

# Tecnologías para el Desarrollo de Aplicaciones Empresariales sobre Internet

Pablo Sánchez

Dpto. Ingeniería Informática y Electrónica  
Universidad de Cantabria  
Santander (Cantabria, España)  
p.sanchez@unican.es



# Índice

- 1 Índice
- 2 Objetivos y Temario
- 3 Metodología
- 4 Métodos de Evaluación y Calificación
- 5 Bibliografía

# Índice

- 1 Índice
  - Horarios
- 2 Objetivos y Temario
- 3 Metodología
- 4 Métodos de Evaluación y Calificación
- 5 Bibliografía

# Horario de Clases

## Tercer Periodo

	Lunes	Martes	Miércoles	Jueves	Viernes
15:30 - 16:30		Aula 4			
16:30 - 17:30		Aula 4			
17:30 - 18:30				Aula 4	
18:30 - 19:30				Aula 4	

- Las clases de los Martes y Jueves serán indistintamente de teoría o laboratorio, según convenga al ritmo de la asignatura.

# Horario de Clases

## Cuarto Periodo

	Lunes	Martes	Miércoles	Jueves	Viernes
15:30 - 16:30		Aula 4		Aula 4	
16:30 - 17:30		Aula 4		Aula 4	
17:30 - 18:30					
18:30 - 19:30					

- Las clases de los Martes y Jueves serán indistintamente de teoría o laboratorio, según convenga al ritmo de la asignatura.

# Horario de Clases

## Cuarto Periodo

	Lunes	Martes	Miércoles	Jueves	Viernes
15:30 - 16:30		Aula 4		Aula 4	
16:30 - 17:30		Aula 4		Aula 4	
17:30 - 18:30					
18:30 - 19:30					

- Las clases de los Martes y Jueves serán indistintamente de teoría o laboratorio, según convenga al ritmo de la asignatura.

# Horario Tutorías

- A cualquier hora, preferentemente de mañana.
- Si se quiere asegurar disponibilidad, avisad con antelación.
- Se permiten tutorías tanto presenciales como telemáticas.
- Disponible a través de correo, moodle y **Discord**.

# Horario Tutorías

- A cualquier hora, preferentemente de mañana.
- Si se quiere asegurar disponibilidad, avisad con antelación.
- Se permiten tutorías tanto presenciales como telemáticas.
- Disponible a través de correo, moodle y **Discord**.



# Horario Tutorías

- A cualquier hora, preferentemente de mañana.
- Si se quiere asegurar disponibilidad, avisad con antelación.
- Se permiten tutorías tanto presenciales como telemáticas.
- Disponible a través de correo, moodle y **Discord**.

# Horario Tutorías

- A cualquier hora, preferentemente de mañana.
- Si se quiere asegurar disponibilidad, avisad con antelación.
- Se permiten tutorías tanto presenciales como telemáticas.
- Disponible a través de correo, moodle y **Discord**.

# Índice

- 1 Índice
- 2 **Objetivos y Temario**
- 3 Metodología
- 4 Métodos de Evaluación y Calificación
- 5 Bibliografía

# Índice

- 1 Índice
- 2 Objetivos y Temario
  - **Objetivos**
  - Temario
- 3 Metodología
- 4 Métodos de Evaluación y Calificación
- 5 Bibliografía

# Objetivos de la Asignatura

- 1 Comprender el concepto de aplicación empresarial.
- 2 Comprender cómo se divide una aplicación empresarial en capas.
- 3 Saber aplicar patrones para el diseño de una capa de negocio.
- 4 Saber aplicar los principios de *Domain-Driven Design*.
- 5 Comprender el funcionamiento de los patrones de persistencia.
- 6 Saber utilizar herramientas para la generación de puentes objeto-relacional.
- 7 Saber aplicar patrones para el diseño de una capa de servicio.
- 8 Saber especificar e implementar una capa de servicio REST.
- 9 Saber aplicar patrones para el diseño de una capa de presentación.
- 10 Ser capaz de utilizar los servicios, como control de accesos, proporcionados por los servidores de aplicaciones.

