



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

Entrega 3 de proyecto

IIC2513 Tecnologías y aplicaciones WEB

Primer semestre 2022|

Fecha límite de entrega de entrega: Viernes 03 de junio 23:00 hrs.

Las entregas 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 entrega 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 : Viernes 03 de junio, 23:00 hrs.

Indicaciones

El objetivo de esta entrega es unir de forma completa las partes cliente y servidor de su proyecto y entregar la primera versión de su juego. En esta entrega es indispensable y obligatorio que entreguen comportamiento (acciones) a nivel de navegador (browser). Su juego deberá ser ahora totalmente interactivo con el usuario.

Se espera que su juego al menos sea capaz de:

1. Presentar una página inicial (landing page) que introduzca las características del juego y permita la acción de registrarse y/o ingresar (login) al sitio.
2. Usar de manera adecuada middlewares, ya sea con Koa (recomendado) o Express.
3. El registro debe permitir a cualquier persona crear una cuenta con nombre de usuario y clave que el disponga. Como regla, no se debe permitir la repetición del nombre de usuario.
4. El ingreso (login) de un usuario debe validar correctamente el ingreso de la clave de usuario e iniciar sesión, para ello se espera utilicen koa-session y JWT para manejo de sesión, accesos y privilegios.

OBS: Se pueden usar cookies como mecanismo básico para sesión (**por sí solas no bastan para la banda A**), pero se recomienda un sistema más avanzado. Por ejemplo, si usan JWT pueden usar el *Bearer* o *Authotization Header* sin cookies, lo cual puede ser recomendable para evitar problemas con CORS. Se permite exigir el token en cada *request*.

5. Manejo de seguridad en base de datos y en inicio de sesión. Uso de *crypto* o *bcrypt*. Guardar contraseñas encriptadas. Ocultar de lado navegador la password al ingresarla.
6. Deben tener un usuario admin (administrador) que permita borrar jugadas, eliminar usuarios y mantener los aspectos relevantes del juego.
7. Se espera que la usabilidad de su juego (aplicación WEB) sea “amigable” es decir, de diseño cuidado, navegación intuitiva, con títulos, menús, sub páginas y secciones que orienten al jugador.
8. Se espera también tener las reglas detalladas del juego a nivel de usuario en la página, organizadas con secciones y de facilidad para lectura. Se valorará ayudas online sensibles al contexto (esto significa que existan tutoriales o bien tooltips).
9. Gestionar el **flujo completo de las partidas** para grupo de jugadores. Entendiendo una partida como un juego completo desde su configuración inicial (escenario de partida en el tablero) y adición de jugadores, hasta el fin del juego con la determinación del jugador ganador y otros resultados.
10. Administrar la concurrencia de varios jugadores conectados al mismo tiempo consultando al servidor. Los jugadores se entienden pueden pertenecer todos a la misma partida. **No se exige “multipartidas”**, pero se entiende que lograr un manejo de muchas partidas simultáneas es un criterio para la banda A.
11. La navegación para unirse a un juego y para hacer y enviar la jugada del juego (partida) en curso. Cabe destacar que hay dos estrategias (obvias) posibles, que cada movimiento en el tablero sea una jugada que se envíe al servidor y se procese (y almacene) en backend o bien que el usuario pueda “jugar” en frontend (hacer y deshacer sus movimientos) y luego decida enviar la jugada

para ser procesada en backend. Usted puede elegir cualquiera de esas dos aproximaciones.

12. Los endpoints mínimos que se espera ustedes implementen: GET, POST (o PUT), DELETE. Se valorará la coherencia entre usar PUT/POST/PATCH (eso se determinará con justificación que ustedes hagan al respecto. Justificaciones documentadas, claro está). También se valorará el usar PATCH (que no es exigible como vieron) así como también se valorará usar, adecuadamente, otro tipo de request HTTP (como HEAD).
13. Entregar **Documentación** básica **de cada endpoint de la API** expuesta por el servidor. Documentar tanto *requests* como *responses*.
14. Utilizar elementos del DOM que impliquen evento como botones para acciones, radio buttons, listas desplegables, etc. las cuales DEBERÁN ejecutar TODAS las acciones que puedan ser ejecutadas a nivel de cliente (browser).
15. Se espera el uso (básico) de REACT en la parte frontEnd, por lo tanto, documenten sus componentes. Se valorará el uso más intensivo de REACT.
16. Se valorará positivamente el uso de buenas prácticas de programación, estructuras de archivos y directorios, diseño modular y escalable, reutilización de código, código documentado y otros aspectos de calidad en entrega. Por ejemplo, los códigos gigantes y monolíticos no son bien vistos.
17. Se valorará el uso de Github como elemento de calidad: Uso de branches que se asocien a una funcionalidad específica y commits descriptivos, además de la revisión de PRs.

USEN: https://github.com/airbnb/javascript como guía de estilo
--

18. Recuerden su archivo **README.md** 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 los repositorios de su grupo.

Estructura de archivos:

Una estructura apropiada (SI NO USAN EL TEMPLATE) 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. Se debe entregar una aplicación **totalmente funcional a nivel de cliente** que se comunique de manera efectiva con el servidor y se calculen las jugadas según sus reglas de juego (puede mejorar aspectos de manejo cliente en las siguientes entregas) con estructura, estilo y comportamiento.
2. 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 entrega. 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. Recuerden mirar y utilizar la guía de estilo entregada en este enunciado (<https://github.com/airbnb/javascript>)
4. Diseñen adecuadamente sus componentes para usar en REACT.
5. ¿Se complicaron con REACT? pueden seguir sólo con Javascript-DOM para esta entrega, pero para la última entrega esperamos su SPA y REACT en un uso al menos básico.
6. Aprenderán mucho más si trabajan colaborativamente en su grupo, como equipo en lugar de repartirse el trabajo y realizarlo como unidades independientes.

7. Diseñen muy bien las reglas, el entorno, los turnos, la resolución de empates y conflictos
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 entrega o despejar dudas.**
10. 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 entrega por e-mail.