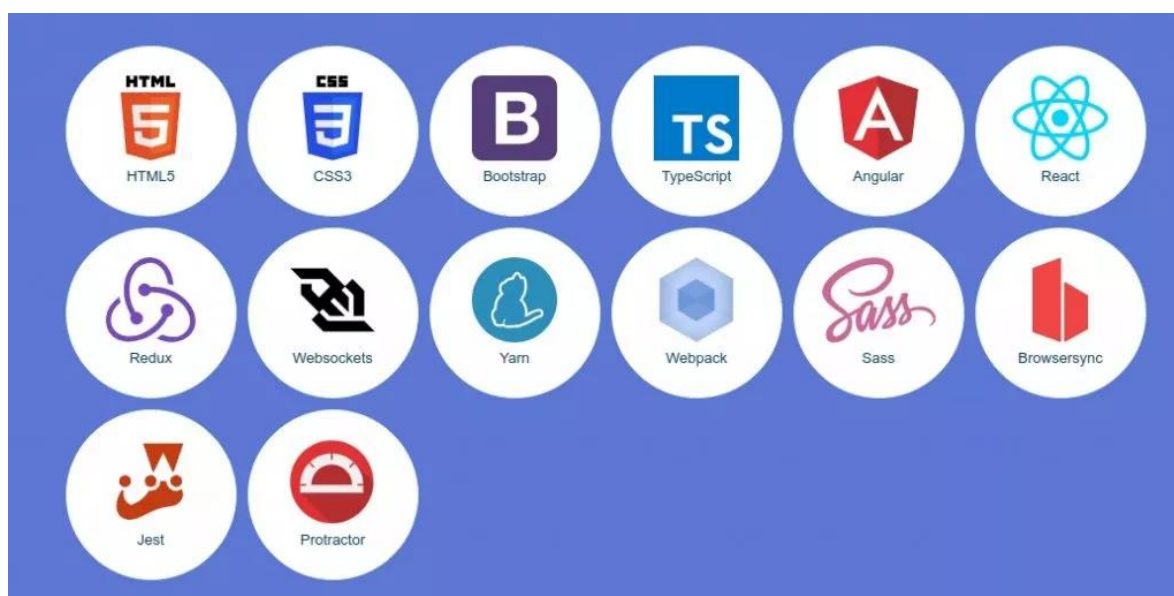


Tecnologías para el desarrollo web

El siguiente paso es hablar de los diferentes lenguajes de programación que existen dirigidos al desarrollo web. Sin embargo, antes de entrar en materia, conviene hacer una distinción entre la definición de tecnologías frontend y tecnologías backend.

Entendemos por tecnologías frontend al conjunto de tecnologías que funcionan y trabajan del lado del cliente. Por el contrario, las tecnologías backend son aquellas que funcionan y operan del lado de los servidores. Actualmente algunos de los lenguajes que se pueden encontrar en cuanto a desarrollo web son los siguientes:

Tecnologías frontend



HTML5: Es la tecnología principal utilizada para estructurar los contenidos en la web, estandarizada por la [w3c](https://www.w3.org/). Es un lenguaje de marcado que permite crear la estructura y elementos de una web que después serán interpretados por los navegadores.

CSS: Su función es describir cómo las estructuras HTML definidas anteriormente deben representarse visualmente. HTML fue creado para describir el contenido en las páginas web, CSS define los estilos visuales que tendrá ese contenido. Combinando **HTML** y **CSS** en dos archivos separados, es posible separar el contenido y la presentación de una web de forma clara y organizada, facilitando el trabajo de los desarrolladores.

JavaScript: Hoy en día es común que las webs incorporen dinamismo, permitan realizar acciones sin necesidad de refrescar la página y actualicen los datos en pantalla de forma automática y rápida. Para lograr que las webs tengan estos comportamientos se utiliza Javascript, un lenguaje empleado para aportar

interactividad, reactividad ante eventos, validación de los datos en formularios, crear cookies, etc.

Sobre estas 3 tecnologías se han creado frameworks de desarrollo web que permiten crear páginas web cada vez más complejas y funcionalidades que aportan cada vez más valor. Las últimas tendencias de desarrollo web en esta área son:

Frameworks y librerías JavaScript: con el fin de mejorar el desarrollo de JavaScript en proyectos grandes, han aparecido frameworks y librerías como [Angular](#), [Vuejs](#) y [React](#). El ecosistema **JavaScript** crece cada año y todos los días se crean nuevas librerías. Una buena referencia para mantenerse actualizados en estos avances [es esta guía](#). También surgen herramientas para poder testear webs basadas en estos frameworks, algunos ejemplos son [Jest](#), [Protactor](#), [Mocha](#), [Karma](#), [Enzyme](#).

Frameworks y librerías CSS: al igual que ocurre con **JavaScript**, también surge la aparición de nuevas librerías y frameworks para **CSS** que permiten gestionar mejor su desarrollo en grandes aplicaciones. Los más conocidos y utilizados son **Bootstrap y Material Design**, ver más en [diseño web](#). También aparecen los denominados **preprocesadores**, programas que permiten extender las funciones CSS con una sintaxis única que después será transformada al CSS original. Algunos ejemplos son: [Sass](#), [Less](#), [Stylus](#), y [PostCSS](#).