# Objetivos de la Asignatura

- 1 Comprender el concepto de aplicación empresarial.
- 2 Comprender cómo se divide una aplicación empresarial en capas.
- 3 Saber aplicar patrones para el diseño de una capa de negocio.
- 4 Saber aplicar los principios de *Domain-Driven Design*.
- 5 Comprender el funcionamiento de los patrones de persistencia.
- 6 Saber utilizar herramientas para la generación de puentes objeto-relacional.
- 7 Saber aplicar patrones para el diseño de una capa de servicio.
- 8 Saber especificar e implementar una capa de servicio REST.
- 9 Saber aplicar patrones para el diseño de una capa de presentación.
- 10 Ser capaz de utilizar los servicios, como control de accesos, proporcionados por los servidores de aplicaciones.

# Objetivos de la Asignatura

- 1 Comprender el concepto de aplicación empresarial.
- 2 Comprender cómo se divide una aplicación empresarial en capas.
- 3 Saber aplicar patrones para el diseño de una capa de negocio.
- 4 Saber aplicar los principios de *Domain-Driven Design*.
- 5 Comprender el funcionamiento de los patrones de persistencia.
- 6 Saber utilizar herramientas para la generación de puentes objeto-relacional.
- 7 Saber aplicar patrones para el diseño de una capa de servicio.
- 8 Saber especificar e implementar una capa de servicio REST.
- 9 Saber aplicar patrones para el diseño de una capa de presentación.
- 10 Ser capaz de utilizar los servicios, como control de accesos, proporcionados por los servidores de aplicaciones.

# Objetivos de la Asignatura

- 1 Comprender el concepto de aplicación empresarial.
- 2 Comprender cómo se divide una aplicación empresarial en capas.
- 3 Saber aplicar patrones para el diseño de una capa de negocio.
- 4 Saber aplicar los principios de *Domain-Driven Design*.
- 5 Comprender el funcionamiento de los patrones de persistencia.
- 6 Saber utilizar herramientas para la generación de puentes objeto-relacional.
- 7 Saber aplicar patrones para el diseño de una capa de servicio.
- 8 Saber especificar e implementar una capa de servicio REST.
- 9 Saber aplicar patrones para el diseño de una capa de presentación.
- 10 Ser capaz de utilizar los servicios, como control de accesos, proporcionados por los servidores de aplicaciones.



# Objetivos de la Asignatura

- 1 Comprender el concepto de aplicación empresarial.
- 2 Comprender cómo se divide una aplicación empresarial en capas.
- 3 Saber aplicar patrones para el diseño de una capa de negocio.
- 4 Saber aplicar los principios de *Domain-Driven Design*.
- 5 Comprender el funcionamiento de los patrones de persistencia.
- 6 Saber utilizar herramientas para la generación de puentes objeto-relacional.
- 7 Saber aplicar patrones para el diseño de una capa de servicio.
- 8 Saber especificar e implementar una capa de servicio REST.
- 9 Saber aplicar patrones para el diseño de una capa de presentación.
- 10 Ser capaz de utilizar los servicios, como control de accesos, proporcionados por los servidores de aplicaciones.

# Objetivos de la Asignatura

- 1 Comprender el concepto de aplicación empresarial.
- 2 Comprender cómo se divide una aplicación empresarial en capas.
- 3 Saber aplicar patrones para el diseño de una capa de negocio.
- 4 Saber aplicar los principios de *Domain-Driven Design*.
- 5 Comprender el funcionamiento de los patrones de persistencia.
- 6 Saber utilizar herramientas para la generación de puentes objeto-relacional.
- 7 Saber aplicar patrones para el diseño de una capa de servicio.
- 8 Saber especificar e implementar una capa de servicio REST.
- 9 Saber aplicar patrones para el diseño de una capa de presentación.
- 10 Ser capaz de utilizar los servicios, como control de accesos, proporcionados por los servidores de aplicaciones.

