**Tecnologías frontend o del lado del cliente**

En esta tipología de tecnologías para Desarrollo Web se enmarcan aquellas que son las que se ocupan de la creación de interfaces de usuario, y de establecer comunicaciones con el servidor.

Logotipo

Descripción generada automáticamente **JavaScript**

Con este lenguaje de programación multiplataforma es posible dar una mayor interactividad y dinamismo a los sitios web. Entre sus distintas funcionalidades, JavaScript permite crear tanto animaciones como objetos, localizar errores en formularios, cambiar elementos web de manera intuitiva, crear cookies, etcétera. Además, también permite desarrollar aplicaciones tan potentes como Facebook o Twitter. Por esto, JavaScript es una de las tecnologías para Desarrollo Web más usadas en la actualidad.



HTML es un lenguaje de marcado usado para la creación de sitios web. Consiste en un conjunto de códigos cortos, que se clasifican como archivos de textos en las etiquetas. Dicho de otra manera, el texto se guarda en un archivo llamado HTML que se puede encontrar a través de los buscadores. Cada una de las etiquetas generadas tiene diferentes funciones.

De forma breve, HTML sirve para describir el contenido de un sitio web, como la información estructurada de párrafos, imágenes, etc. Por eso, HTML es una de las habilidades indispensables de un desarrollador web.

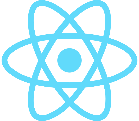


CSS es una de las tecnologías para Desarrollo Web a la que más recurren los programadores gráficos a la hora de desarrollar sus proyectos. Es un lenguaje que sirve, sobre todo, para indicar la representación visual de las estructuras HTML. De esta manera, el lenguaje CSS sirve para acotar y trazar el aspecto visual de las etiquetas generadas por el HTML.

**Frameworks y librerías frontend**

Además de los lenguajes de programación que definen el «idioma» y cómo estará escrito el código**,** es importante **destacar la importancia de los frameworks de desarrollo**. Una serie de herramientas y librerías de código pre-escrito que facilitan a los desarrolladores las tareas cotidianas o de uso común para evitar ejecutarlas desde cero.

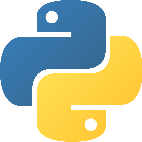
**Angular**Entre los más sonados y populares encontramos el framework Angular, perfecto para aplicaciones web. Además, utiliza HTML para realizar UI y es compatible con las últimas versiones de Chrome, Firefox, Edge, Android y iOS.

 **React**React está escrito en código abierto en JavaScript, muy útil para construir las interfaces de usuario. Es una biblioteca, aunque muchos programadores la consideran un framework por la excelencia de sus componentes. Así mismo, admite y combina diferentes idiomas y tecnologías.

**Tecnologías backend o del lado del servidor**

Son las que se ocupan de la implementación de comportamientos de la web en el servidor.

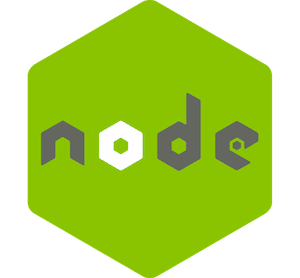
Es una de las mejores tecnologías para desarrollo web, ya que es de uso general y se adapta especialmente a este tipo de desarrollos. Se usa para conectar la plataforma web con el servidor de datos, permitiendo administrar los archivos del server, editar código, compiladores, la recopilación de datos o la modificación de la BBDD, entre más aspectos.

**Python**Sencillo pero potente, porque usa un menor número de líneas de código. Además, tiene varias librerías y es totalmente gratuito.

**Frameworks backend**

Los frameworks generan diversos beneficios en el desarrollo. El código web se encuentra organizado y estructurado desde el primer momento y se favorece el control y la seguridad de la plataforma filtrando la entrada y salida de datos.

Además, constan de una comunidad muy amplia y en continuo crecimiento. Por lo que siempre se pueden realizar consultas o estar al tanto de las nuevas actualizaciones.

 **NodeJS**

JavaScript se creó para permitir el desarrollo de la parte frontal, pero los últimos años ha ido evolucionando y ha creado su lugar en el backend con NodeJS. El lenguaje de JS para llevar a cabo toda la parte del servidor.

 **Laravel**

Tecnología de código abierto y uno de los frameworks más utilizados mundialmente. Ideal para desarrollar de forma elegante y simple aplicaciones y servicios web utilizando PHP.