Gestores de paquetes: con el fin de gestionar librerías y plugins los desarrolladores utilizan gestores de paquetes, siendo los más extendidos [npm](#) y [yarn](#).

Herramientas para **construcción y desarrollo del código:** las más destacadas aquí son [NPM Scripts](#), [Gulp](#), [Webpack](#) y [Rollup](#).

Tecnologías backend



Los **lenguajes backend más populares** este último año han sido los siguientes en base a tipologías:

Lenguajes de scripting:

- Python
- Ruby
- JavaScript (Nodejs)
- PHP

Lenguajes funcionales:

- Elixir
- Scala

- Java
- Clojure

Otras opciones:

- Golang
- Rust

En función del lenguaje existen diferentes **frameworks que permiten el desarrollo a gran escala**, los más populares en la actualidad son:

- [Phoenix](#) (Elixir)
- [Meteor](#) (JavaScript Nodejs)
- [Ruby on Rails](#) (Ruby)
- [Django](#) (Python)
- [Flask](#) (Python)
- [Spring Boot y Spring Cloud](#) (Java)

Las **bases de datos** más utilizadas en base a su tipología son:

- Relacionales: MySQL, MariaDB, PostgreSQL, Oracle.
- NoSQL: MongoDB, Cassandra, CouchBase, RethinkDB, Elastic Search, Neo4j

En arquitecturas distribuidas es común el intercambio de mensajes entre los distintos módulos software, para ello se emplea **middleware de mensajería**:

- RabbitMQ
- Kafka

Motores de búsqueda o **Search Engines**:

- ElasticSearch
- Solr
- Sphinx

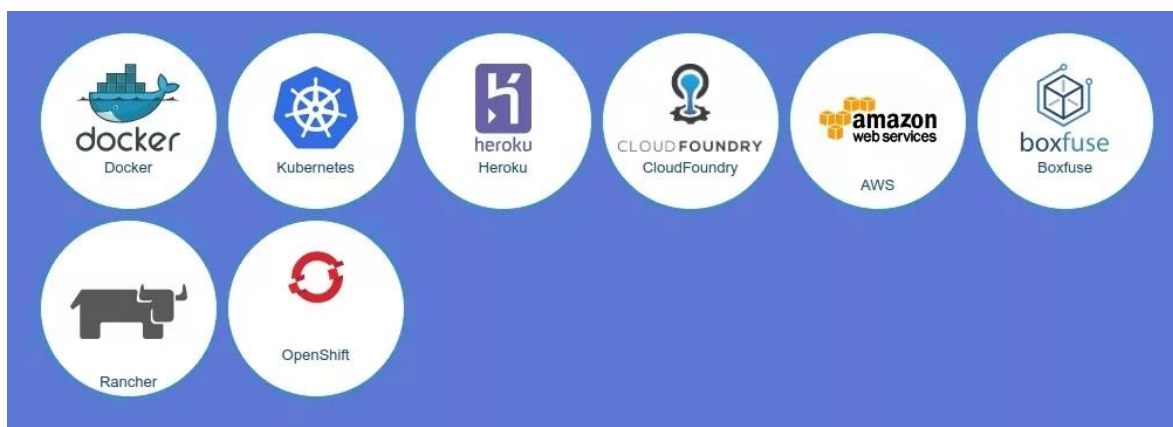
Sistemas de cacheo:

- Redis
- Memcached

Otro factor muy importante es la **seguridad**, crucial para cualquier tipo de software, las metodologías más comunes de **autenticación y autorización** son:

- OAuth
- Autenticación básica
- Autenticación por token
- JWT
- OpenID

Tecnologías DevOps



Con el creciente aumento de servicios en la nube, ha crecido la demanda de expertos en el área DevOps, personal especializado en tareas como:

- **Lenguajes de programación:** Go, Rust, C, C++, Python, Ruby, JavaScript (Nodejs)
- **Sistemas Operativos:** gestión de procesos, hilos y concurrencia, sockets, gestión I/O, virtualización.
- **Gestión de Servidores:** Linux, Windows.
- **Proveedores de servicios cloud:** Amazon Web Services, Google Cloud Platform, Azure, Digital Ocean.
- **Redes y Seguridad:** DNS, OSI Model, SSH, HTTP, HTTPS, FTP, SFTP, SSL / TLS.
- **Aprovisionamiento de infraestructuras IT:** Contenedores (Docker), Gestión de la Configuración (Ansible), Orquestación (Kubernetes), Aprovisionamiento (Terraform)
- **Integración Continua y Entrega Continua:** Jenkins, Circle CI, Gitlab.
- **Monitorización de infraestructura y aplicaciones:** Nagios, Datadog, NewRelic, Zabbix.
- **Gestión de logs, eventos y notificaciones:** ELK, Graylog, Splunk, Papertrail.