# Objetivos de la Asignatura

- 1 Comprender el concepto de aplicación empresarial.
- 2 Comprender cómo se divide una aplicación empresarial en capas.
- 3 Saber aplicar patrones para el diseño de una capa de negocio.
- 4 Saber aplicar los principios de *Domain-Driven Design*.
- 5 Comprender el funcionamiento de los patrones de persistencia.
- 6 Saber utilizar herramientas para la generación de puentes objeto-relacional.
- 7 Saber aplicar patrones para el diseño de una capa de servicio.
- 8 Saber especificar e implementar una capa de servicio REST.
- 9 Saber aplicar patrones para el diseño de una capa de presentación.
- 10 Ser capaz de utilizar los servicios, como control de accesos, proporcionados por los servidores de aplicaciones.

# Objetivos de la Asignatura

- 1 Comprender el concepto de aplicación empresarial.
- 2 Comprender cómo se divide una aplicación empresarial en capas.
- 3 Saber aplicar patrones para el diseño de una capa de negocio.
- 4 Saber aplicar los principios de *Domain-Driven Design*.
- 5 Comprender el funcionamiento de los patrones de persistencia.
- 6 Saber utilizar herramientas para la generación de puentes objeto-relacional.
- 7 Saber aplicar patrones para el diseño de una capa de servicio.
- 8 Saber especificar e implementar una capa de servicio REST.
- 9 Saber aplicar patrones para el diseño de una capa de presentación.
- 10 Ser capaz de utilizar los servicios, como control de accesos, proporcionados por los servidores de aplicaciones.

# Objetivos de la Asignatura

- 1 Comprender el concepto de aplicación empresarial.
- 2 Comprender cómo se divide una aplicación empresarial en capas.
- 3 Saber aplicar patrones para el diseño de una capa de negocio.
- 4 Saber aplicar los principios de *Domain-Driven Design*.
- 5 Comprender el funcionamiento de los patrones de persistencia.
- 6 Saber utilizar herramientas para la generación de puentes objeto-relacional.
- 7 Saber aplicar patrones para el diseño de una capa de servicio.
- 8 Saber especificar e implementar una capa de servicio REST.
- 9 Saber aplicar patrones para el diseño de una capa de presentación.
- 10 Ser capaz de utilizar los servicios, como control de accesos, proporcionados por los servidores de aplicaciones.

# Objetivos de la Asignatura

- 1 Comprender el concepto de aplicación empresarial.
- 2 Comprender cómo se divide una aplicación empresarial en capas.
- 3 Saber aplicar patrones para el diseño de una capa de negocio.
- 4 Saber aplicar los principios de *Domain-Driven Design*.
- 5 Comprender el funcionamiento de los patrones de persistencia.
- 6 Saber utilizar herramientas para la generación de puentes objeto-relacional.
- 7 Saber aplicar patrones para el diseño de una capa de servicio.
- 8 Saber especificar e implementar una capa de servicio REST.
- 9 Saber aplicar patrones para el diseño de una capa de presentación.
- 10 Ser capaz de utilizar los servicios, como control de accesos, proporcionados por los servidores de aplicaciones.

# Resultados de Aprendizaje

- 1 Elaborar modelos de dominio utilizando *POJOs*.
- 2 Clasificar elementos de dominio conforme a la metodología *Domain-Driven Design*.
- 3 Anotar un modelo de dominio para persistirlo utilizando *JPA*.
- 4 Usar *Spring* para la definición de repositorios de acceso a datos.
- 5 Especificar una interfaz *REST* para la manipulación de un modelo de dominio.
- 6 Serializar objetos en formato *JSON* usando anotaciones.
- 7 Implementar un controlador *REST* usando *Spring*.
- 8 Implementar una capa de servicio usando *Spring*.
- 9 Implementar una interfaz *SPA* utilizando *AngularJS*.
- 10 Ser capaz de utilizar *tokens de autenticación*.
- 11 Utilizar los servicios de *Spring* para el control de accesos.

