



Enunciado entrega 1

IIC2513 Tecnologías y aplicaciones WEB

Fecha límite de esta entrega: lunes 8 de abril, 23:59 hrs.

Indicaciones

El objetivo de esta primera entrega es la presentación de su juego. Se espera que entreguen un diseño preliminar de cómo se verá su juego en el browser y cuáles serán las reglas que rijan su juego.

Consideren este trabajo una propuesta tipo borrador (**nada de programación**).

Introducción

Lo que comparten en común **todas** las aplicaciones web es:

- Un propósito**, es decir lo que prometemos entregar al usuario (resolver una necesidad)
- Originalidad** en su propuesta. Eso no significa que deba ser algo “nuevo”, la originalidad muchas veces es presentar de otra forma lo que se viene usando
- Una arquitectura cliente-servidor
- El protocolo que sigue su aplicación, con solicitudes y respuestas de un cliente a un servidor (como fue mostrado en clases)
- El manejo de sesiones de usuario, que pueden incluir credenciales del tipo login-Password (usuario y contraseña)

A través de las clases de cátedra y ayudantías y de este proyecto, **ustedes adquirirán todos los conceptos** necesarios para entender cualquier aplicación WEB, sus partes fundamentales, la mecánica requerida y la arquitectura base.

Es por eso que una parte relevante de una aplicación WEB es el diseño general de todos sus aspectos en cuanto al **PROPÓSITO**. ¿Qué hace? ¿Cómo lo hace? ¿Cómo interactúa el usuario? ¿Qué reglas rigen a nuestra aplicación?, etc.

Sobre la entrega

Deberán entregar un documento pdf (nada de código aún) con:

- Las características básicas de la aplicación que implementará, incluyendo la definición del propósito de su aplicación, lo que esta aplicación hará. Por ejemplo “Este es un juego de estrategia que permitirá, a un máximo de 4 jugadores...”
- El comportamiento de su aplicación de manera detallada, describiendo las funcionalidades que se le presentarán a los diversos usuarios (casos de uso) y lo que debe realizar el usuario



para interactuar y obtener los resultados esperados (comprometidos) de su funcionalidad en específico. Es decir, todos los aspectos de interacción para todos los tipos de usuario. Debe detallarse el “viaje del usuario” desde que se conecta (y también como se registra), ingresa sus credenciales, interactúa (O sea, cómo se juega, turnos, partidas, etc.) y concluye su uso (como termina el juego)

