

### Nombre del Cuso: Laboratorio de Introducción a la Programación y Computación II

<b>Categoría:</b> Obligatorio	<b>Semestre:</b> 2do Semestre
<b>Docente:</b> Inga. Claudia Rojas	<b>Auxiliar:</b> Noe Martinez
<b>Edificio:</b> T3	<b>Salón:</b> 014
<b>Día que se imparte:</b> Martes	<b>Horario:</b> 10:50 – 12:30

#### 1. Descripción del laboratorio

El laboratorio de Introducción a la Programación y Computación II cubrirá conceptos de desarrollo de software web y móvil (back-end y front-end), como los distintos componentes se enlazan para construir una solución que cumpla con los requerimientos del cliente y las tendencias del mercado.

#### 2. Objetivos

##### General

Mediante la realización de proyectos, ejercicios y resolución de dudas sobre los temas aprendidos se ampliarán y reforzarán los conocimientos adquiridos de los temas expuestos en la clase, de igual manera se brindará a los estudiantes herramientas para realizar los proyectos de la mejor manera.

##### Específicos

- Implementar conceptos sobre bases de datos relacionales.
- Aplicar una metodología de desarrollo de software en la realización del proyecto.
- Aprender y aplicar los distintos diagramas del análisis del sistema al desarrollo del software.
- Conocer los distintos entornos web e implementarlos con una base de datos relacional.
- Realizar reportes en base a consultas sobre una base de datos.

#### 3. Metodología

1. Se impartirán clases magistrales, con la participación de los estudiantes para el manejo de aplicaciones orientadas a cubrir los temas del contenido.
2. Autoaprendizaje y lectura constante de información relacionada al laboratorio.
3. Exámenes cortos, tareas, prácticas y proyectos, para el control del progreso.

#### 4. Competencias terminales

Al finalizar el laboratorio el estudiante desarrolla las siguientes competencias:

- Dominio en la implementación de una metodología de desarrollo ágil.
- Capacidad para desarrollar software en entorno web y móvil.
- Capacidad para construir una arquitectura basada en servicios.

## 5. Observaciones

1. Se requiere de un 80% de asistencia al laboratorio.
2. Es obligatorio aprobar el laboratorio con una nota mínima de 61 puntos para tener derecho a examen final y aprobación del curso.
3. Solo se calificarán exámenes y proyectos de estudiantes asignados en el curso.
4. El estudiante puede llevar el laboratorio en otra sección, pero la calificación de cada fase de proyecto debe ser exclusiva del auxiliar de la sección donde se encuentra asignado.

## 6. Contenido temático del laboratorio

### 1. Buenas prácticas de codificación

- 1.1. Documentación de código
- 1.2. Manejo de errores
- 1.3. Nombramiento de variables
- 1.4. Uso de variables globales
- 1.5. Estructura del código

### 2. Herramientas y lenguajes de desarrollo

#### 2.1. HTML

- 2.1.1. Estructura básica de un documento HTML
- 2.1.2. Encabezado
- 2.1.3. Formato y atributos de texto
- 2.1.4. Imágenes
- 2.1.5. Tablas
- 2.1.6. Formularios
- 2.1.7. Botones

#### 2.2. CSS

- 2.2.1. ¿Qué es CSS?
- 2.2.2. Cómo incluir CSS en un documento HTML
- 2.2.3. Formato (márgenes, bordes, fondos, colores, estilos, etc.)