**Bases de datos**

Estas son un conjunto de datos que pertenecen a un mismo contexto y se encuentran recogidos sistemáticamente para poder utilizarlos en cualquier momento. Para su gestión, existe una serie de sistemas encargados de almacenar, modificar y extraer información de ellas empleando programas específicos.

 **MySQL**

Es el gestor de bases de datos más utilizado en todo el mundo por los programadores actuales. Se utiliza para almacenar la información, y trabaja con múltiples tablas que permiten crear nuevas bases de datos y hacer cualquier consulta.

 **SQL Server**

Microsoft SQL Server es un sistema de gestión de base de datos relacional, desarrollado por la empresa Microsoft.

El lenguaje de desarrollo utilizado (por línea de comandos o mediante la interfaz gráfica de Management Studio) es Transact-SQL (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crear tablas y definir relaciones entre ellas (DDL).

 **PostgreSQL**

PostgreSQL, también llamado Postgres, es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto, publicado bajo la licencia PostgreSQL,1​ similar a la BSD o la MIT.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyados por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

PostgreSQL no tiene un gestor de errores (bugs), haciendo muy difícil conocer el estado de corrección de los mismos.2​

**Metodologías de desarrollo de software**

Las metodologías de desarrollo de software tradicionales se caracterizan por definir total y rígidamente los requisitos al inicio de los proyectos de ingeniería de software. Los ciclos de desarrollo son poco flexibles y no permiten realizar cambios, al contrario que las metodologías ágiles; lo que ha propiciado el incremento en el uso de las segundas.

La organización del trabajo de las metodologías tradicionales es lineal, es decir, las etapas se suceden una tras otra y no se puede empezar la siguiente sin terminar la anterior. Tampoco se puede volver hacia atrás una vez se ha cambiado de etapa. Estas metodologías, no se adaptan nada bien a los cambios, y el mundo actual cambia constantemente. Las principales metodologías tradicionales o clasicas son:

**Waterfall (cascada):** es una metodología en la que las etapas se organizan de arriba a abajo, de ahí el nombre. Se desarrollan las diferentes funciones en etapas diferenciadas y obedeciendo un riguroso orden. Antes de cada etapa se debe revisar el producto para ver si está listo para pasar a la siguiente fase. Los requisitos y especificaciones iniciales no están predispuestos para cambiarse, por lo que no se pueden ver los resultados hasta que el proyecto ya esté bastante avanzado.

**Prototipado**: se basa en la construcción de un prototipo de software que se construye rápidamente para que los usuarios puedan probarlo y aportar feedback. Así, se puede arreglar lo que está mal e incluir otros requerimientos que puedan surgir. Es un modelo iterativo que se basa en el método de prueba y error para comprender las especificidades del producto.

**Espiral**: es una combinación de los dos modelos anteriores, que añade el concepto de análisis de riesgo. Se divide en cuatro etapas: planificación, análisis de riesgo, desarrollo de prototipo y evaluación del cliente. El nombre de esta metodología da nombre a su funcionamiento, ya que se van procesando las etapas en forma de espiral. Cuanto más cerca del centro se está, más avanzado está el proyecto.

**Incremental**: en esta metodología de desarrollo de software se va construyendo el producto final de manera progresiva. En cada etapa incremental se agrega una nueva funcionalidad, lo que permite ver resultados de una forma más rápida en comparación con el modelo en cascada. El software se puede empezar a utilizar incluso antes de que se complete totalmente y, en general, es mucho más flexible que las demás metodologías.

**Diseño rápido de aplicaciones (RAD):** esta metodología permite desarrollar software de alta calidad en un corto periodo de tiempo. Los costes son mucho más altos y el desarrollo más flexible, aunque requiere una mayor intervención de los usuarios. Por otro lado, el código puede contener más errores, y sus funciones son limitadas debido al poco tiempo del que se dispone para desarrollarlas. El objetivo es iterar el menor número posible de veces para conseguir una aplicación completa de forma rápida.

**Metodologías de desarrollo de software ágiles**

Como hemos dicho al principio del artículo, las metodologías ágiles de desarrollo de software son las más utilizadas hoy en día debido a su alta flexibilidad y agilidad. Los equipos de trabajo que las utilizan son mucho más productivos y eficientes, ya que saben lo que tienen que hacer en cada momento. Además, la metodología permite adaptar el software a las necesidades que van surgiendo por el camino, lo que facilita construir aplicaciones más funcionales.

Las metodologías ágiles se basan en la metodología incremental, en la que en cada ciclo de desarrollo se van agregando nuevas funcionalidades a la aplicación final. Sin embargo, los ciclos son mucho más cortos y rápidos, por lo que se van agregando pequeñas funcionalidades en lugar de grandes cambios.