3. Las reglas que gestionará el servidor para tratar cada interacción (descripción de la mecánica de las reglas, *nada de programación*) Por ejemplo: **Regla n-ésima¹**: “Cada vez que el jugador caiga en una casilla no explorada por él, el servidor la podrá asignar al jugador que la descubrió siempre y cuando no tenga un dueño previo. Si tiene dueño previo el jugador podrá decidir si ataca al dueño original de la casilla recién descubierta o no”.
4. Un primer borrador de su protocolo de comunicación. Esto es, qué información se debe mandar para que el servidor procese la jugada Por ejemplo: Cada movimiento que haga un jugador enviará al servidor una tupla con: (id del jugador, casilla a la que se movió, listado de acciones que realizó)
5. El diseño de la página “landing page”, esta es la página de inicio (el index.html) que se llega por defecto cuando se va a la dirección WEB de su aplicación. Este tipo de páginas tienen típicamente un “acerca de...”, las reglas de operación o juego, y otra información de carácter general. ¡**Apliquen imaginación y entreguen contenido original!**
6. Deberán considerar el comportamiento para un usuario registrado y para otro no registrado. Por ejemplo, un usuario no registrado sólo podrá ver cómo está el resultado/avance de una partida que esté catalogada como pública.
7. Para los usuarios registrados deberán considerar varios perfiles y roles diferentes, por ejemplo usuario administrador, usuario jugador y los comportamientos entre los usuarios registrados, por ejemplo, un usuario que sea el dueño de un juego es distinto de otro que ha sido invitado a participar.
8. Un mockup para el registro de un usuario (¿tendrá un avatar?, ¿logo? ¿Qué datos pedirán?, etc)
9. El registro para comenzar o integrarse a un juego. En este caso la lógica que seguirá el registro de usuario/jugador a un juego en particular (no es el registro de un usuario como “persona” en el sitio) y el saber cómo se unirán a otros participantes ¿cómo se “juntan” varios usuario para el uso de la aplicación? ¿Hay algún timeout?
10. La navegación (es decir la interacción con la pantalla de usuario en el browser) ¿Selección de menús? ¿Drag and drop? ¿combinaciones de lo anterior? ¿Hot-keys? (teclas especiales que al presionarlas hacen algo). En este caso se requiere un mock-up (es decir, pantallas dibujadas a modo de borrador, utilizando para ello un programa gratuito que les permita hacer mock-ups, incluso puede ser en una ppt que grafique el uso de la pantalla). Junto al mockup se requiere, por supuesto, explicaciones y descripciones de las pantallas.
11. La interfaz de usuario permite visualizar los tableros, gráficos y/o mapas correspondientes que usted defina.
12. Mock-ups y explicación de los mismos para los tableros de control, por ejemplo en el juego ¿habrá marcación de puntaje? ¿Habrà información pública (para todos los participantes) y privada (para un grupo o un sólo participante?). Por ejemplo, en un juego de cartas, yo puedo ver las que están en la mesa pero nadie puede ver “mi mano”.
13. Descripción preliminar de los objetos y/o instancias que intervendrán (por ejemplo en un juego de aventuras, los tipos de personajes, las características de los personajes, el modelo

¹ Por regla n-ésima nos referimos a que es una regla cualquiera dentro de muchas otras reglas



- de objetos asociado a ello, herencias posibles, diferentes tipos de armas, o de “encantamientos”, niveles que puede obtener cada personaje, etc.)
14. Modelo preliminar de entidad relación (trabajaremos con Postgres)
 15. Deberán incluir eventos “fortuitos” o aleatorios que afecten la jugada, por ejemplo “cartas de destino” o lanzamiento de dados o casillas “sorpresa”, etc.

Forma de entrega

Se realizará entregando un documento en PDF que incluya todo lo anterior solicitado además de todos los otros puntos que ustedes consideren relevantes para explicar los alcances de su aplicación.

NOTA: Si tienen un programa de mockups, por ejemplo, que genera otro formato distinto a pdf, pregúntenle al ayudante asignado a su grupo si se puede y las restricciones a ese tipo de entrega.

NO se aceptarán:

- Entregas por mail (ya sea al profesor o ayudantes)
- Entrega de documentos que no sea en PDF (salvo el acuerdo previo con el ayudante)

Sobre los mockups

Los mockups no son un diseño final, sino más bien un esquema de los elementos que tendrá su aplicación, cómo están organizados dichos elementos y algo de diseño preliminar (cuando vean las herramientas para hacer mockups se darán cuenta de los elementos con los que pueden trabajar). Pueden ver varios ejemplos de mockups en la web.

Pueden hacer los diagramas a lápiz y después escanearlos o pueden usar la herramienta de diseño que más les guste. Algunos ejemplos de herramientas online y colaborativas son Balsamiq, Cacao y Mockingbird, aunque en general tienen sólo trials gratuitos. Una herramienta de escritorio y completamente gratuita es Pencil.

Finalmente deben entregar un PDF que se pueda entender con facilidad.

Sobre el Modelo de Datos

Representen su modelo de datos con un diagrama ER (Entidad Relación). Ésto les permitirá tener algo intermedio que luego podrán mapear tanto a tablas de bases de datos como a un modelo orientado a objetos. Traten de capturar lo más que puedan pero no es necesario que sea absolutamente exhaustivo (muy probablemente cambiará durante el desarrollo del proyecto). Sólo los atributos que obviamente se van a necesitar. **Privilegien la simplicidad cuando sea posible.**

Hay muchas herramientas para generar este tipo de diagramas y pueden utilizar la que más les guste. Si quieren un ejemplo: <https://erdplus.com>.

