



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIAS DE COMPUTACIÓN  
IIC2143 – INGENIERÍA DE SOFTWARE (II/2021)

---

# Entrega 1

PROYECTO SEMESTRAL  
ENTREGA: 8 DE OCTUBRE, 23:59.

---

## 1. Objetivos

- Definir modelo de la plataforma mediante un diagrama E/R.
- Comenzar el desarrollo de funcionalidades.

## 2. Diagrama E/R

Antes de comenzar con el desarrollo de funcionalidades, una práctica indispensable es realizar un modelo de datos a partir de sus relatos de usuario. Esto les proveerá una visión general de los datos que debe manejar su aplicación antes de empezar y previene hacer grandes cambios durante el desarrollo, debido a detalles no considerados. Para esto, deben generar un diagrama E/R que muestre el modelo de datos considerado.

## 3. Funcionalidades y requisitos mínimos de desarrollo

Dada la planificación de Sprint acordada durante el *Sprint Review*, deben adherirse lo más posible a ella en cuanto al desarrollo de funcionalidades. También, deben mantener los requisitos mínimos de desarrollo. Es decir, su trabajo debe:

- Reflejarse en su tablero de *Trello*, donde se plasme un correcto flujo de tarjetas por cada funcionalidad trabajada.
- Seguir la filosofía de *Gitflow* para gestionar su repositorio.
- Respetar su guía de estilo revisada por *Rubocop*.
- Escribir tests con *RSpec* y no tener tests fallados.
- Montar en producción su aplicación a *Heroku*.
- Tener el código actualizado de su aplicación en la rama *main* de su repositorio en Github.

## 4. RSpec + SimpleCov

Para esta entrega debes escribir tests para tu aplicación utilizando la gema *RSpec*. En esta ocasión, habrán dos requisitos: contar con **al menos 15 tests unitarios** y contar con **un cierto porcentaje de cobertura**. El porcentaje de cobertura en testing representa el porcentaje de líneas de código que son probadas por la suite de tests de la aplicación. Para esta entrega, deberás tener **al menos un 40 % de cobertura** en *Modelos*, *Controladores* y *Helpers* respectivamente. Para calcular la cobertura, deberás instalar una gema llamada *SimpleCov*. Los pasos para poder utilizar esta gema de la manera más simple posible pueden ser encontrados en el repositorio de GitHub del curso.

## 5. Bonus

Para esta entrega su *product owner* le pedirá comenzar con el desarrollo de los CRUDs de la aplicación. Para aquellos grupos que, dentro de la cantidad que tenga que desarrollar, haga dos de estos sin usar *scaffold*, podrá obtener un máximo de dos décimas extra en su nota de la entrega (un punto por cada CRUD completo sin usar *scaffold*).

## 6. Sprint Review

Tras esta entrega, nuevamente deben agendar su reunión de *Sprint Review* con su *product owner*. Esta debe realizarse durante los 3 días hábiles siguientes a la fecha de entrega.

## 7. Entrega

En su repositorio de *GitHub* asignado deben alojar el código de su aplicación. Incluyan un archivo `README.md` con la dirección de su aplicación montada en *Heroku*. La fecha de entrega es **8 de Octubre, 23:59**. Se revisará el último *commit* en su rama *main* antes de dicho plazo.

## 8. Distribución del puntaje

La entrega consta de un total de 23 puntos distribuidos de la siguiente manera:

- Diagrama E/R: 4 ptos
- Aplicación en Heroku: 3 ptos
- Trello: 2 ptos
- Rubocop: 2 ptos
- Gitflow: 2 ptos
- RSpec: 3 ptos
- Features logradas: 4 ptos
- Calidad de features: 3 ptos

## 9. Política de integridad académica

Los alumnos de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile deben mantener un comportamiento acorde a la Declaración de Principios de la Universidad. En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada alumno conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Docencia de la Escuela de Ingeniería en el SIDING.

Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica. Todo trabajo presentado por un alumno para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno, sin apoyo en material de terceros. Por “trabajo” se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros. Si un alumno copia un trabajo, obtendrá nota final 1.1 en el curso y se solicitará a la Dirección de Pregrado de la Escuela de Ingeniería que no le permita retirar el curso de la carga académica semestral. Por “copia” se entiende incluir en el trabajo presentado como propio partes hechas por otra persona. Obviamente, está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la referencia correspondiente. Lo anterior se entiende como complemento al Reglamento del Alumno de la Pontificia Universidad Católica de Chile. Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.