# Resultados de Aprendizaje

- 1 Elaborar modelos de dominio utilizando *POJOs*.
- 2 Clasificar elementos de dominio conforme a la metodología *Domain-Driven Design*.
- 3 Anotar un modelo de dominio para persistirlo utilizando *JPA*.
- 4 Usar *Spring* para la definición de repositorios de acceso a datos.
- 5 Especificar una interfaz *REST* para la manipulación de un modelo de dominio.
- 6 Serializar objetos en formato *JSON* usando anotaciones.
- 7 Implementar una controlador *REST* usando *Spring*.
- 8 Implementar una capa de servicio usando *Spring*.
- 9 Implementar una interfaz *SPA* utilizando *AngularJS*.
- 10 Ser capaz de utilizar *tokens de autenticación*.
- 11 Utilizar los servicios de *Spring* para el control de accesos.



# Resultados de Aprendizaje

- 1 Elaborar modelos de dominio utilizando *POJOs*.
- 2 Clasificar elementos de dominio conforme a la metodología *Domain-Driven Design*.
- 3 Anotar un modelo de dominio para persistirlo utilizando *JPA*.
- 4 Usar *Spring* para la definición de repositorios de acceso a datos.
- 5 Especificar una interfaz *REST* para la manipulación de un modelo de dominio.
- 6 Serializar objetos en formato *JSON* usando anotaciones.
- 7 Implementar un controlador *REST* usando *Spring*.
- 8 Implementar una capa de servicio usando *Spring*.
- 9 Implementar una interfaz *SPA* utilizando *AngularJS*.
- 10 Ser capaz de utilizar *tokens de autenticación*.
- 11 Utilizar los servicios de *Spring* para el control de accesos.

# Resultados de Aprendizaje

- 1 Elaborar modelos de dominio utilizando *POJOs*.
- 2 Clasificar elementos de dominio conforme a la metodología *Domain-Driven Design*.
- 3 Anotar un modelo de dominio para persistirlo utilizando *JPA*.
- 4 Usar *Spring* para la definición de repositorios de acceso a datos.
- 5 Especificar una interfaz *REST* para la manipulación de un modelo de dominio.
- 6 Serializar objetos en formato *JSON* usando anotaciones.
- 7 Implementar un controlador *REST* usando *Spring*.
- 8 Implementar una capa de servicio usando *Spring*.
- 9 Implementar una interfaz *SPA* utilizando *AngularJS*.
- 10 Ser capaz de utilizar *tokens de autenticación*.
- 11 Utilizar los servicios de *Spring* para el control de accesos.

# Resultados de Aprendizaje

- 1 Elaborar modelos de dominio utilizando *POJOs*.
- 2 Clasificar elementos de dominio conforme a la metodología *Domain-Driven Design*.
- 3 Anotar un modelo de dominio para persistirlo utilizando *JPA*.
- 4 Usar *Spring* para la definición de repositorios de acceso a datos.
- 5 Especificar una interfaz *REST* para la manipulación de un modelo de dominio.
- 6 Serializar objetos en formato *JSON* usando anotaciones.
- 7 Implementar un controlador *REST* usando *Spring*.
- 8 Implementar una capa de servicio usando *Spring*.
- 9 Implementar una interfaz *SPA* utilizando *AngularJS*.
- 10 Ser capaz de utilizar *tokens de autenticación*.
- 11 Utilizar los servicios de *Spring* para el control de accesos.

# Resultados de Aprendizaje

- 1 Elaborar modelos de dominio utilizando *POJOs*.
- 2 Clasificar elementos de dominio conforme a la metodología *Domain-Driven Design*.
- 3 Anotar un modelo de dominio para persistirlo utilizando *JPA*.
- 4 Usar *Spring* para la definición de repositorios de acceso a datos.
- 5 Especificar una interfaz *REST* para la manipulación de un modelo de dominio.
- 6 Serializar objetos en formato *JSON* usando anotaciones.
- 7 Implementar un controlador *REST* usando *Spring*.
- 8 Implementar una capa de servicio usando *Spring*.
- 9 Implementar una interfaz *SPA* utilizando *AngularJS*.
- 10 Ser capaz de utilizar *tokens de autenticación*.
- 11 Utilizar los servicios de *Spring* para el control de accesos.