Recomendaciones



1. Definan bien claramente el tipo de juego que presentarán. ¿Es de conquista? ¿de uso de recursos? ¿de competición por quién llega primero a una meta?
2. Revisen juegos de mesa conocidos (Catán, Wingspan, Ataque, The red cathedral, Scythe, etc.) u otros como los famosos “play by eMail (o mail) pueden ver un listado en:

https://en.wikipedia.org/wiki/List_of_play-by-mail_games

3. Recuerden que el juego es por turnos, **no es una partida simultánea en tiempo real**. Tal vez un problema sea el resolver cómo “avisar” que el otro tiene que jugar. Piensen en estrategias adecuadas y eficientes para ello. Los turnos pueden ser cíclicos (juega uno, luego el otro y así sucesivamente) pero también puede ser que todos deban entregar su jugada antes de cierto límite de tiempo (fecha y hora dadas) y el servidor mezcle todas las jugadas y resuelva. Piensen también en la resolución de empates y conflictos
4. Piensen en distintas opciones de presentación y la mejor forma de incorporar la funcionalidad en su proyecto. **No se queden con la primera idea que les venga a la cabeza**
5. Aprenderán mucho más si trabajan colaborativamente en su grupo, como equipo en lugar de repartirse el trabajo y realizarlo como unidades independientes
6. Investiguen y vean aplicaciones que los puedan inspirar, se deberá citar toda fuente de inspiración relevante deberán mencionarla (la fuente y como llegar a ella)
7. Diseñen muy bien las reglas, el entorno de su aplicación, las funcionalidades, la interacción, los usuarios y sus perfiles/roles
8. Planifiquen el trabajo para que les permita la colaboración entre los integrantes del equipo
9. Pregunten y consulten, usen las issues, colaboren entre ustedes (**NO COPIEN**) los ayudantes están para apoyarlos
10. **Trabajen con tiempo, no esperen a último momento para comenzar con la tarea o despejar dudas**
11. Comiencen con una interfaz de usuario (UI) simple; **no es necesario que su aplicación se vea “bonita” o “totalmente completa” desde el comienzo**
12. No traten de resolver aún detalles específicos de integración, comunicación, consultas a BDD u otros temas de implementación ajenos al alcance de esta primera entrega
13. Siempre podrán, justificadamente, cambiar alguna funcionalidad o regla, mejorar algún aspecto de su aplicación, etc.
14. Si hay problemas con algún compañero(a) y no lo pueden resolver, comuníquese con el la coordinadora de ayudantes o con el profesor

Dudas



Para que todo el curso se vea beneficiado, hagan sus preguntas a través de las issues del curso en el repositorio en GitHub. No se responderá ninguna duda de entregas de proyecto por e-mail.

Ejemplo (que pueden usar si lo desean)

El juego y las reglas que se describen a continuación, son las **mínimas** para un tipo de juego. Hay mucho en el diseño que dilucidar y también en la forma del juego, los turnos, etc. Esto es un ejemplo que puede inspirarlos:

Juego de dominio de territorio y recursos

El objetivo típico de estos juegos es conquistar todos los territorios o ciertas zonas específicas, incluso algún tótem, bandera o ícono particular. Este tipo de juego implica enfrentamientos entre los participantes y debe contemplar al menos:

1. Territorios que se enfrentan (puede ser la tierra, star trek, imperio romano, fantasía que ustedes quieran)
2. Cada territorio posee un propietario, límites y recursos que entrega (hay territorios con más recursos, otros con menos)
3. Adicionalmente un territorio puede recibir mejoras en defensa los cuales servirán de “bono” a la defensa de una agresión desde otro territorio
4. Los territorios pueden tener o no una unidad de defensa
5. Pueden haber distintos tipos de unidades que defienden un territorio (por ejemplo legiones, manipulas, super legiones, ejércitos, etc) Las unidades tienen cierta cantidad de puntos de ataque, cierta cantidad de puntos de defensa y cierta “resistencia”
6. Cada turno, se calcula la cantidad de recursos y se le entregan al jugador específico el cual deberá gestionar sus recursos para mejorar defensa de un territorio, mejorar la capacidad de ataque o defensa de una unidad en específico
7. Las unidades solo se pueden mover a un territorio adyacente.
8. Si el territorio le pertenece al mismo jugador, simplemente la unidad se mueve
9. Si el territorio le pertenece a otro jugador, se considerará un ataque
10. Por turno, el computador deberá calcular ataques, defensas y mejoras de territorios según las reglas y coherencia que ustedes determinen. Algo incoherente, por ejemplo, sería que un territorio conquistado, pueda mejorar en el mismo turno su defensa luego de ser conquistado. El computador en caso de empate debe determinar por algún método el ganador (¿un random? ¿simular tirada de dados?)

Elementos mínimos que debe tener su juego



Al definir su juego y presentarlo al ayudante éste puede ser aceptado, aceptado con reparos (que deben mejorar algún aspecto) o rechazado.

Lo importante del juego que postulen, es que deben tener (si o si):

1. Al menos 2 jugadores que jugarán por turnos no sincrónicos (uno primero, otro después)
2. Un mapa o tablero donde jugar (casillas, mapa, otro)
3. Características de las casillas del mapa o tablero (recursos, “sorpresas”)
4. Personajes, unidades u otro “objeto” que requieren algún tipo de recurso
5. 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)
6. Posibilidad de mejora de unidades, territorio, personajes, etc.
7. Un servidor (back-end) donde se dirige y comunica el resultado del turno
8. Estados almacenados del juego y visualización del mismo en un tablero que permita a un jugador seguir (visualizar) el desarrollo del juego

Éxito y disfruten el viaje

Apéndices

1. Sobre las bandas de calificación

Se crearon las bandas de calificación pues el propósito del proyecto y del curso, es entregar aplicaciones web funcionales (es decir, que hagan cosas), es decir se quiere calificar el valor que entrega la aplicación desde el punto de vista de su uso.

Una calificación que otorgue décimas por partes y piezas de un conjunto que no funcione puede terminar con una nota 4 o 5 para una aplicación que no haga nada. ¡Queremos que aprendan a hacer! y queremos que lo que hagan funcione :-)

2. Sobre la “utilidad” del proyecto

Las clases y el proyecto le entregarán todo lo necesario para entender cómo funciona cualquier aplicación WEB. La “utilidad” de una aplicación Web está dada por lo que ustedes hagan con las partes y piezas que sabrán ensamblar y entender sus beneficios, restricciones y usos para otorgar un valor buscado por un usuario. Así pues, los juegos son útiles pues enseñan a pensar de manera innovadora, permiten explorar al máximo *la interacción en front, tiene manejo de sesiones, entrega de información parcial o específica según roles/usuarios, manera historial de un juego y de las*



jugadas, acumular puntaje, ver partidas anteriores, tener diferentes niveles de usuarios, trabajar la confidencialidad de la información, entregar tableros de control, etc. Si revisan todo lo que se listó (que está en verde) como característica de un juego y lo reemplazan por un sitio de e-commerce, encontrarán que pueden reemplazar cada concepto y tendrá una tienda Web (Hagamos el ejercicio: *interacción en la tienda, tiene manejo de sesiones, entrega de información parcial o específica según roles/compradores, manera historial de compra y de las compras, acumular productos y precios, ver compras anteriores, tener diferentes niveles de usuarios, trabajar la confidencialidad de la información, entregar tableros de control*)

De todas formas mi recomendación es **no** hacer sitios de e-commerce desde cero, mejor usen: Magento, Woocommerce, Shopify, Ecwid, Opencart, Weebly, Bigcommerce, Tienda nube, Wix. O vean, por ejemplo:

<https://www.tooltester.com/en/blog/shopify-alternatives/>