

# Sistemas Distribuidos e Internet

## Presentación de la Asignatura

-----

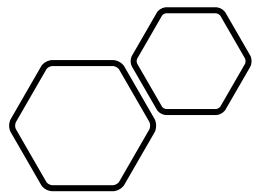
Miguel Sánchez Santillán

Edward Rolando Núñez Valdez

Jordán Pascual Espada

Ruben Luque Lodeiro

Enrique de la Cal Marín



# Full stack developer

- Un FS developer es un desarrollador de amplio espectro: Front End, Back End, DB, Dev. Ops, Mobile Apps....



[Video Enlace](#)





# Objetivos

---

## Tecnologías

Por un lado, se introducirá al alumno a las tecnologías más representativas en la industria.

## Ingeniería

Por otro, se abordarán las formas más recomendables de aplicar estas tecnologías, principalmente mediante la revisión de patrones arquitectónicos y las consideradas “buenas prácticas de desarrollo”.

# Contenidos

## Desarrollo web clásico

- **[JEE]** Introducción a las tecnologías y aplicaciones Web
- **[Patrones]** Ingeniería para el desarrollo Web
- **[Spring Boot]** Desarrollo clásico de aplicaciones Web
- **[Selenium]** Prueba sistemática de aplicaciones Web

## Desarrollo web ágil

- **[Node.js]** Desarrollo ágil de aplicaciones Web
- **[REST/SOAP]** Desarrollo de aplicaciones Web basado en servicios
- Seguridad y legislación en desarrollo

# Laboratorios

Se publicarán con una semana de antelación un **guión** para ser desarrollado de forma autónoma antes de la clase (GitHub)

En clase de laboratorio se aclararán dudas del **guión** y se realizarán Ejercicios adicionales (GitHub)

Sesiones (12-13)

- Desarrollo web clásico
  - JEE (2)
  - Spring(3)
  - Selenium
  - Libre
- Desarrollo web ágil
  - Node.Js(3)
  - Rest
  - SOAP
  - Libre

# Proyectos semanales - individuales

## Trabajo individual

## Una entrega por semana + una entrega en clase

- Entrega en GitHub
- IDGIT Individual
- Commits de formato fijo

Nota: Subir el código a GitHub en este punto. Incluir el siguiente Commit Message ->

`"SDI-IDGIT-1.0-Create servlet project."`

OJO: sustituir IDGIT por tu número asignado (p.e. 2021-101):

`"SDI-2223-101-1.0-Create servlet project."`

*(No olvides incluir los guiones y NO incluyas BLANCOS)*

# Proyectos de entrega en equipo

**Trabajo en equipos de 3-5 personas**

**Dos proyectos de entrega:**

- Desarrollo con Spring Boot/Selenium
- Desarrollo con Node.js/Servicios Web/Selenium
- Entregas parciales en el CV

# Seguimiento del desarrollo mediante GIT/GITHUB

Tanto las prácticas de **Laboratorio** como las de **Entrega** se seguirán mediante GIT/GITHUB

## Proyectos semanales de Laboratorio

- Commits en los puntos indicados en los enunciados.
- Al final de cada práctica se deberá marcar el proyecto con la etiqueta que se indique en el propio enunciado (ej. "sdi-springboot-p1")

## Prácticas de Entrega

- Se recomiendan commits frecuentes

## Parámetros Git/GitHub

- Usuario: Cuenta GitHub personal del alumno
- Repositorios (ej. <https://github.com/apellidosNombre/repositorio.git>):
  - Cada alumno dispondrá de un ID numérico=**x** publicado en el CV que será el que se usará para identificar sus repositorios git.
  - En cada grupo de guiones se indicará el repositorio a emplear.
    - **sdi-x-lab-y.git (JEE, Spring, NodeJs),**
    - Y dentro de cada repositorio se incluirá una **etiqueta al finalizar** cada práctica (ej. "sdi-spring-p1").
  - Uno por práctica de entrega:
    - **sdi-x-entrega-sdi1.git (Spring)**
    - **sdi-x-entrega-sdi2.git (Node.Js + Servicios Web)**
- Agregar como colaborador para cada repositorio a la cuenta **sdigitHubuniovi**



# Pruebas de autoría y criterios de exclusión

**Todos** los alumnos que entreguen una práctica deberán presentarse a una **prueba de autoría de realización individual**.

Para que un alumno quede **excluido** de la realización de esa prueba deberán cumplir con los siguientes criterios:

- Aprobar parcial de teoría correspondiente a la entrega práctica
- Entregar un 90% de los commits en tiempo y forma correspondientes a los guiones de laboratorio **(Verificado en Github)**
- Realizar al menos el 50% de los ejercicios adicionales propuestos en clases en cada práctica de laboratorio **(Verificado en Github)**

# Seminarios

Monográficos en el contexto de la asignatura pero independientes de la teoría, aunque estos contenidos pueden entrar en los exámenes de teoría

## Temas

- Introducción a HTTP
- Repaso para Parcial1
- Modelos de Negocio en la Web (... , SEO/SEM
- Jhipster
- Blockchain (Aplicación)
- TCP/UDP & Web Sockets
- Repaso para Parcial2

# Evaluación General

## Ordinaria

- Dos parciales de teoría (Nteo1, Nteo2)
  - $N_{teo} = N_{teo1} * 0,5 + N_{teo2} * 0,5$
- Dos proyectos de entrega (Nprac1, Nprac2)
  - $N_{prac} = N_{prac1} * 0,5 + N_{prac2} * 0,5$
- $N_{teo1}, N_{teo2}, N_{prac1}, N_{prac2} \geq 4$
- $N_{teo}, N_{prac} \geq 4,0$
- **Nota final** =  $0,4 * N_{teo} + 0,6 * N_{prac}$
- Asistencia mínima a clase de laboratorio: 80%
- Una prueba de autoría por entrega (Ver criterios de exclusión de la misma)

## Extraordinaria/Diferenciada:

- Dos parciales de teoría
- Dos proyectos de entrega
- Prueba de autoría obligatoria

Se guardan las notas aprobadas ( $N_{teo1}, N_{teo2}, N_{prac1}, N_{prac2} \geq 5$ ) hasta Enero23 (incluido).

# Evaluación Prácticas de entrega

## Nota EQUIPO

- Casos de prueba (Web TESTING) - WEBT
- Correcciones en Código - COD
- Penalizaciones - PEN
- INFORME - INFOR
- **NOTA EQUIPO = SUMAPOND(WEBT, COD, PEN , INFOR)**

## NOTA INDIVIDUAL

- Esta es la calificación válida de practicas para cada alumno.
- Coevaluación holística
- Cada compañero es evaluado por el resto (FIC)
- FIC
  - Factor Individual de Coevaluación
  - Fórmula de FIC se publicará con el enunciado de los entregables
- **NOTA INDIVIDUAL = FIC \* NOTA EQUIPO**