



Entrega Final

IIC2513 Tecnologías y aplicaciones WEB

Fecha límite de entrega: LUNES de 21 noviembre **11:00 AM** hrs.

Recuerden que los enunciados sólo 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, LUNES 21 de noviembre 11:00 AM hrs.

Nota: Como está indicado en el programa del curso, para la última entrega se permitirá el uso de un **máximo de dos cupones de atraso**.

Indicaciones

¡Por fin! la última entrega y a estas alturas del semestre ustedes ya saben comunicar un cliente (browser) con un servidor, atendido por un middleware y que implementa APIs (o bueno, rutas o endpoints que permiten acceder a recursos usando métodos HTTP dados por GET y al menos uno de los siguientes: POST/PUT/PATCH. Junto a eso, pueden trabajar con DOM y CSSOM para lograr una renderización en el browser y utilizar eventos que respondan a la interacción del usuario. Eso es bastante para estos meses. ¿No creen? Felicitaciones si asintieron con su cabeza pues quiere decir que están cerca de lograrlo.

El objetivo de esta **última entrega** es **arreglar, mejorar, enmendar y dejar a punto su proyecto**. Es decir, entregar una aplicación WEB funcional y que cumpla en mayor parte con su diseño original.

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! En particular, es el momento de atacar todos aquellos aspectos que han sido señalados por los



ayudantes, para terminar su trabajo, de forma integral, entregando una aplicación funcional y sin baches.

En esta entrega es indispensable y **obligatorio** que entreguen:

1. **Aspectos mínimos de todas las entregas anteriores, considerando:**
 - a. Un nivel de usabilidad mínima que incluya reglas de negocio mínimas que les permita interactuar con su aplicación.
 - b. Manejo de usuarios, es decir el registro, de inicio de sesión y manejo de permisos/roles/perfiles, de manera SEGURA.
 - c. Comunicación completa entre el cliente y el servidor.
2. Capacidad de multipartida/multisimulación. Es decir, varios usuarios interactúan a su vez en varias partidas/simulaciones gestionadas simultáneamente por el servidor. Además que cada usuario puede estar participando de más de una partida/simulación activa a la vez.
3. Implementación del **mecanismo de respaldo de seguridad** de los datos de usuario de su aplicación y que deberá ser desplegado (deploy) el front en Netlify (u otro) y el back en Heroku (u otro). Este mecanismo de respaldo de seguridad es el que está enunciado en la entrega anterior, en esa ocasión era opcional pero en esta entrega se va a exigir.
4. **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
5. Un conjunto de tests automatizados de su aplicación. Estos test deben ser pruebas unitarias para su código Javascript así como tests para sus endpoints. Para esto, usen JEST (otras opciones de testing como Mocha o Jasmine, conversen con su ayudante).
6. Uso **obligatorio** de linter (Eslint es el recomendado)
7. 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*)
8. 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. 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. Control de errores (más comunes)



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. Usar métodos donde corresponden (MVC)

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

1. Lograr que se guarden todas las partidas/simulaciones y que un usuario pueda ver el historial de sus juegos/simulaciones, tanto del actual como de partidas/simulaciones anteriores para el mismo usuario.
2. Permitir animar en el tablero una partida/simulación terminada (completada) desde su estado inicial, mostrando el paso a paso de cómo se desarrolló la simulación/partida.
3. Entregar un panel de control (dashboard) que muestre estadísticas de cada usuario y del resto de los usuarios. Las estadísticas pueden entregar un ranking relativo, cantidad de territorios, poder de sus jugadores, estado de sus héroes, simulaciones más exitosas, parámetros de simulación óptimos, ranking de simulaciones, peores escenarios, etc. todo dependiendo del tipo de aplicación que hayan implementado.
4. Otros criterios que se definan de común acuerdo con su ayudante,.

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)

Condiciones y restricciones

Aplican todas las restricciones generales entregadas en los enunciados anteriores y se pueden sumar y agregar las acordadas con el ayudante. Recuerde que en la última entrega sólo se podrán usar un **máximo de DOS cupones de atraso** por lo que la entrega puede ser realizada, **a más tardar** para quienes tienen cupones suficientes, el **miércoles 23 a las 11:00 AM**.

Recuerden que se mirará en detalle el uso de Github por parte de **TODOS** los integrantes del grupo. Un integrante que no tenga actividad en github (commits)



podrá ser penalizado con un **porcentaje de descuento** de la nota que obtenga su grupo.

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 proyecto. 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” (“endpoints”) 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. **RECUERDEN, el integrante que no tenga movimientos en *github*, será penalizado en su calificación.**
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, comuníquese 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 el desarrollo.**
9. Siempre podrán, justificadamente, cambiar alguna regla, mejorar algún aspecto del juego, etc.

10. Fecha límite de entrega, LUNES 21 de noviembre 11:00 AM hrs.