# Resultados de Aprendizaje

- 1 Elaborar modelos de dominio utilizando *POJOs*.
- 2 Clasificar elementos de dominio conforme a la metodología *Domain-Driven Design*.
- 3 Anotar un modelo de dominio para persistirlo utilizando *JPA*.
- 4 Usar *Spring* para la definición de repositorios de acceso a datos.
- 5 Especificar una interfaz *REST* para la manipulación de un modelo de dominio.
- 6 Serializar objetos en formato *JSON* usando anotaciones.
- 7 Implementar un controlador *REST* usando *Spring*.
- 8 Implementar una capa de servicio usando *Spring*.
- 9 Implementar una interfaz *SPA* utilizando *AngularJS*.
- 10 Ser capaz de utilizar *tokens de autenticación*.
- 11 Utilizar los servicios de *Spring* para el control de accesos.

# Resultados de Aprendizaje

- 1 Elaborar modelos de dominio utilizando *POJOs*.
- 2 Clasificar elementos de dominio conforme a la metodología *Domain-Driven Design*.
- 3 Anotar un modelo de dominio para persistirlo utilizando *JPA*.
- 4 Usar *Spring* para la definición de repositorios de acceso a datos.
- 5 Especificar una interfaz *REST* para la manipulación de un modelo de dominio.
- 6 Serializar objetos en formato *JSON* usando anotaciones.
- 7 Implementar un controlador *REST* usando *Spring*.
- 8 Implementar una capa de servicio usando *Spring*.
- 9 Implementar una interfaz *SPA* utilizando *AngularJS*.
- 10 Ser capaz de utilizar *tokens de autenticación*.
- 11 Utilizar los servicios de *Spring* para el control de accesos.

# Resultados de Aprendizaje

- 1 Elaborar modelos de dominio utilizando *POJOs*.
- 2 Clasificar elementos de dominio conforme a la metodología *Domain-Driven Design*.
- 3 Anotar un modelo de dominio para persistirlo utilizando *JPA*.
- 4 Usar *Spring* para la definición de repositorios de acceso a datos.
- 5 Especificar una interfaz *REST* para la manipulación de un modelo de dominio.
- 6 Serializar objetos en formato *JSON* usando anotaciones.
- 7 Implementar un controlador *REST* usando *Spring*.
- 8 Implementar una capa de servicio usando *Spring*.
- 9 Implementar una interfaz *SPA* utilizando *AngularJS*.
- 10 Ser capaz de utilizar *tokens de autenticación*.
- 11 Utilizar los servicios de *Spring* para el control de accesos.

# Resultados de Aprendizaje

- 1 Elaborar modelos de dominio utilizando *POJOs*.
- 2 Clasificar elementos de dominio conforme a la metodología *Domain-Driven Design*.
- 3 Anotar un modelo de dominio para persistirlo utilizando *JPA*.
- 4 Usar *Spring* para la definición de repositorios de acceso a datos.
- 5 Especificar una interfaz *REST* para la manipulación de un modelo de dominio.
- 6 Serializar objetos en formato *JSON* usando anotaciones.
- 7 Implementar un controlador *REST* usando *Spring*.
- 8 Implementar una capa de servicio usando *Spring*.
- 9 Implementar una interfaz *SPA* utilizando *AngularJS*.
- 10 Ser capaz de utilizar *tokens de autenticación*.
- 11 Utilizar los servicios de *Spring* para el control de accesos.



