioteca adicional que quieran usar debe ser aprobado por el equipo docente.

## 7. Entregables

El proyecto se divide en 3 entregas

Evaluación	Publicación	Entrega	Ponderación
Entrega 1	viernes 14 de abril	viernes 21 de abril	25 %
Entrega 2	viernes 19 de mayo	viernes 2 de junio	35 %
Entrega 3	viernes 9 de junio	viernes 30 de junio	40 %

## 8. Evaluación y Descuentos

Cada entrega de proyecto será evaluada con una nota de 1 a 7 siguiendo una escala discreta que se detalla a continuación:

Calificación	Nota asociada	Criterios por banda
A	7,0	La entrega excede con creces lo solicitado
B+	6,0	La entrega cumple algo más allá de lo solicitado
B	5,5	La entrega cumple con lo mínimo solicitado en el enunciado
C	4,0	La entrega cumple parcialmente con lo mínimo solicitado
D	2,5	La entrega está muy por debajo del mínimo solicitado
E	1,0	No se realizó la entrega en los plazos o en la forma

**Cada banda de calificación tiene un criterio asociado, pero para alcanzarlo significa que se cumple con los criterios exigidos para el criterio anterior.** Así, para tener la opción a la banda de la A, tendrá que cumplir con todos los criterios dados por la banda B+ y recursivamente hacia abajo.

El espacio de evaluación con el ayudante guía debe ser gestionado por el grupo. En esta reunión se evaluará que ambos integrantes tengan manejo de 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 individuo tendrá un descuento del 50 % sobre el puntaje obtenido en la entrega.

Adicionalmente, el equipo docente verificará la utilización de GitHub y la actividad de cada integrante del grupo en el repositorio. Si se detecta una diferencia significativa entre los integrantes del grupo tanto en el espacio de evaluación de la reunión como en el repositorio, el integrante que haya trabajado notoriamente menos, será notificado.

Si en la siguiente entrega se repite la situación, la nota de dicho integrante caerá de banda. Es decir, si en la Entrega 1 se le notifica que debe involucrarse más en el desarrollo y en la Entrega 2 no lo hace y el grupo obtiene un 5,5 (banda B), su nota bajará a 4,0 (banda C).

Este sistema de notificación solo será valido para la Entrega 1 y 2. Para la Entrega 3, el descuento se aplica para esa misma entrega.

Los descuentos son acumulativos. Esto es, si una persona no asiste a la reunión con el ayudante y además en 2 entregas presenta una participación significativamente menor, primero se le baja de banda por su contribución y luego se aplica el 50 % asociado a la inasistencia.

## 9. Atraso

Para cada entrega se entregará un cupón de descuento. En caso de no usarlo, este se acumula para ser usado en las siguientes entregas.

Los días de atraso incluyen los fines de semana. Es decir, si la entrega está fijada para el viernes, y el grupo quiere usar 2 cupones, deberá entregar el domingo.

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

## 10. Tipos de Juego

Se recomiendan los siguientes tipos de juego:

- **Juegos de puzzle:** Los jugadores deben resolver problemas o tareas usando lógica y estrategia. La dificultad va aumentando a medida que van resolviendo los desafíos. (ej. *RushHour*)
- **Dominio de territorio y recursos:** El objetivo típico de estos juegos es conquistar todos los territorios, las “capitales” o ciertas zonas específicas, incluso algún tótem, bandera o ícono particular. Este tipo de juego implica enfrentamientos entre los participantes
- **Gestión de organización o país u otro:** El objetivo de este juego es por lo general obtener cierta cantidad de puntos, donde el ganador es el que sobrevive o el que tiene mayor puntaje enfrentado a las dificultades aleatorias que el servidor arroja y que ustedes deben resolver con sus recursos y eventual colaboración entre participantes.
- **Aventura y descubrimiento:** El objetivo de este juego es por lo general obtener cierta cantidad de elementos de recompensa (por ejemplo tesoros, banderas, etc). antes que el resto de los jugadores. Los jugadores poseen un equipo especialista y van avanzando por un mapa “oculto” que a medida que van saltando de casilla en casilla, descubren, con sus recompensas o sus dificultades (o a veces, nada).

## 11. Postulación de otro tipo de juego

Si postulan un tipo de juego diferente, este puede ser aceptado, aceptado con reparos (que deben mejorar algún aspecto) o rechazado. por supuesto pueden postular un tipo de juego que mezcle los tipos de juego presentados en este enunciado. Lo importante del juego que postulen, es que deben tener:

- Un mapa o tablero donde jugar
- Características de las casillas del mapa o tablero (recursos, “sorpresas”)

- Personajes, unidades u otro “objeto” que requieren algún tipo de recurso
- Recursos limitados entre los participantes (o repartidos de manera dispar, por ejemplo cada jugador tiene el monopolio o exclusividad de algún elemento necesario para avanzar en el juego)
- Posibilidad de mejora de unidades, territorio, personajes, etc.
- Un servidor que dirige y comunica el resultado del turno
- Estados almacenados del juego y visualización del mismo en un tablero que permita a un jugador seguir el desarrollo del juego

## **12. Recomendaciones**

- Diseñen muy bien las reglas de negocio de su aplicación, el entorno, los estados, la información requerida y su modelo de almacenamiento.
- Planifiquen el trabajo para que les permita la colaboración entre los integrantes del equipo.
- Pregunten y consulten, usen los canales dispuestos por el cuerpo docente para hacer consultas, colaboren entre ustedes (NO COPIEN) los ayudantes están para apoyarlos.
- Trabajen con tiempo, no esperen a último momento para despejar dudas.
- Comiencen con una interfaz de usuario (UI) simple y minimalista; no es necesario que su aplicación se vea “bonita” desde el comienzo
- Se recomienda (no es obligatorio) realizar prototipos no funcionales (mockups) de su aplicación. La idea es que las utilicen en las entregas siguientes así que preocúpense de diseñarlos adecuadamente. Los detalles de cada entrega se darán en su enunciado específico, este es un marco general para las entregas.
- Es importante que conozcan los detalles de diseño y decisiones tomadas en la construcción de su aplicación. El conocimiento en detalle de lo que se está realizando es parte de los aspectos que serán evaluados.
- Siempre podrán, justificadamente, cambiar alguna regla que ustedes hayan definido para su aplicación o bien mejorar algún aspecto de la misma, de la interfaz, modelo de datos, funcionalidad, etc.