



Pontificia Universidad Católica de Chile  
Escuela de ingeniería  
Departamento de ciencias de la computación  
Profesor: Hernán Cabrera

## Tarea 6

### IIC2513 Tecnologías y aplicaciones WEB

Segundo semestre 2020

**Fecha límite de entrega de tarea: miércoles 02 de diciembre 21:30 hrs.**

**Fecha límite de evaluación de pares: viernes 04 de diciembre 23:59 hrs.**  
**(<https://forms.gle/ddZCKowVqkEDJ5CH6>)**

Las tareas tienen como objetivo que enfrenten un desafío de diseño y programación que les permita practicar los conocimientos adquiridos durante este semestre.

Las tareas, si bien siguen un orden pedagógico relacionado a los contenidos del curso, están vinculadas entre sí y la última de ellas tendrá como resultado una aplicación totalmente funcional con un lado cliente y otro servidor. Es decir, ustedes deberán cumplir con objetivos intermedios de la tarea de manera de llevar un diseño e implementación 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 dan la línea general de la funcionalidad que deben implementar, pero sin entrar en mayores detalles ni puntos específicos de tal manera que ustedes demuestren su capacidad de trabajo en equipo y resolución de problemas, y **discutan sus posibilidades y criterios de evaluación con su ayudante de seguimiento.**

***Salvo excepciones, tendrán total libertad en cuanto a la interfaz, el diseño y la implementación de su trabajo.***

**Fecha límite de entrega de tarea 6, miércoles 02 de diciembre 21:30 hrs.**

## **Indicaciones**

---

El objetivo de esta entrega es **arreglar, mejorar, enmendar y dejar a punto el proyecto**. Sabemos que no siempre se puede llegar con todo terminado y que hay elementos que han faltado, ya sea por tiempo, dificultad, carga académica o infortunio. Pues esta es la oportunidad de mejorar todos los aspectos pendientes, que han sido señalados por los ayudantes, para terminar un trabajo, su trabajo, de forma integral.

En esta entrega es indispensable y obligatorio que entreguen:

1. **Aspectos mínimos de las 5 entregas anteriores, considerando:**
  - a. Un nivel de juego mínimo que incluya reglas mínimas que les permita jugar.
  - b. Manejo de usuarios y de inicio de sesión.
  - c. Comunicación entre un cliente y un servidor.
2. **Documentación** final actualizada de la aplicación:
  - a. Modelo **Entidad-Relación**
  - b. Documentación de la **API**
  - c. Readme con descripción de las **herramientas usadas** y todos los **pasos para ejecutar** la aplicación
3. Indicaciones precisas de parte de los ayudantes de lo que tienen que tener obligatoriamente implementado.

**Para optar al 6**, también se debe cumplir con criterios de calidad del código:

Ítems mínimos (deben cumplirse todos):

1. No usar Inline CSS
2. No usar html para estilo ni posicionamiento (sin tablas)
3. Unobtrusive JS y/o uso apropiado de React
4. Construir las rutas o URL de links usando `ctx.router.url` o similar (en lugar de hardcoded)
5. Validaciones bien definidas en los modelos
6. Convenciones JavaScript, recordar la referencia citada en este curso de estilo en Airbnb <https://github.com/airbnb/javascript> (camelCase y otras buenas prácticas del lenguaje, respeto sintaxis *linter*)

Ítems importantes que se tendrán en cuenta:

1. En las vistas, no usar consultas a db o lógica muy compleja
2. Sacar provecho de middlewares
3. Sin funciones gigantes en los route helpers ("fat models / skinny controllers")
4. Resource oriented
5. Usar métodos donde corresponden (MVC)

Si un grupo ya tiene todas las funcionalidades pedidas con anterioridad, para el 7 se le puede pedir una de las siguientes funcionalidades (siempre y cuando no estén ya implementadas en las entregas anteriores):

1. Capacidad de multipartida. Es decir, varios jugadores interactúan a su vez en varias partidas administradas simultáneamente por el servidor. Además que cada jugador pueda estar participando de más de una partida activa a la vez.
2. Lograr que se guarden todas las partidas del juego y que un usuario pueda ver el historial de sus juegos, tanto del actual como de partidas anteriores para el mismo usuario.
3. Permitir animar en el tablero una partida terminada desde su estado inicial, es decir las jugadas propias y la del resto de los jugadores, paso a paso mostrando cómo se han movido sus piezas, logros, etc.
4. Entregar un panel de control (dashboard) que muestre estadísticas de cada jugador y del resto de los jugadores. Las estadísticas pueden entregar un ranking relativo, cantidad de territorios, poder de sus jugadores, estado de sus héroes, etc. todo dependiendo del tipo de juego que han implementado.
5. Poder elegir el tipo de “terreno” o tablero en el que se desarrollará el juego. Para esto puede disponer de una cantidad de tableros que se puedan seleccionar y luego crear y ejecutar el juego en dicha configuración..

**La entrega de todos los archivos se hará en el repositorio de Github creado para su grupo.**

**NO se aceptarán:**

- Entregas por mail (ya sea al profesor o ayudantes)
- Entrega en otro sistema que no sea el que se ha provisto para estos efectos (el repositorio Github entregado por el coordinador y deploy con Heroku o equivalente)
- Respuestas atrasadas para la **evaluación de pares**

## **Condiciones y restricciones**

---

Aplican todas las restricciones generales entregadas en los talleres anteriores y se pueden sumar y agregar las acordadas con el ayudante.

## **Recomendaciones**

---

1. Si el protocolo es complejo o tienen aspectos de modelo de datos o de usabilidad que los complican, no teman de realizar cambios y ajustes necesarios para entregar con éxito su tarea. Redefinir o rediseñar no está mal en la medida que cumplan con los requerimientos establecidos en las entregas.

2. Implementen todos los puntos de “contacto” (como quien dice, los “end points”) que requieran y todas las APIs necesarias, no tiene que ser una sola y recomendamos que sean más atomizadas.
3. Aprenderán mucho más si trabajan colaborativamente en su grupo, como equipo en lugar de repartirse el trabajo y realizarlo como unidades independientes.
4. Recuerden que hay una evaluación de pares la cual no es un castigo al que no trabaje sino más bien una protección a los que sí se esfuerzan.
5. Si hay problemas con su compañero(a) y no lo pueden resolver, comuníquense con el profesor o con el coordinador.
6. Planifiquen el trabajo para que les permita la colaboración entre los integrantes del equipo
7. Pregunten y consulten, usen foro, colaboren entre ustedes (NO COPIEN). Los ayudantes están para apoyarlos
8. **Trabajen con tiempo**, no esperen a último momento para **comenzar con la tarea o despejar dudas**.
9. Siempre podrán, justificadamente, cambiar alguna regla, mejorar algún aspecto del juego, etc.

## **Dudas**

---

Para que todo el curso se vea beneficiado, hagan sus preguntas sobre el material del curso, sobre tecnologías web, y sobre el proyecto a través de los **foros del curso** dispuestos para estos efectos. No se responderá ninguna duda de tareas por e-mail.