



**Unión Europea**  
**Fondo Social Europeo**  
**Iniciativa de Empleo Juvenil**  
El FSE invierte en tu futuro



GOBIERNO  
DE ESPAÑA

MINISTERIO  
DE INDUSTRIA, COMERCIO  
Y TURISMO



Escuela de  
organización  
industrial



ANGULAR

# 0- INTRODUCCIÓN

Profesor: Pablo Matías Garramone Ramírez

# OBJETIVO

- Vuestra preparación académica para la inserción en el mercado laboral.
- Primera piedra para vuestra formación y sirva de puente para estudios avanzados.
- Conocimiento exhaustivo del Front end, HTML5, CSS, JavaScript, TypeScript y ANGULAR.
- Programar aplicaciones en la parte cliente.

# Horario Angular

- Serán de Lunes a Viernes de 8:00 a 14:00
- Habrá 2 descansos:
  - 10:40 a 11:00
  - 12:50 a 13:00

# Material del curso

- Usaremos una plataforma Moodle como aula virtual en principio usaremos la ubicada en:
- <http://213.32.68.22/moodle>
- Toda la documentación estará subida en esta plataforma.
- Mi correo para dudas o cualquier cuestión es: **pablo@iessanvicente.com**

# Evaluación

- El día a día
- Proyecto al final de curso
- La idea no es una nota, sino vuestra preparación y formación máxima



# Contenido

**Mis 150 horas las distribuiré (esto es estimativo):**

## **Bloque I**

- Presentación
- Instalación del Entorno de trabajo (IDE)
- Repaso de programación básica del Cliente
  - HTML 5
  - CSS
  - JAVASCRIPT
    - Fundamentos del lenguaje
    - Colecciones y funciones arrows
    - DOM
    - Eventos
  - TYPESCRIPT

## **Bloque II**

- Introducción a GIT
- Fundamentos Angular
  - Introducción y Componentes
  - Data Binding, Pipes y Ciclo de Vida
  - Componentes anidados y Servicios
  - Acceso a Servicios Web
  - Router de Angular
  - Formularios en Angular

# METODOLOGIA

- Curso eminentemente práctico.
- Realizaremos pequeños bloques teóricos con cantidades suficientes de práctica para asimilarlos.
- Adaptabilidad al alumnado.
- No dejamos a nadie atrás.
- Mínimo de evolución
- Todo se realiza en horas de clase

# PRESENTACIÓN

- Empecemos por el título, Desarrollo de Aplicaciones Web en Angular ¿qué significa?

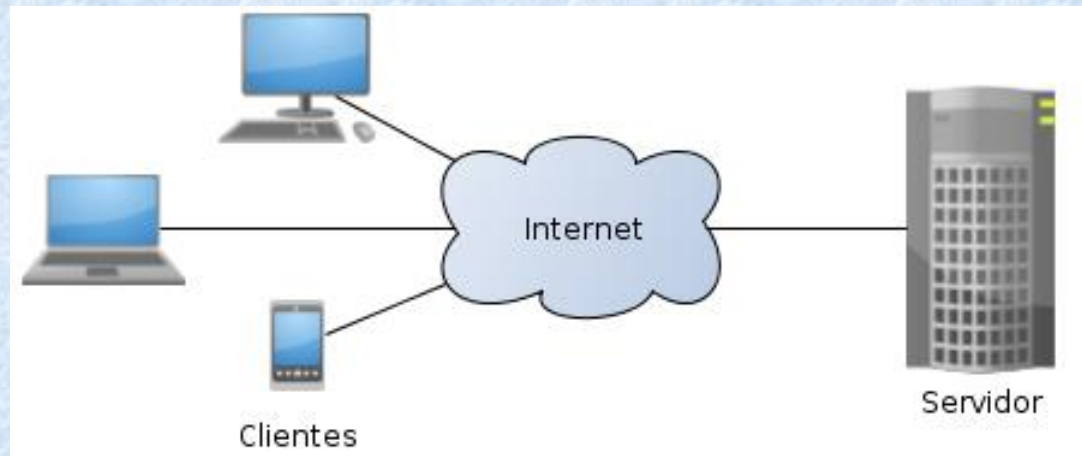


# Tipos de Aplicaciones

- **Escritorio**
- **Web**

# PROBLEMA APLICACIONES WEB

- Dependen de la conexión.
- Debe enviar todo por red, por lo que debe ocupar lo menos posible.
- Debe ir rápido.
- CLIENTE (Front End) – SERVIDOR (Back end)



# Tipos de Ficheros

- Planos
- Binarios
- Conceptos de bit (0,1) y byte. Código Ascii. ¿Por qué 1 byte son 8 bits?
- Ejemplo de tamaño fichero plano y binario.
- ¿Cuál usamos para enviar por red?



