



Aspectos y consideraciones generales sobre el proyecto

IIC2513 Tecnologías y aplicaciones WEB

Segundo semestre 2022

Introducción

El proyecto, a desarrollar a lo largo del curso, tiene como objetivo que los estudiantes enfrenten un desafío de diseño, programación, implementación y despliegue en ambiente web, que les permita poner en práctica los conocimientos adquiridos durante este semestre.

Las entregas del proyecto están vinculadas entre sí y la última de las entregas tendrá como resultado una aplicación totalmente funcional con un lado cliente y otro servidor (¡en la nube!).

Para lograr lo anterior, usted deberá cumplir con objetivos intermedios, mejor conocidos como “entregas parciales de proyecto”, de manera de llevar un diseño 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 que guiarán cada entrega parcial darán la línea general de la funcionalidad que deberá implementar, pero **sin entrar en mayores detalles ni puntos específicos** de tal manera que usted, junto a su grupo, demuestren su capacidad de trabajo en equipo y resolución de problemas.

Salvo excepciones, condiciones y restricciones, que se señalarán oportunamente en cada enunciado de entrega, usted tendrá total libertad en cuanto a la interfaz y la implementación de su trabajo.

Para el desarrollo del proyecto, deberá formar un grupo (que se mantendrá para todas las entregas) de 2 ó 3 personas.

Para inscribir los grupos se creará un formulario en Google Forms donde podrán registrarlos.

Quienes no tengan grupo, al final del periodo de inscripción, serán asignados aleatoriamente a quienes no tengan grupo. Éstos grupos serán de 3 personas.



Marco general de las entregas

El resumen del proyecto en una línea:

Deberán construir una aplicación Web.

La aplicación web a construir podrá ser (1) un juego por turnos (con más de dos participantes) o bien (2) un simulador de cadena de producción. Su grupo podrá elegir el tipo de proyecto que decida realizar.

Reglas y características generales del proyecto de tipo juego

El juego a realizar será propuesto por cada grupo. Las propuestas serán evaluadas por el equipo docente y se acordará entre ambas partes (grupo y ayudante) sus características finales, esto será parte de la retroalimentación de la entrega 0.

El juego, en la entrega final al menos:

1. Deberá tener registro y autenticación de usuarios (login y password)
2. Deberá usar una base de datos para almacenar información relevante
3. Deberá ser un juego basado en turnos, es decir, una persona realiza todas sus jugadas y luego esperará a que el otro jugador la realice (por ejemplo como en el juego Ataque). Es posible también que todos los jugadores hagan sus jugadas y las entreguen al servidor el cual, una vez que reciba TODAS las jugadas (o se cumpla un tiempo de envío), para que el servidor resuelva combinando simultáneamente todas las jugadas.
4. El lado cliente debe tener una interfaz gráfica atractiva, completa y que permita realizar múltiples acciones (como seleccionar, drag and drop, u otras que el diseño permita. No tiene que tener todas las acciones, pero si se espera que sea una interfaz no-trivial)
5. El lado cliente debe presentar un área de juego (tablero, mapa, u otro) **compuesto de hexágonos**
6. Se deben manejar diversos perfiles de usuarios y acciones para que estos puedan realizar acciones acorde a su perfil. Por ejemplo, perfil jugador, perfil administrador, perfil superusuario, perfil “moderador” de juego, etc.
7. Debe tener un proceso (estados) que se manejen por base de datos. Por ejemplo, estado de los turnos, turnos pasados por jugador, indicar qué jugador posee el turno actual, etc.