# Resultados de Aprendizaje

- 1 Elaborar modelos de dominio utilizando *POJOs*.
- 2 Clasificar elementos de dominio conforme a la metodología *Domain-Driven Design*.
- 3 Anotar un modelo de dominio para persistirlo utilizando *JPA*.
- 4 Usar *Spring* para la definición de repositorios de acceso a datos.
- 5 Especificar una interfaz *REST* para la manipulación de un modelo de dominio.
- 6 Serializar objetos en formato *JSON* usando anotaciones.
- 7 Implementar un controlador *REST* usando *Spring*.
- 8 Implementar una capa de servicio usando *Spring*.
- 9 Implementar una interfaz *SPA* utilizando *AngularJS*.
- 10 Ser capaz de utilizar *tokens de autenticación*.
- 11 Utilizar los servicios de *Spring* para el *control de accesos*.

# Índice

- 1 Índice
- 2 Objetivos y Temario
  - Objetivos
  - **Temario**
- 3 Metodología
- 4 Métodos de Evaluación y Calificación
- 5 Bibliografía

# Temario

- ❶ Fundamentos.
- ❷ Capa de Negocio: Entidades de Dominio.
- ❸ Capa de Persistencia.
- ❹ Capa de Negocio: Servicios.
- ❺ Capa de Presentación.
- ❻ Seguridad: Control de Accesos.

# Temario

- 1 Fundamentos.
- 2 Capa de Negocio: Entidades de Dominio.
- 3 Capa de Persistencia.
- 4 Capa de Negocio: Servicios.
- 5 Capa de Presentación.
- 6 Seguridad: Control de Accesos.

# Temario

- 1 Fundamentos.
- 2 Capa de Negocio: Entidades de Dominio.
- 3 Capa de Persistencia.
- 4 Capa de Negocio: Servicios.
- 5 Capa de Presentación.
- 6 Seguridad: Control de Accesos.

# Temario

- 1 Fundamentos.
- 2 Capa de Negocio: Entidades de Dominio.
- 3 Capa de Persistencia.
- 4 Capa de Negocio: Servicios.
- 5 Capa de Presentación.
- 6 Seguridad: Control de Accesos.

# Temario

- ❶ Fundamentos.
- ❷ Capa de Negocio: Entidades de Dominio.
- ❸ Capa de Persistencia.
- ❹ Capa de Negocio: Servicios.
- ❺ Capa de Presentación.
- ❻ Seguridad: Control de Accesos.

# Temario

- ❶ Fundamentos.
- ❷ Capa de Negocio: Entidades de Dominio.
- ❸ Capa de Persistencia.
- ❹ Capa de Negocio: Servicios.
- ❺ Capa de Presentación.
- ❻ Seguridad: Control de Accesos.



# Índice

- 1 Índice
- 2 Objetivos y Temario
- 3 Metodología
- 4 Métodos de Evaluación y Calificación
- 5 Bibliografía

# Índice

- 1 Índice
- 2 Objetivos y Temario
- 3 Metodología
  - **Plataforma**
  - Actividades
- 4 Métodos de Evaluación y Calificación
- 5 Bibliografía

# Plataforma de Trabajo

- Las plataformas de trabajo de la asignatura son *moodle* y *Discord*.
- Todas las notificaciones oficiales se harán a través de *Discord*.
- Es obligación del alumno estar atento a las posibles notificaciones y avisos que se realicen a través de *Discord*.

# Plataforma de Trabajo

- Las plataformas de trabajo de la asignatura son *moodle* y *Discord*.
- Todas las notificaciones oficiales se harán a través de *Discord*.
- Es obligación del alumno estar atento a las posibles notificaciones y avisos que se realicen a través de *Discord*.

# Plataforma de Trabajo

- Las plataformas de trabajo de la asignatura son *moodle* y *Discord*.
- Todas las notificaciones oficiales se harán a través de *Discord*.
- Es obligación del alumno estar atento a las posibles notificaciones y avisos que se realicen a través de *Discord*.

# Índice

- 1 Índice
- 2 Objetivos y Temario
- 3 Metodología
  - Plataforma
  - **Actividades**
- 4 Métodos de Evaluación y Calificación
- 5 Bibliografía

# Clases en Aula

## Objetivo

Entender los conocimientos teóricos que constituyen la base de las habilidades y destrezas a adquirir al final de la asignatura.

