



Enunciado entrega 1

IIC2513 Tecnologías y aplicaciones WEB

Fecha límite de esta entrega: sábado 02 de septiembre, 23:45 hrs.

Indicaciones

El objetivo de esta primera entrega es la presentación de su proyecto. Se espera que entreguen: la maqueta de cómo se verá (algo muy preliminar), la funcionalidad de su aplicación (es decir, qué ocurrirá en front-end), las reglas de negocio (el comportamiento ante las acciones en front-end, mucho de ello será parte de back-end como es consultas a bases de datos). Consideren este trabajo una propuesta tipo borrador (**nada de programación**).

Adicionalmente se espera un diseño de su página de inicio (landing page o el famoso index.html), esquema para perfiles de usuario (jugador, administrador, etc.), casos de uso que tendrá su aplicación, funcionalidades, botones, “pestañas”, estructura de las páginas y otros aspectos que permitan, a un usuario cualquiera, entender la aplicación y su mecánica.

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. El mejor caso es Chico Trujillo (¿lo conocen?). Se hizo famoso re-interpretando, de una forma original y nueva, canciones realmente antiguas
- 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.



La entrega

En particular, **deberán entregar un documento (nada de código aún) con:**

1. Las características básicas de la aplicación que implementará. 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...”
2. 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)
4. Un primer borrador de su protocolo de comunicación. Esto es, qué información se debe mandar para que el servidor procese la jugada (Ej. id del jugador, casilla a la que se movió, acciones que realizó, recursos que compró, etc.)
5. El diseño de la página “landing page”, esta es la página de inicio 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. **¡Sorpréndenos con contenido original!**
6. Deberán considerar el comportamiento para un usuario registrado y para otro no registrado (por ejemplo, un no registrado podrá ver cómo está el resultado de una partida). 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.
7. Un mockup para el registro de un usuario (¿tendrá un avatar?, ¿logo? ¿Qué datos pedirán?, etc)
8. 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 un timeout?
9. 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.
10. La interfaz de usuario que permite visualizar los tableros, gráficos y/o mapas correspondientes que usted defina.
11. 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”.



12. 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 de objetos asociado a ello, herencias posibles, diferentes tipos de armas, o de “encantamientos”, niveles que puede obtener cada personaje, etc.)
13. Modelo preliminar de entidad relación (trabajaremos con Postgres)
14. 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?
15. Deberán incluir eventos “fortuitos” o aleatorios que afecten la jugada, por ejemplo “cartas de destino” o lanzamiento de dados o casillas “sorpresa”, etc.
16. Recuerden que el juego es por turnos, **no es simultánea** la partida. 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 y el servidor mezcle todas las jugadas y resuelva. Piensen también en la resolución de empates y conflictos

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. 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**
2. Aprenderán mucho más si trabajan colaborativamente en su grupo, como equipo en lugar de repartirse el trabajo y realizarlo como unidades independientes
3. Investiguen y vean aplicaciones que los puedan inspirar, toda fuente de inspiración relevante deberán indicarla (la fuente y como llegar a ella)
4. Diseñen muy bien las reglas, el entorno de su aplicación, las funcionalidades, la interacción, los usuarios y sus perfiles/roles
5. Planifiquen el trabajo para que les permita la colaboración entre los integrantes del equipo
6. Pregunten y consulten, usen las issues, colaboren entre ustedes (**NO COPIEN**) los ayudantes están para apoyarlos
7. **Trabajen con tiempo, no esperen a último momento para comenzar con la tarea o despejar dudas**
8. Comiencen con una interfaz de usuario (UI) simple; **no es necesario que su aplicación se vea "bonita" o "totalmente completa" desde el comienzo**
9. 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
10. Siempre podrán, justificadamente, cambiar alguna funcionalidad o regla, mejorar algún aspecto de su aplicación, etc.
11. 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.

Tipos de juegos sugeridos

Los siguientes son tres tipos de juegos que pueden utilizar y su detalle. Las reglas que se describen a continuación, son las **mínimas** por tipo de juego. Hay mucho en el diseño que dilucidar y también en la forma del juego, los turnos, etc. ¡Eso es parte de lo que ustedes harán!

1. Dominio de territorio y recursos

El objetivo típico de estos juegos es conquistar todos los territorios, las “capitales” 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?)

