



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2113 — Diseño Detallado de Software

Proyecto Semestral - Entrega 2

Entrega

- **Fecha y hora:** Martes 26 de abril de 2022, 20:00
- **Lugar:** Repositorio personal asignado para esta entrega en GitHub

Objetivos

- Realizar el primer prototipo de una solución de acuerdo a ciertas especificaciones
- Implementar una serie de requerimientos siguiendo buenas prácticas de desarrollo

1. Descripción

Para esta entrega, cada equipo de estudiantes deberá coordinar el diseño y prototipado de tres (3) partes de un sistema de software. Cada integrante del equipo deberá realizar un mock-up de un conjunto acotado de funcionalidades de la aplicación, asociadas exclusivamente al *front-end*. El foco de esta entrega se encuentra en el diseño de la capa de presentación (*front-end*) más que en funcionalidad.

Debido a que cada *stack* de tecnologías está asociado a un tema específico, la división de funcionalidades que cada integrante de un equipo deberá diseñar e implementar se encuentra en el apéndice de este enunciado:

- [Rails](#) (*back-end*) y [React](#) (*front-end*): **Especificaciones para Rails & React**
- Rails (*back-end*) y [Unity](#) (*front-end*): **Especificaciones para Rails & Unity**
- Rails (*back-end*) y HTML + CSS puro (*front-end*): **Especificaciones para Rails & HTML + CSS puro**

En específico, en esta entrega lo que se busca es que cada estudiante realice los mock-ups correspondientes a una de las funcionalidades de la aplicación y que además implemente una primera versión de esa funcionalidad. No se espera que se implemente el *back-end* correspondiente. Quienes forman un equipo deberán trabajar en funcionalidades diferentes, con el objetivo de que en futuras entregas puedan integrar estas partes.

Para realizar preguntas técnicas, sobre requisitos o sobre el proyecto o entrega en general se deberá utilizar el repositorio oficial del curso en GitHub: [IIC2113/Syllabus](#).

2. Entregables

Cada integrante del equipo deberá completar los siguientes entregables:

- Mock-ups correspondientes a la funcionalidad elegida
- Implementación del *front-end* correspondiente a la funcionalidad elegida

3. Distribución de Puntaje

- (30 %) Mock-up de la funcionalidad elegida
- (70 %) Código correspondiente a la presentación de la funcionalidad

4. Notas

- Si bien esta entrega requiere coordinación dentro de cada equipo, tanto la entrega como su corrección y nota asignada son individuales
- Se espera que cada equipo se ponga en contacto con su ayudante durante la primera semana de la entrega para aclarar lo que se espera del proyecto en general y de la entrega en particular, además de aclarar detalles de la entrega que no estén claros

Apéndices

A. Especificaciones para Rails & React



Dado que se trabaja con React, antes de partir se sugiere que planifiquen de manera grupal la organización de los componentes y definan la manera en que estarán estructurados los datos provenientes del *back-end*. Para esto pueden basarse en sus diagramas de la E1. La estructura de los componentes y datos no necesariamente tiene que ser definitiva para todo el proyecto, aunque mantener cierto grado de consistencia les podría ser útil para las futuras entregas.

Integrante 1

Lista de productos Se debe poder visualizar la lista de productos disponibles y su información asociada. La manera de mostrar los productos queda a criterio de quién lo realiza, sin embargo es importante que se muestre un mínimo de información. Los atributos mínimos que se deben mostrar por producto son:

- Nombre
- Precio
- Marca
- Categoría

Categorías de productos Cada producto debe estar asociado a una categoría. Las categorías son de libre elección y gestionadas por el rol de administrador. Debe existir una vista donde se puedan crear categorías (para el administrador) y además asociarle a los productos una categoría. Finalmente, se debe implementar en esa vista un filtro por categoría, donde se muestre todos los productos de la categoría. Los atributos a mostrar son los mismos que para los especificados en la parte de lista de productos.

Integrante 2

Carrito de compras Se debe poder agregar productos a un carrito de compras, pudiendo visualizar la lista de productos añadidos al carrito, el valor de cada uno y el total de todos los elementos dentro de este. Se debe poder modificar la cantidad de un determinado producto dentro del carrito o eliminarlo completamente de este.

Integrante 3

Historial de compras Se debe poder visualizar el historial de compras realizadas por el usuario. Se debe poder ver pedidos por separado por fecha y los productos específicos de cada uno, con el valor y cantidad de cada producto, el valor total del pedido y si se aplicó algún descuento por algún cupón.

Cupones (Promociones) Deben existir cupones de descuento que se pueden aplicar sobre los productos. Estos pueden ser de dos maneras. Existen cupones que pueden ser aplicados sobre todos los productos, independiente a su categoría (sobre el valor total del carrito). Pero también existirán aquellos que solo podrán aplicar descuentos sobre alguna categoría en específico, sin afectar el precio de aquellos productos que no se encuentran en dicha categoría. Para esta entrega se pide generar una vista de cupones que muestre cuáles ya han sido gastados.