- Clases magistrales, fundamentalmente de pizarra.
- Sin conocimiento teórico es imposible alcanzar las habilidades prácticas.

# Clases en Aula

## Objetivo

Entender los conocimientos teóricos que constituyen la base de las habilidades y destrezas a adquirir al final de la asignatura.

- Clases magistrales, fundamentalmente de pizarra.
- Sin conocimiento teórico es imposible alcanzar las habilidades prácticas.



# Clases en Aula

## Objetivo

Entender los conocimientos teóricos que constituyen la base de las habilidades y destrezas a adquirir al final de la asignatura.

- Clases magistrales, fundamentalmente de pizarra.
- Sin conocimiento teórico es imposible alcanzar las habilidades prácticas.

# Clases en Laboratorio

## Objetivo

Aplicar los conceptos teóricos aprendidos en las clases de aula al desarrollo de un sistema software real de mediana escala, con el objetivo de desarrollar las competencias procedimentales y actitudinales propias de un Ingeniero.

- Desarrollo de una aplicación empresarial de tamaño y complejidad moderada.
- El desarrollo estará dividido en varias etapas, al final de cada una de las cuales se realizará su correspondiente entrega.

# Clases en Laboratorio

## Objetivo

Aplicar los conceptos teóricos aprendidos en las clases de aula al desarrollo de un sistema software real de mediana escala, con el objetivo de desarrollar las competencias procedimentales y actitudinales propias de un Ingeniero.

- Desarrollo de una aplicación empresarial de tamaño y complejidad moderada.
- El desarrollo estará dividido en varias etapas, al final de cada una de las cuales se realizará su correspondiente entrega.

# Clases en Laboratorio

## Objetivo

Aplicar los conceptos teóricos aprendidos en las clases de aula al desarrollo de un sistema software real de mediana escala, con el objetivo de desarrollar las competencias procedimentales y actitudinales propias de un Ingeniero.

- Desarrollo de una aplicación empresarial de tamaño y complejidad moderada.
- El desarrollo estará dividido en varias etapas, al final de cada una de las cuales se realizará su correspondiente entrega.

# Índice

- 1 Índice
- 2 Objetivos y Temario
- 3 Metodología
- 4 **Métodos de Evaluación y Calificación**
- 5 Bibliografía

# Índice

- 1 Índice
- 2 Objetivos y Temario
- 3 Metodología
- 4 Métodos de Evaluación y Calificación
  - **Fórmula Calificación**
  - Ponderación de las Etapas del Proyecto
  - Prueba Final
- 5 Bibliografía

# Cálculo de la Calificación Final

## Fórmula de Cálculo de la Calificación Final

$$\text{Calificación Final} = \text{Proyecto} \times 0.75 + \text{Prueba Final Escrita} \times 0.25$$

- Para poder aplicar la fórmula anterior se debe obtener un mínimo de 5.00 en el proyecto.

# Cálculo de la Calificación Final

## Fórmula de Cálculo de la Calificación Final

$$\text{Calificación Final} = \text{Proyecto} \times 0.75 + \text{Prueba Final Escrita} \times 0.25$$

- Para poder aplicar la fórmula anterior se debe obtener un mínimo de 5.00 en el proyecto.



# Índice

- 1 Índice
- 2 Objetivos y Temario
- 3 Metodología
- 4 Métodos de Evaluación y Calificación
  - Fórmula Calificación
  - Ponderación de las Etapas del Proyecto
  - Prueba Final
- 5 Bibliografía

# Ponderación de las Etapas del Proyecto

E01	Modelo de Domino	10 %
E02	Domain-Driven Design	10 %
E03	Implementación POJOs	5 %
E04	Anotaciones JPA	10 %
E05	Desarrollo de los Repositorios	5 %
E06	Especificación de la API REST	10 %
E07	Implementación del Controlador HTTP	10 %
E08	Implementación de la Capa de Servicio	10 %
E09	Implementación del Router SPA	10 %
E10	Implementación de Compomentes	15 %
E11	Implementación de LLamadas AJAX	5 %
E12	Implementación del Control de Accesos	10 %

