

# IIC2143

## Ingeniería de Software

Semestre 1 2018  
Prof. Jaime Navón



# Algo sobre su profe ...

- Lo que más me gusta es aprender cosas nuevas
- Lo segundo que más me gusta es enseñar lo que aprendo
- He visto nacer el PC, la Web, el Smartphone y la Inteligencia Artificial
- Veo a los alumnos como hijos: asumo responsabilidad de educarlos como ingenieros, ser duro cuando hay que serlo pero estar siempre ahí para ayudar.
- Me gusta la música (amplio espectro). Mi banda favorita es R.E.M.

# Metodología

- Clases Expositivas (dos a la semana)
- Clase Práctica Plataforma Web (una a la semana)
- Desarrollo de un Proyecto (Grupos de 3)
- Lectura individual complementaria

# ¿Qué aprenderemos?

- Proceso de Desarrollo Agil
- Levantar requisitos funcionales
- Dibujar gráficamente el software
- Modelar un problema con objetos
- Diseñar soluciones que puedan cambiarse
- Conocer sobre arquitecturas posibles
- Aprender como se asegura la calidad del software
- Aprender a construir una aplicación Web (SaaS)

# Desarrollo de Software

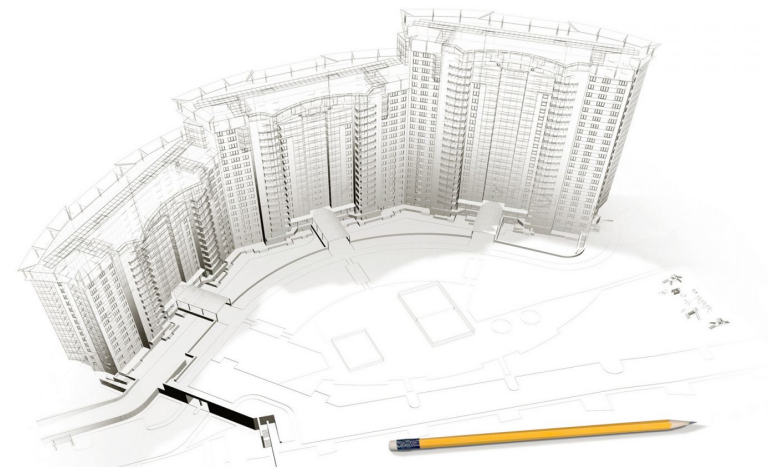
- Introducción a la Programación



- Programación Avanzada



- Ingeniero de Software



# Objetivos del Proyecto

- Aprender los aspectos fundamentales que involucra el desarrollo de una aplicación Web
- Aterrizar algunos de los conceptos aprendidos en clases a la experiencia personal real
  - proceso ágil
  - levantamiento de requisitos
  - diseño

# Material de Clases

- Las láminas usadas en clase serán publicadas antes de cada clase pero Ud deberá completarlas con lo que se discuta en la clase
- El profesor publicará además con anticipación lectura SOBRE tema de cada clase
- Esto les permitirá sacar mayor partido de la clase y agregar notas sobre las láminas

# Por qué un proyecto Web

- Gran parte del software que usas en tu computador es presentado como servicio (SaaS)
  - gmail
  - facebook
  - google docs
  - dropbox
- Interfaz Web de la mayoría de las aplicaciones comerciales



# Ventajas de SaaS

- Clientes no requieren correr un SO específico o instalar una aplicación dada
- Clientes no tienen responsabilidad de backup de sus datos
- Interacción natural entre clientes
- Puede ofrecerse un volumen de datos gigantesco que está siempre actualizado
- No es necesario distribuir binarios del software
- Actualización frecuente de la aplicación

# Aprender Ruby/Rails

- El esfuerzo mayor recaerá en ustedes mismos
  - Michael Hartl Ruby on Rails Tutorial and Book
  - <https://www.railstutorial.org/book>
- Algunas clases iniciales de apoyo
- Clases prácticas en horario de ayudantía (altamente recomendable)

# Recomendaciones

- Tratar de venir a clases !!
- Evitar usar computadores en clases (excepto en las clases prácticas)
- Tomar apuntes
- Preguntar y comentar mucho (habrá bonos)
- Dedicar esfuerzo tanto al proyecto como al material de clases