B. Especificaciones para Rails & Unity



Para este proyecto se deberá desarrollar un centro de videojuegos (~~INSERTE NOMBRE AQUÍ~~), donde los usuarios podrán acceder a diferentes tipos de juegos y ganar monedas o logros para aumentar su nivel de jugador en el centro de videojuegos.

Tomando en cuenta los detalles mencionados en la entrega anterior, para esta entrega cada uno de los integrantes deberá desarrollar uno de los 5 tipos de juegos mencionados. No se pueden repetir los tipos de juegos, por lo que se deben poner de acuerdo como equipo para que cada uno realice un tipo de juego diferente. Para más detalles sobre las características de cada juego, revisar el enunciado de la entrega 1.

Para esta entrega se espera que cada uno de los juegos sea jugable, pero no tiene que estar terminado por completo. Por ejemplo, en el caso de que un juego tenga varios niveles, para esta entrega será suficiente que se pueda jugar un solo nivel. A continuación, se detallan los mínimos que se esperan en cada uno de los tipos de juegos:

■ Atrapar objetos:

- 1 nivel de juego
- Personaje con sus sprites y movimientos
- Definir el tipo de objetos a atrapar junto a sus sprites
- Permitir atrapar un objeto
- Mínimo una estadística durante el juego (tiempo, objetos atrapados, puntaje, etc.)

No es necesario que los objetos aparezcan aleatoriamente, ni que esté definida la ponderación o el puntaje de cada objeto al ser atrapado.

■ Atacar enemigos:

- 1 nivel de juego
- Definir el tipo de enemigo o enemigos en caso de ser varios
- Personaje y enemigo(s) con sus sprites y movimientos
- Permitir atacar al enemigo y eventualmente derrotarlo, o derrotarlo directamente si es de varios enemigos
- Mínimo una estadística durante el juego (tiempo, enemigos eliminados, puntaje, etc.)

■ Evadir obstáculos:

- 1 nivel de juego o una distancia finita en caso de que sea un nivel infinito
- Personaje con sus sprites y movimientos
- Definir el o los tipos de obstáculos a mostrar

- Permitir al jugador evadir los obstáculos
- Mínimo una estadística durante el juego (tiempo, distancia recorrida, objetos evadidos, puntaje, etc.)

■ **Laberinto:**

- 2 niveles de juego o laberintos
- Personaje junto a sus sprites y movimientos
- Permitir colisiones con los obstáculos o murallas
- Mínimo una estadística durante el juego (tiempo, niveles superados, porcentaje de avance, puntaje, etc.)

■ **Salto infinito:**

- 1 nivel de juego o una distancia finita en caso de que sea un nivel infinito
- Personaje junto a sus sprites y movimientos
- Plataformas junto a sus sprites
- Aparición de plataformas de forma aleatoria
- Monedas para atrapar (pueden no ser aleatorias por ahora)
- Mínimo una estadística durante el juego (tiempo, altura máxima, altura actual, puntaje, etc.)

En caso de tener alguna duda con respecto a si se puede no implementar alguna funcionalidad para esta entrega, consultar a su ayudante de grupo. En caso de no implementar alguna funcionalidad que no estaba clara sin preguntar a su ayudante se considerará como que no se realizó.

C. Especificaciones para Rails & HTML + CSS puro



Para esta entrega se les pedirá que desarrollen funcionalidades relacionadas a la interfaz de usuario del planner de cursos que diseñaron en la entrega pasada.

Las funcionalidades requeridas deberán ser realizadas haciendo uso de HTML, CSS y Javascript, para entregar interactividad a su aplicación. Se dejará a elección de cada grupo la posibilidad de trabajar con JQuery o Javascript Vanilla.

Está prohibido el uso de cualquier *framework* CSS (Bootstrap, Bulma, etc.).

En esta entrega deberán desarrollar como mínimo las siguientes funcionalidades:

Funcionalidad 1: Usuario y página de inicio

Para esta funcionalidad, la implementación puede ser realizada en formato de parallax o como navbar, diseñando cada vista por separado. Las vistas a desarrollar son las siguientes:

- Landing Page con información de la aplicación
 - ¿Qué es?
 - Instrucciones de uso
 - Carreras disponibles
- Formularios de registro e inicio de sesión (se espera el uso de *Modals*)
- Validación de requisitos de contraseña
- Botón para poder visualizar la contraseña ingresada
- Manejo de errores en formularios
- Sección de foro, con posibilidad de dar like a comentarios, responder, etc.

Funcionalidad 2: Vistas de alumno

- Visualizar cursos inscritos por semestre (se espera el uso de tablas)
- Hover para detalles de cursos (requisitos, convalidaciones)
- Información de estado del curso con colores (aprobados en verde, por ejemplo)
- Perfil de usuario. Se debe poder editar información de este. Se debe poder subir una foto de perfil.

Funcionalidad 3: Vista de administrador

- Vistas relacionadas a la creación de carreras, asignación de mayor y/o minor
- Filtrar listado de cursos por carrera
- Vistas relacionadas a la creación de cursos con los campos correspondientes
- Buscador de usuarios. Se debe poder filtrar por nombre, correo electrónico y carrera. Se espera que al seleccionar un usuario se muestre información de este