2. Gestión de organización o país u otro



El objetivo de este juego es por lo general obtener cierta cantidad de puntos, donde el ganador es el que sobrevive o el que tiene mayor puntaje enfrentado a las dificultades aleatorias que el servidor arroja y que ustedes deben resolver con sus recursos y eventual colaboración entre participantes. Este juego debe tener al menos:

1. Repartir aleatoriamente recursos escasos a la organización, por ejemplo “ciudades” y pueden recibir presupuesto, alimentos, profesionales u otros
2. Los recursos sirven para producir ciertos resultados ya sean finales o intermedios
3. Los recursos intermedios son producidos por alguna cadena de producción que ustedes determinen. Por ejemplo, se requiere carbón y hierro para producir acero y acero para producir arados y arados para producir comida (las cadenas las definen ustedes)
4. Deben tener al menos 2 producciones de tipo “cadena de producción”, es decir, recursos iniciales que se convierten en otro final
5. Puede haber solicitud de comercio al servidor el cual intercambia alimento por dinero y viceversa
6. Al final todos los recursos producen alguna unidad de intercambio (monetaria u otra) que sirve para la gestión
7. Para poder producir lo que sea, se deben tener especialistas que requerirán cierta cantidad de recursos para mantenerlo.
8. Los recursos de un especialista son de dos tipos: monetarios y de alimentación
9. Hay especialistas en producir comida (por ejemplo agricultores), otros en producir bienes (que al final es recurso monetario)
10. Usted deberá equilibrar sus recursos
11. Debe haber algunas restricciones, por ejemplo el terreno para producir alimentos es limitado
12. Deben tener la posibilidad de mejorar (vía recursos monetarios) a los especialistas para que logren más producción
13. Los especialistas requieren diversos usos de recursos monetarios y de alimentación
14. Aleatoriamente el servidor puede arrojar algún tipo de catástrofe a todos los jugadores a la vez que afecte la generación de alimentos o de recursos monetarios (por ejemplo, una inundación hace que la producción de alimentos caiga un 20%)
15. Si la producción no alcanza para mantener a los especialistas, estos se van del lugar
16. El juego comienza con una cierta cantidad de recursos, terreno y especialistas fija para cada jugador.

3. Aventura/descubrimiento

El objetivo de este juego es por lo general obtener cierta cantidad de elementos de recompensa (por ejemplo tesoros, banderas, etc). antes que el resto de los jugadores. Los jugadores poseen un equipo especialista y van avanzando por un mapa “oculto” que a medida que van saltando de casilla en



casilla, descubren, con sus recompensas o sus dificultades (o a veces, nada). Este juego debe tener al menos:

1. Un tablero (casillas hexagonales) y diversos equipos descubriendo el territorio común (mapa desconocido)
2. Los jugadores poseen un grupo de personajes (o razas) particulares
3. Típicamente un personaje “fuerte” o guerrero, otro mágico, otro que sana, etc.
4. Los jugadores se desplazan por el tablero, descubriendo la zona y obteniendo las recompensas ocultas en el mapa
5. El mapa es generado por ustedes, pero los recursos son colocados en el mapa de forma aleatoria. por el servidor
6. Hay recursos de tipo renovable (que los puede obtener más de un jugador) y hay recursos que se extinguen, es decir, lo obtiene el que primero lo recupera.
7. El “grupo” de personajes se mueven todos en bloque
8. En las casillas también hay amenazas que deben sortear (un dragón, un mago malvado, etc)
9. Si dos o más equipos van a una misma zona (casilla) del mapa, se enfrentan y luchan por la zona.
10. Ustedes determinan las reglas del enfrentamiento
11. Los personajes pueden “mejorar” de acuerdo a los recursos que hayan obtenido
12. Hay un casilla “zona cero” que es exclusiva de un jugador (inconquistable) y que solo sirve para curar al equipo. Un equipo herido es enviado por el servidor automáticamente a esa casilla para reponer fuerzas cuando uno o más de los personajes tiene una salud menor a cierto parámetro.
13. Los personajes consumen energía cada vez que tienen enfrentamientos o pueden usar cierta cantidad de unidades de su habilidad por turno (por ejemplo, el que sana puede entregar cierta cantidad de puntos de sanación por turno, al igual que el mago)

Postulación de otro tipo de juego

Si postulan un tipo de juego diferente, este puede ser aceptado, aceptado con reparos (que deben mejorar algún aspecto) o rechazado. por supuesto pueden postular un tipo de juego que mezcle los tipos de juego presentados en este enunciado. Lo importante del juego que postulen, es que deben tener:

1. Al menos 3 jugadores que jugarán por turnos (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 que dirime y comunica el resultado del turno
8. Estados almacenados del juego y visualización del mismo en un tablero que permita a un jugador seguir el desarrollo del juego

Éxito

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/>