# Código Ascii

## Caracteres ASCII de control

00	NULL	(carácter nulo)
01	SOH	(inicio encabezado)
02	STX	(inicio texto)
03	ETX	(fin de texto)
04	EOT	(fin transmisión)
05	ENQ	(consulta)
06	ACK	(reconocimiento)
07	BEL	(timbre)
08	BS	(retroceso)
09	HT	(tab horizontal)
10	LF	(nueva línea)
11	VT	(tab vertical)
12	FF	(nueva página)
13	CR	(retorno de carro)
14	SO	(desplaza afuera)
15	SI	(desplaza adentro)
16	DLE	(esc.vínculo datos)
17	DC1	(control disp. 1)
18	DC2	(control disp. 2)
19	DC3	(control disp. 3)
20	DC4	(control disp. 4)
21	NAK	(conf. negativa)
22	SYN	(inactividad sínc)
23	ETB	(fin bloque trans)
24	CAN	(cancelar)
25	EM	(fin del medio)
26	SUB	(sustitución)
27	ESC	(escape)
28	FS	(sep. archivos)
29	GS	(sep. grupos)
30	RS	(sep. registros)
31	US	(sep. unidades)
127	DEL	(suprimir)

## Caracteres ASCII imprimibles

32	espacio	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	\$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

## ASCII extendido (Página de código 437)

128	Ç	160	á	192	Ł	224	Ó
129	ü	161	í	193	ł	225	ô
130	é	162	ó	194	Ł	226	Ô
131	â	163	ú	195	ł	227	Ò
132	ä	164	ñ	196	—	228	ö
133	à	165	Ñ	197	†	229	Õ
134	á	166	ª	198	ã	230	μ
135	ç	167	º	199	Ä	231	þ
136	ê	168	¿	200	Ł	232	ƒ
137	ë	169	®	201	ł	233	Ů
138	è	170	™	202	Ł	234	Ù
139	ï	171	½	203	ł	235	Ú
140	ì	172	¼	204	Ł	236	ý
141	î	173	ı	205	=	237	Ÿ
142	Ä	174	«	206	Ł	238	—
143	Å	175	»	207	□	239	'
144	É	176	⋮	208	ø	240	≡
145	æ	177	⋮	209	Ð	241	±
146	Æ	178	⋮	210	Ê	242	≡
147	ô	179	—	211	Ë	243	¾
148	õ	180	—	212	È	244	ŕ
149	ò	181	À	213	ı	245	§
150	û	182	Â	214	í	246	÷
151	ù	183	Ã	215	î	247	˙
152	ÿ	184	©	216	ï	248	°
153	Ö	185	Ł	217	Ĵ	249	ˆ
154	Ü	186	Ł	218	Ł	250	˙
155	ø	187	Ł	219	■	251	ˆ
156	£	188	Ł	220	■	252	ˆ
157	Ø	189	¢	221	ı	253	ˆ
158	×	190	¥	222	ı	254	■
159	f	191	Œ	223	■	255	nbsp

# Solución y Evolución

- Lenguajes de Marca o etiquetas.
- El propio texto se usa para construir etiquetas, y compartir estructura, estilo y contenido.



# LATEX

Ejemplo de código LaTeX:

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\title{\Ejemplo}
\begin{document}
Este es el texto ejemplo de \LaTeX{}
Con datos en \emph{cursiva} o \textbf{negrita}.
Ejemplo de f\''ormula
\begin{align}
E &= mc^2
\end{align}
\end{document}
```

Que con un traductor daría lugar al resultado:

Este es el texto ejemplo de L<sup>A</sup>T<sub>E</sub>X

Con datos en *cursiva* o **negrita**. Ejemplo de fórmula

$$E = mc^2$$

(1)

# RTF (Word pad)

```
{\rtf1\ansi\ansicpg1252\deff0\deflang3082{\fonttbl{\f0\fnil\fchar  
set0 Calibri;}}  
\viewkind4\uc1\pard\sa200\sl276\slmult1\lang10\f0\fs22 soy \i  
cursiva\i0\par }
```

Produce el resultado:

*soy cursiva*

# HTML

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Ejemplo1</title>
  </head>
  <body>
    <p>Párrafo de ejemplo</p>
  </body>
</html>
```

# Evolución

- Aparición del CSS. División de estructura y estilos.
- Aparición de JavaScript. Dinamismo a las páginas.
- Aparición de TypeScript. Hacer tipado a JavaScript
- Aparición de Angular. Framework que facilita el trabajo del desarrollo el front end

# ¿Qué es Angular?

- Angular es un **framework** para aplicaciones web desarrollado en **TypeScript**, de código abierto, mantenido por Google.
- Se utiliza para crear y mantener aplicaciones web de una sola página.
- Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de **Modelo Vista Controlador** (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.



# Concepto Framework

- Estructura conceptual y tecnológica de soporte definido.
- Normalmente con módulos de software concretos, que puede servir de base para la organización y desarrollo de software.
- Capas o cajas negras de las partes más tediosas o repetitivas que ya esta hechas y nosotros solo las usamos.
- Ejemplos conceptuales:
  - Calculadora.
  - Puerta de un coche

# Ventajas Framework

- Evitar escribir código repetitivo.
- Utilizar buenas prácticas.
- Permitir hacer cosas avanzadas que tú no harías.
- Desarrollar más rápido.

# DUDAS

- ¿Alguna pregunta?
- Pues a trabajar...