# Índice

- 1 Índice
- 2 Objetivos y Temario
- 3 Metodología
- 4 Métodos de Evaluación y Calificación
  - Fórmula Calificación
  - Ponderación de las Etapas del Proyecto
  - **Prueba Final**
- 5 Bibliografía

# Prueba Final

- 1 La prueba contendrá ejercicios y cuestiones sobre razonamientos teóricos.
- 2 Se podrá hacer uso de todo tipo de material escrito.
- 3 En ningún caso se dejará hacer uso de dispositivos electrónicos, especialmente, de aquellos con capacidades de comunicación inalámbrica.
- 4 El material escrito debe servir para consultar cuestiones puntuales, pero en el caso ideal no debería hacerse ningún uso de los mismos.
- 5 Aquellas respuestas que sean una simple copia de lo disponible en algún tipo de material de la asignatura se calificarán con 0 puntos.

# Prueba Final

- 1 La prueba contendrá ejercicios y cuestiones sobre razonamientos teóricos.
- 2 Se podrá hacer uso de todo tipo de material escrito.
- 3 En ningún caso se dejará hacer uso de dispositivos electrónicos, especialmente, de aquellos con capacidades de comunicación inalámbrica.
- 4 El material escrito debe servir para consultar cuestiones puntuales, pero en el caso ideal no debería hacerse ningún uso de los mismos.
- 5 Aquellas respuestas que sean una simple copia de lo disponible en algún tipo de material de la asignatura se calificarán con 0 puntos.

# Prueba Final

- 1 La prueba contendrá ejercicios y cuestiones sobre razonamientos teóricos.
- 2 Se podrá hacer uso de todo tipo de material escrito.
- 3 En ningún caso se dejará hacer uso de dispositivos electrónicos, especialmente, de aquellos con capacidades de comunicación inalámbrica.
- 4 El material escrito debe servir para consultar cuestiones puntuales, pero en el caso ideal no debería hacerse ningún uso de los mismos.
- 5 Aquellas respuestas que sean una simple copia de lo disponible en algún tipo de material de la asignatura se calificarán con 0 puntos.

# Prueba Final

- 1 La prueba contendrá ejercicios y cuestiones sobre razonamientos teóricos.
- 2 Se podrá hacer uso de todo tipo de material escrito.
- 3 En ningún caso se dejará hacer uso de dispositivos electrónicos, especialmente, de aquellos con capacidades de comunicación inalámbrica.
- 4 El material escrito debe servir para consultar cuestiones puntuales, pero en el caso ideal no debería hacerse ningún uso de los mismos.
- 5 Aquellas respuestas que sean una simple copia de lo disponible en algún tipo de material de la asignatura se calificarán con 0 puntos.

# Prueba Final

- 1 La prueba contendrá ejercicios y cuestiones sobre razonamientos teóricos.
- 2 Se podrá hacer uso de todo tipo de material escrito.
- 3 En ningún caso se dejará hacer uso de dispositivos electrónicos, especialmente, de aquellos con capacidades de comunicación inalámbrica.
- 4 El material escrito debe servir para consultar cuestiones puntuales, pero en el caso ideal no debería hacerse ningún uso de los mismos.
- 5 Aquellas respuestas que sean una simple copia de lo disponible en algún tipo de material de la asignatura se calificarán con 0 puntos.




# Índice


- 1 Índice
- 2 Objetivos y Temario
- 3 Metodología
- 4 Métodos de Evaluación y Calificación
- 5 **Bibliografía**

# Bibliografía Principal

 Martin Fowler.  
*Patterns of Enterprise Application Architecture.*  
Addison-Wesley Professional, 2002.

 Eric J. Evans.  
*Domain-Driven Design.*  
Pearson, 2003.

 Leonard Richardson and Sam Ruby.  
*RESTful Web Services.*  
O'Reilly, 2007.

 Emmit Scott.  
*SPA Design and Architecture: Understanding Single Page Web Applications.*  
Manning Publications, 2015.