8. Debe tener páginas estáticas (HTML y CSS) que entreguen información como instrucciones, ayuda, etc.
9. Debe tener páginas dinámicas, adicionales a las estáticas, que varíe su contenido dependiendo de las acciones que el usuario realice
10. Debe tener un panel que permita ver resumen de acciones hechas por usuario (un análogo sería una aplicación de banco que muestre transacciones realizadas en un periodo fijo de tiempo)
11. Debe manejar seguridad (encriptación) de la información, cuando corresponda
12. La implementación y desarrollo debe proveer un set de pruebas automatizados
13. Debe tener un panel de administración que permita auditar las acciones realizadas por los distintos usuarios y otras acciones acordes a este perfil, como por ejemplo dar de baja a usuarios, suprimir juegos u otras

Reglas y características generales del proyecto de de cadena de producción

La cadena de producción a seguir se podrá elegir entre cuatro tipos de cadena de producción. El diseño de su solución será evaluada por el equipo docente y se acordará entre ambas partes (grupo y ayudante) sus características finales, esto será parte de la retroalimentación de la entrega 0.

Las cadenas productivas son todas seleccionadas desde:

<https://www.dnp.gov.co/programas/desarrollo-empresarial/Paginas/analisis-cadenas-productivas.aspx>

Estas son (recuerde, debe elegir sola una de entre estas):

<https://colaboracion.dnp.gov.co/CDT/Desarrollo%20Empresarial/Tabaco.pdf>

<https://colaboracion.dnp.gov.co/CDT/Desarrollo%20Empresarial/Azucar.pdf>

<https://colaboracion.dnp.gov.co/CDT/Desarrollo%20Empresarial/Caucho.pdf>

<https://colaboracion.dnp.gov.co/CDT/Desarrollo%20Empresarial/Ceramica.pdf>



La aplicación, en su entrega final al menos:

1. Deberá tener registro y autenticación de usuarios (login y password)
2. Deberá usar una base de datos para almacenar información relevante
3. Deberá tener un diseño de tal manera que **cumpla con los flujos presentados en el documento de la cadena productiva** elegida y permitir SIMULAR dicha cadena productiva variando ya sea la cantidad de insumos o bien la cantidad de producto final deseado
4. Deberán seleccionar, desde el documento, otra característica (una adicional) a simular más allá de lo presentado en este documento, por ejemplo simulación de costos, empleos generados, etc.
5. El lado cliente debe tener una interfaz gráfica atractiva, completa y que permita realizar múltiples acciones (como seleccionar, drag and drop, u otras que el diseño permita. No tiene que tener todas las acciones, pero si se espera que sea una interfaz no-trivial)
6. El lado cliente, en su visualización, debe presentar tablero de control que muestre el flujo de a simular y que permita gráficamente cambiar cantidades de insumo y ver sus efectos en los productos intermedios y finales. El tablero también deberá permitir ingresar cantidades de producto final y ver en la simulación los insumos requeridos y su comportamiento
7. Se deben manejar diversos perfiles de usuarios y acciones para que estos puedan realizar acciones acorde a su perfil. Por ejemplo, perfil planificador, perfil administrador, perfil superusuario, etc.
8. Debe tener un proceso (estados) que se manejen por base de datos. Por ejemplo, simulaciones realizadas por tipo de usuario, algún estado de la simulación como aceptada, rechazada, en revisión, etc. Dependiendo de su implementación
9. Debe tener páginas estáticas (HTML y CSS) que entreguen información como instrucciones, ayuda, etc.
10. Debe tener páginas dinámicas, adicionales a las estáticas, que varíe su contenido dependiendo de las acciones que el usuario realice
11. Debe tener un panel que permita ver resumen de acciones hechas por usuario (un análogo sería una aplicación de banco que muestre transacciones realizadas en un periodo fijo de tiempo)
12. Debe manejar seguridad (encriptación) de la información, cuando corresponda
13. La implementación y desarrollo debe proveer un set de pruebas automatizados



14. Debe tener un panel de administración que permita auditar las acciones realizadas por los distintos usuarios y otras acciones acordes a este perfil, como por ejemplo dar de baja a usuarios, suprimir simulaciones, cambiar de perfiles a los usuarios u otras que ustedes consideren

