

# REPASO

**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev

# Repaso

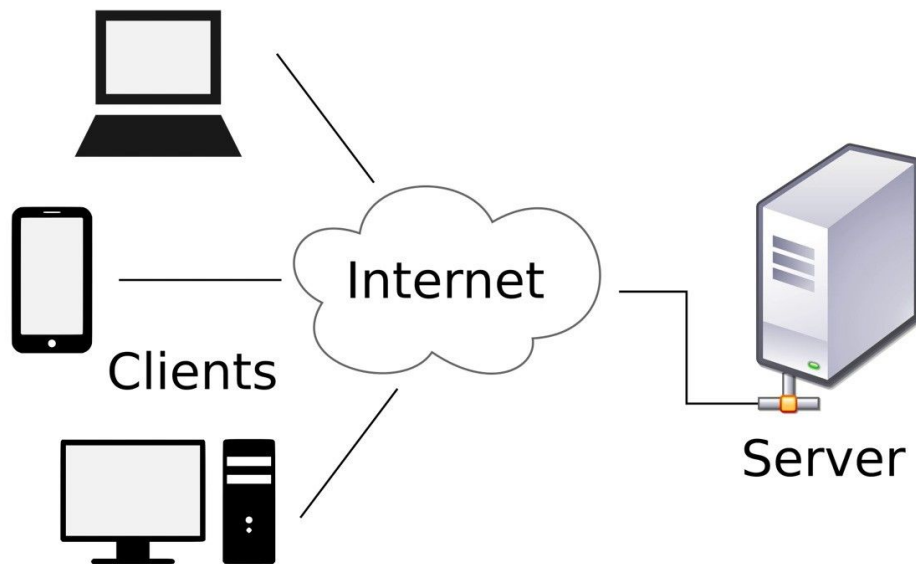
- Funciones, parámetros.
- Comandos en terminal.
- Git y github.
- Prototype y POO.
- ES6 (acceso a objetos y arreglos, arrow function, destructuring).

# Conceptos de servidores web

**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev

# Intro al modelo cliente servidor



# Intro al concepto de URL

Uniform resource identifier, sirve para acceder a un recurso físico o abstracto por Internet. A continuación se muestran los elementos mínimos de una URL: **protocolo://dominio:port/path?**

<https://www.tutorialesprogramacionya.com/javascriptya/nodejsya/detalleconcepto.php>

# Localhost

Es la dirección de mi propio computador, también se le conoce como dirección loopback.

- **Localhost:** <http://localhost:8080>
- **Loopback:** <http://127.0.0.1:8080>

# Domain Name Server (DNS)

Un servidor de nombres es un servidor de hardware o software que implementa un servicio de red para proveer respuestas a las consultas en un servicio de directorio. Traduce una ip a un nombre de dominio y viceversa.

## Ejemplos

- 127.0.0.1 => localhost
- localhost => 127.0.0.1
- 196.890.123.112 => www.google.com
- www.google.com => 196.890.123.112

# Mime Types

Los mime types indican la naturaleza y formato de los archivos que son transmitidos en la respuesta de una solicitud web.

- [Listado.](#)



# Semantic Versioning

**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev

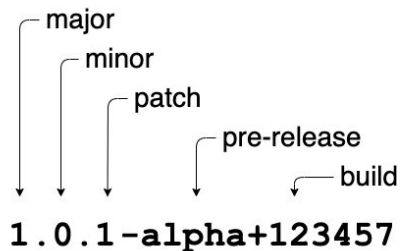
# Semantic Versión

Es un conjunto simple de reglas y requerimientos que dictan cómo asignar e incrementar los números de la versión de un software.

- Evitan la pérdida de versiones y mejoran la gestión de dependencias.



# Funcionamiento de semantic versión



Las versiones van incrementando de acuerdo a diferentes reglas:

- La versión **PARCHE** cuando reparas errores compatibles con versiones anteriores.
- La versión **MENOR** cuando añades funcionalidad que compatible con versiones anteriores.
- La versión **MAYOR** cuando realizas un cambio incompatible en el proyecto.

[Official Doc](#)

# Símbolos

Symbol	Dependency	Versions	Changes
tilde (~)	~3.9.2	3.9.*	-bug fix
caret (^)	^3.9.2	3.*.*	-backwards compatible new functionality -old functionality deprecated, but operationa -large internal refactor -bug fix

- **3.9.2:** Se instala exactamente la versión 3.9.2
- **~3.9.2:** Se instala la versión más reciente posible que no tengas nuevas funcionalidades.
- **^3.9.2:** Se instala la versión más reciente posible que no tenga breaking changes, es decir 3.X.Y

# Entendiendo conceptos

- Servicio  $\approx$  endpoint  $\approx$  recurso  $\approx$  servicio web.
- API  $\approx$  API REST  $\approx$  API Restful.
- Request = Petición = Solicitud.
- Response = Respuesta.
- Path  $\approx$  Dirección  $\approx$  Url  $\approx$  URI.
- File = Fichero = Archivo.
- Terminal = Línea de comandos = CLI = Bash.
- Script  $\approx$  Programa  $\approx$  Conjunto de líneas de código o instrucciones.
- Entidad (BD)  $\approx$  Clase (Programación)  $\approx$  Recurso (API).

