



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

IIC2113 — Diseño Detallado de Software

# Proyecto Semestral - Entrega 4

## Entrega

- **Fecha y hora:** Viernes 24 de junio de 2022, 13:00
- **Lugar:** Repositorio grupal en la organización del curso en GitHub

## Objetivos

- Implementar una serie de requerimientos siguiendo buenas prácticas de desarrollo
- Aplicar los principios fundamentales, SOLID, CLEAN y otros temas vistos en la parte teórica del curso
- Comprender las implicaciones del diseño en una aplicación ya sea en base a las dificultades o beneficios encontrados en el proceso.

## 1. Descripción

Para esta entrega, cada equipo de estudiantes deberá trabajar en la satisfacción de tres nuevos requerimientos, uno de los cuales es no funcional y los restantes son funcionales.

El requisito no funcional es la incorporación de *tests*. El objetivo es lograr un *coverage* de **al menos 70 %** en el *back-end* de la aplicación. Sabemos que la mejor manera de realizar *tests*, sin que sea algo agobiante, es escribirlos a medida que se desarrollan funcionalidades, por lo que si ya cumplen con el porcentaje de *coverage* esperado, no es necesario hacer nuevos.

Por otra parte, el primer requisito funcional es consumir una API externa. Los datos a consumir son de libre elección, pero deben ser aprobados y confirmados por quien sea ayudante de cada grupo. A continuación, dejamos un ejemplo de APIs que podrían consumir según el tipo de proyecto:

- **Rails y Unity:** Consumir una API que retorne nombres aleatorios para jugar. ([link](#))
- **Rails y HTML+CSS:** Consumir información desde BuscaCursos para mostrar información de los cursos. ([link](#))
- **Rails y React:** Consumir una API que entregue información adicional de productos. ([link](#))

Finalmente, además de consumir una API deberán exponer una serie de *endpoints* para ser consumidos que entreguen información en formato JSON. La interfaz específica la podrán diseñar libremente, pero deberá estar bien documentada en el repositorio (ver sección de entregables). Nuevamente, el requerimiento dependerá del tipo de proyecto:

- **Rails y Unity:** Exponer una API a través de la cual se pueda consultar los 10 mejores puntajes del centro de juegos de cada juego, considerando nombre de usuario, puntaje y juego.

- **Rails y HTML+CSS:** Exponer una API a través de la cual se pueda consultar los detalles del perfil de usuario e interacciones en el foro.
- **Rails y React:** Exponer una API a través de la cual se puedan consultar los cupones disponibles.

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

Como equipo deberán entregar, en uno o más repositorios:

- Implementación del *front-end* y del *back-end* de la solución.
- Evidencia del uso de Pull Requests y *code review* entre integrantes del equipo para cada aporte en código (se utilizará GitHub para evaluar este punto).
- Documentación en base a diagramas UML actualizados con el estado actual de la solución.
- **Nuevo.** Documentación de API expuesta en el archivo API.md de cada repositorio, indicando las rutas disponibles, la información que entregan, el formato de la respuesta y cualquier información adicional necesaria para consumir los recursos.
- **Nuevo.** Tests con un *coverage* de **al menos 70 %** en el *back-end* de la aplicación

## 3. Distribución de Puntaje

- (25 %) El código desarrollado presenta uso de buenas prácticas de desarrollo.
- (20 %) Tests con un *coverage* de **al menos 70 %** en el *back-end* de la aplicación.
- (20 %) El código consume una API cumpliendo con lo solicitado para cada proyecto.
- (15 %) El código expone una API cumpliendo con lo solicitado para cada proyecto.
- (5 %) Documentación de API expuesta en el archivo API.md cumple con lo solicitado y refleja lo realizado.
- (10 %) Se utilizó durante el desarrollo una metodología de Pull Requests y *code review* en GitHub.
- (5 %) Los diagramas UML de la documentación reflejan el estado actual de la aplicación.

## 4. Notas

- 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. **El no reunirse con su ayudante durante la primera semana de la entrega significará un descuento de 5 décimas en la evaluación.**
- Cada grupo deberá solicitar (y justificar) los repositorios que quiera utilizar. No hay límite mínimo o máxima sobre cuántos repositorios podrán asignarse a un equipo, pero **se espera que utilicen su criterio y tomen decisiones de cómo organizar su código.**