Indicaciones generales a ambos tipos de proyecto

1. Para alojar sus aplicaciones en Internet, la plataforma soportada y recomendada será [Heroku](#) y Netlify. Si desean utilizar otro proveedor, pueden hacerlo, pero deberán justificar y documentar su elección. Noten que las cápsulas, guías y apoyo de los ayudantes será para las plataformas señaladas.
 2. Recomendamos, para usuarios Windows, utilizar el sistema operativo Ubuntu montado sobre una máquina virtual Virtualbox (<https://www.virtualbox.org/>) en vez de utilizar wsl.
 3. El código deberá entregarse en GitHub. Les indicaremos, una vez oficializados los grupos, cuáles son los repositorios asignados por grupo. Es importante que todos los integrantes utilicen el GitHub de manera colaborativa. **SE REVISARÁ EL USO DE GITHUB PARA EVALUAR SI ALGUIEN TRABAJÓ O NO.**
 4. El resultado final esperado es que ustedes realicen el despliegue real de su aplicación web (juego o cadena de producción) en modo producción en la nube y que ésta quede disponible para ser utilizada por cualquier usuario de Internet.
 5. Para evaluar cada entrega, **todos** los miembros del equipo tendrán una reunión de corrección con el ayudante que se les haya asignado. Hay que aclarar que es RESPONSABILIDAD DEL GRUPO el contactar y agendar la reunión con el ayudante.
-



6. El avance logrado en cada entrega se evaluará en base a lo mostrado, cuando corresponda, en el repositorio Github en la fecha límite de la entrega. Para entregas más avanzadas se mirará lo entregado en Heroku/Netlify y se harán pruebas funcionales, es decir evaluación si la aplicación hace o no lo que debiese estar haciendo. Recordar que Github se utilizará como una medida de trabajo de cada integrante del grupo. Si el ayudante detecta que uno o más integrantes del grupo tienen baja interacción en su repositorio de Github, puede descontar a esos alumnos un porcentaje de la nota del grupo, todo esto consta en el programa del curso.
7. En el caso de que un grupo no entregue nada, la nota será de un 1.0.
8. Para asegurar una participación equitativa y justa de los integrantes del grupo se implementará una evaluación de pares. Los detalles y funcionamiento de la evaluación de pares se entregarán en documento aparte.

Condiciones y restricciones

1. La aplicación debe desarrollarse usando **al menos**: Node.js, Koa, HTML, CSS, Javascript y React.
2. Para la construcción de la aplicación pueden usar JQuery si lo estiman pertinente.
3. Para base de datos **deberán** usar Postgres y para conexión a la base de datos pueden utilizar pg, knex o un ORM como sequelize
4. Pueden usar también EJS, SACC/SCSS, less, stylus u otros preprocesadores de CSS
5. **No está** permitido el uso de lenguajes que compilen a JavaScript, como CoffeeScript o TypeScript. Tampoco podrán usar bibliotecas de componentes HTML/CSS/JS como Bootstrap o Zurb Foundation.
6. ***Cualquier package, framework o biblioteca adicional que quieran usar debe ser aprobado por el equipo docente.***

Recomendaciones

1. Diseñen muy bien las reglas de negocio de su aplicación, el entorno, los estados, la información requerida y su modelo de almacenamiento.



2. Planifiquen el trabajo para que les permita la colaboración entre los integrantes del equipo.
3. Pregunten y consulten, **usen los canales dispuestos por el cuerpo docente para hacer consultas**, colaboren entre ustedes (**NO COPIEN**) los ayudantes están para apoyarlos.
4. **Trabajen con tiempo**, no esperen a último momento para despejar dudas.
5. Comiencen con una interfaz de usuario (UI) simple y minimalista; no es necesario que su aplicación se vea "bonita" desde el comienzo
6. Para la primera entrega deberán crear maquetas o prototipos (*mockups*) de su aplicación. La idea es que las utilicen en las entregas siguientes así que preocúpense de diseñarlos adecuadamente. Los detalles de cada entrega se darán en su enunciado específico, este es un marco general para las entregas.
7. Siempre podrán, justificadamente, cambiar alguna regla que ustedes hayan definido para su aplicación o bien mejorar algún aspecto de la misma, de la interfaz, modelo de datos, funcionalidad, etc.