#### 2.3. JavaScript

- 2.3.1. Introducción al lenguaje JavaScript
- 2.3.2. Conceptos principales y uso
- 2.3.3. Aplicaciones JavaScript HTML

#### 2.4. Bootstrap

- 2.4.1. Introducción
- 2.4.2. Diseño
- 2.4.3. Componentes
- 2.4.4. Utilidades

#### 2.5. Introducción Angular

- 2.5.1. ¿Qué es Angular?
- 2.5.2. Estructura de una aplicación Angular
- 2.5.3. ¿Qué es un componente?
- 2.5.4. Configuración del entorno de desarrollo
- 2.5.5. Puesta en marcha e instalación de Angular
- 2.5.6. TypeScript
  - 2.5.6.1. ¿Qué es TypeScript?
  - 2.5.6.2. Propiedades y tipos de datos
  - 2.5.6.3. El constructor, métodos de un componente y NgOnInit
  - 2.5.6.4. Diferencia entre let y var
  - 2.5.6.5. Crear clases, modelos de datos y objetos
  - 2.5.6.6. Condicionales
  - 2.5.6.7. TypeScript (TS) y programación orientada a objetos (POO)

### 2.5.7 Componentes

- 2.5.7.1 Creación y estructura de un componente
- 2.5.7.2 Configuración de un componente
- 2.5.7.3 Bootstrapping de nuestro primer componente
- 2.5.7.4 Eventos en los ciclos de vida de un componente
- 2.5.7.5 Configuración de eventos

### 2.5.8 Módulos

- 2.5.8.1 Estructura de un módulo
- 2.5.8.2 Configuración de un módulo

### 2.5.9 Directivas

- 2.5.9.1 ¿Qué es una directiva?
- 2.5.9.2 Preparación del proyecto para directivas
- 2.5.9.3 Directivas estructurales
- 2.5.9.4 Directivas de atributo
- 2.5.9.5 Directivas personalizadas
- 2.5.9.6 Host listener
- 2.5.9.7 Host binding

### 2.5.10 Pipes

- 2.5.10.1 Introducción a las pipes y filtros
- 2.5.10.2 Filtros para transformar caracteres
- 2.5.10.3 Pipes personalizados

### 2.5.11 Formularios reactivos

- 2.5.11.1 Creación de formularios reactivos
- 2.5.11.2 Validación y control de errores
- 2.5.11.3 Envío de un formulario

## 3. JSON

- 3.1. Definición
- 3.2. Sintaxis

## 4. NodeJs

- 4.1. Introducción a Node.js
- 4.2. Módulos
- 4.3. Servidor web en Node.js
- 4.4. Procesamiento POST y GET
- 4.5. Uso de NPM

## 5. Ionic

- 5.1. Instalando Cordova & Ionic
- 5.2. Crear una aplicación
- 5.3. Corriendo aplicación en el Browser
- 5.4. Estructura de archivos
- 5.5. Estructura de una vista de Ionic
- 5.6. Navegación
  - 5.6.1. Creando un nuevo page
  - 5.6.2. Navegar a un page
  - 5.6.3. Navegación por Tabs
  - 5.6.4. Parámetros de navegación
  - 5.6.5. Navegación hacia atrás
  - 5.6.6. Navegación raíz
- 5.7. Estilos (Theming)
  - 5.7.1. Introducción a estilos
  - 5.7.2. Colores Personalizados
  - 5.7.3. Alineación y transformación de texto
  - 5.7.4. Padding y Márgenes

## 7. Evaluación de rendimiento académico

El laboratorio debe ser aprobado con nota mínima de 61 puntos y es obligatorio para optar a examen final del curso.

Procedimiento de evaluación	Ponderación	Envío enunciado	Entrega
Prácticas	Práctica 1	1 pt.	-
	Práctica 2	1 pt.	-
	Práctica 3	1 pt.	-
<b>Total prácticas</b>	<b>03 pts.</b>		
Proyecto	Fase I	05 pts.	-
	Fase II	10 pts.	-
	Fase III	12 pts.	-
<b>Total proyecto</b>	<b>27 pts.</b>		
Nota de promoción	30 pts.		

## 8. Cronograma de actividades

Tema	Cantidad de días para cubrir el tema
Buenas prácticas de codificación	1
HTML	
CSS	2
JavaScript	
Bootstrap	1
Introducción Angular	
TypeScript	3
Componentes	
Módulos	
Directivas	3
Pipes	
Formularios reactivos	
JSON	2
NodeJS	
Ionic	2