



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

## Tarea 3

### IIC2513 Tecnologías y aplicaciones WEB

Segundo semestre 2020  
Fecha límite de entrega de tarea:, 20:00 hrs.

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 y la implementación de su trabajo.***

**Fecha límite de entrega de tarea 2:, 20:00 hrs.**

### **Indicaciones**

---

El objetivo de esta entrega es continuar con el diseño visual de su aplicación en la capa cliente, mejorar la usabilidad de la misma, ajustar elementos de diseño y agregar comportamiento a nivel de programación del lado del cliente (Javacript-DOM).

En esta entrega es indispensable y obligatorio que entreguen comportamiento (acciones) a nivel de browser. La página deberá ser ahora totalmente interactiva con el usuario.

En particular, se deberá entregar al menos tres **tipos de archivos**: *html*, *css* y *.js*. ¡Esto no significa que serán sólo tres archivos!, puede tener **varios para html, css y .js**, que es lo recomendado. Al respecto, estos archivos a entregar deberán contemplar **al menos** (incluyendo en gris elementos acumulados):

1. Un archivo index.html como punto de entrada
2. Un archivo con estilos para index.html
3. Un archivo index.js
4. Un archivo detallando el protocolo de comunicación para el archivo de envío de jugadas como también el de respuesta de parte del servidor. Se debe especificar toda la información que se enviará cliente -> servidor y servidor -> cliente en los momentos designados. Se debe incluir el formato de ésta. Se recomienda fuertemente el **uso de JSON**.

OBS: El protocolo de comunicación debe contener información del usuario conectado y todos los detalles que usted estime necesarios para que se realice una jugada efectiva.

OBS: El protocolo de respuesta del servidor debe incluir el número de jugada y toda la información requerida para que el browser pueda mostrar en pantalla el resultado del turno.

5. Una página de inicio de su juego (aplicación WEB) que sea “amigable” es decir, de navegación obvia, con títulos, menús, sub páginas y secciones que orienten al jugador.
6. Las reglas detalladas del juego a nivel de usuario en la página, organizadas con secciones y de facilidad para lectura.
7. Una sección “acerca de..”,
8. Una sección de “nosotros” donde se identifique el grupo (nombre) y se individualice a sus integrantes
9. Una sección para el login y registro de usuario. Utilizando **Local Storage**, por esta entrega (mientras no nos conectemos con el server), deberán permitir cambiar de usuario (login) guardando en el LocalStorage todos los usuarios registrados y comparando credenciales de acceso de acuerdo a lo almacenado en **Local Storage**.

OBS: *(DEMÁS está decir que este mecanismo es sólo temporal y JAMÁS deberán desarrollar un login y/o registro usando este mecanismo)*

10. La navegación para hacer una jugada y para enviar la jugada. En este caso la jugada DEBERÁ ser almacenada utilizando **Local Storage**, lo que permitirá

recuperar la jugada en cualquier momento y ser modificada cuantas veces quiera el usuario. El **formato** debe ser coherente con el **protocolo diseñado** que permitirá enviar la información al servidor, en próximas entregas.

OBS: Para los puntos 9 (temporal) y 10 deberán proveer mecanismos para recuperar la información de registro desde Local Storage (en el navegador del cliente) así como también permitir la modificación de la información cuantas veces se requiera.

11. Acciones que permitan al usuario ver el resultado de una jugada “simulada” como si hubiese sido entregada por el servidor. Es decir, un archivo indicando a un jugador el resultado de una jugada X, con sus pérdidas, ganancias, avances, estado de personajes, territorios, etc.
12. La interfaz de usuario para visualizar los mapas correspondientes.
13. Los botones para acciones, radio buttons, listas desplegables, puntuación, mensajes, etc. las cuales DEBERÁN ejecutar TODAS las acciones que puedan ser ejecutadas a nivel de cliente (browser), en particular:
  - a. Revisión del estado individual de unidades/personajes/territorios
  - b. Reportes en pantalla con información de recursos/unidades/territorios o cualquier otro elemento relevante como información para que el jugador pueda diseñar su estrategia de juego (pócimas de salud de un mago, armas especiales de algún héroe, etc.)
14. De ser necesario, deberán entregar también, en el directorio correspondiente, todas las imágenes y otros recursos requeridos para el despliegue correcto de su página
15. Un archivo **README.md**. Para esto, deben ya tener la información del template rellena en sus repositorios grupales (pueden modificar la estructura y diseño mientras incluyan apropiadamente la información pedida). Deben incluir al final una **sección para la tarea 3** que contendrá:
  - a. Consideraciones sobre su entregable en lo relativo al diseño de su HTML y mejoras/cambios con respecto a la entrega anterior.
  - b. Consideraciones sobre las reglas (cambios que hayan implementado)
  - c. Consideraciones sobre su archivo .js y las acciones que se pueden realizar a nivel de navegador (client-side)
  - d. Detalle de las mejoras que introdujeron a su diseño visual con el HTML y CSS así como usabilidad y acciones, respecto a la entrega 2
  - e. Cualquier tipo de supuesto, restricción, o información relevante para corregir su entrega

**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 github entregado por el coordinador)

### **Estructura de archivos:**

Una estructura apropiada para su repositorio tras esta entrega **podría** verse como el siguiente ejemplo:

```
-
+-- documents/
+-- src/
|   +-- assets/
|   |   +-- styles/
|   |   +-- imgs/
|   |   +-- scripts/
|   +-- views/
|   |
|   +-- lib/
+-- index.html
+-- index.js
+-- .gitignore
+-- README.md
-
```

### **Condiciones y restricciones**

---

1. En esta entrega se espera que se entregue la mayor cantidad de programación a nivel de cliente. Para cualquier necesidad de probar **comunicación con el server**, simularlo con **Local Storage** (no debiese ser mucha información)
2. Se debe entregar una aplicación **totalmente funcional a nivel de cliente** (puede mejorar aspectos de manejo cliente en las siguientes entregas) con estructura, estilo y comportamiento.
3. Las reglas del juego pueden sufrir variaciones con respecto a la entrega anterior, estas modificaciones **deben señalarlas** en su archivo README.txt

### **Recomendaciones**

---

1. Piensen en distintas opciones de usabilidad e interacción con su página WEB. diseñen varias veces para encontrar la mejor forma de abordar su tarea. No se queden con la primera idea que les venga a la cabeza.
2. Pongan atención a la navegación, al protocolo que diseñen y a la mecánica del juego.

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. Diseñen muy bien las reglas, el entorno, los turnos, la resolución de empates y conflictos
7. Planifiquen el trabajo para que les permita la colaboración entre los integrantes del equipo
8. Pregunten y consulten, usen foro, colaboren entre ustedes (NO COPIEN). Los ayudantes están para apoyarlos
9. **Trabajen con tiempo**, no esperen a último momento para **comenzar con la tarea o despejar dudas**.
10. No traten de resolver aún detalles específicos de integración o comunicación con el servidor. Sin embargo **ya es momento que vayan pensando en sus modelos de soporte en back-end (base de datos)**
11. 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.