Este tipo de metodologías permite construir equipos de trabajo autosuficientes e independientes que se reúnen cada poco tiempo para poner en común las novedades. Poco a poco, se va construyendo y puliendo el producto final, a la vez que el cliente puede ir aportando nuevos requerimientos o correcciones, ya que puede comprobar cómo avanza el proyecto en tiempo real.

Las principales metodologías ágiles son:

**Kanban**: metodología de trabajo inventada por la empresa de automóviles Toyota. Consiste en dividir las tareas en porciones mínimas y organizarlas en un tablero de trabajo dividido en tareas pendientes, en curso y finalizadas. De esta forma, se crea un flujo de trabajo muy visual basado en tareas prioritarias e incrementando el valor del producto.

**Scrum**: es también una metodología incremental que divide los requisitos y tareas de forma similar a Kanban. Se itera sobre bloques de tiempos cortos y fijos (entre dos y cuatro semanas) para conseguir un resultado completo en cada iteración. Las etapas son: planificación de la iteración (planning sprint), ejecución (sprint), reunión diaria (daily meeting) y demostración de resultados (sprint review). Cada iteración por estas etapas se denomina también sprint.

**Lean**: está configurado para que pequeños equipos de desarrollo muy capacitados elaboren cualquier tarea en poco tiempo. Los activos más importantes son las personas y su compromiso, relegando así a un segundo plano el tiempo y los costes. El aprendizaje, las reacciones rápidas y potenciar el equipo son fundamentales.

**Programación extrema (XP)**: es una metodología de desarrollo de software basada en las relaciones interpersonales, que se consideran la clave del éxito. Su principal objetivo es crear un buen ambiente de trabajo en equipo y que haya un feedback constante del cliente. El trabajo se basa en 12 conceptos: diseño sencillo, testing, refactorización y codificación con estándares, propiedad colectiva del código, programación en parejas, integración continua, entregas semanales e integridad con el cliente, cliente in situ, entregas frecuentes y planificación

**TECNOLOGIA DE FRONTEND Y BACKEND SELECCIONADA**

**¿Por qué debemos usar Angular?**

**Coherencia y reutilización del código**

La estructura de Angular se basa en componentes, lo que hace que sean altamente reutilizables ahorrándonos tiempo a la hora de desarrollar. Podemos crear una interfaz con componentes móviles garantizando un proceso de desarrollo estable y permitiendo incorporar más funciones con códigos más precisos.

**Fácil de aprender**

Angular es muy fácil de aprender, sobre todo si tenemos conocimientos de JavaScript, HTML y CSS. Adquirir conocimientos de Angular nos abre más oportunidades en el sector del desarrollo y nos ayuda en los proyectos con un código más breve, facilitándonos el trabajo y ahorrándonos tiempo.

**Soporte de Google y gran soporte de la comunidad**

Como ya hemos comentado, Angular es un framework desarrollado y mantenido por Google, siendo además, uno de los más fiables. Además, tiene un gran apoyo debido a la comunidad de desarrolladores de código abierto.

**Funciones orientadas a SPA**

Angular es compatible con el desarrollo de aplicaciones de una sola página, consiguiendo una transición más rápida del sitio web. Antes al navegar por una web, se cargaban páginas nuevas con cada clic, ahora, cambia la página actual con datos nuevos del servidor, de esta manera el sitio web parece y funciona como una aplicación nativa.

**Enlace de datos bidireccional**

Una de las razones principales de la gran demanda de Angular es su funcionalidad de enlace de datos bidireccional. Usando el enlace de datos bidireccional, nuestra aplicación simplificará su capa de presentación.

**IU declarativa**

Angular usa código HTML para la interfaz de usuario ya que, en comparación con JavaScript, es un lenguaje más fácil. HTML es también un lenguaje declarativo e intuitivo y que seguro nos facilitará el trabajo.

**Perfecta integración y gran productividad**

Cuando implementamos Angular en proyectos de desarrollo web, se integra perfectamente con otras bibliotecas. La integración de Angular viene prediseñada con frameworks como Wijmo o Ionic, por ejemplo.

Como ya hemos comentado, Angular es un framework fiable y bien diseñado que nos ahorra mucho trabajo a los desarrolladores.

Tiene mejores IDE (entornos de desarrollo integrado) que nos permiten localizar los errores casi de forma instantánea y además tiene muchas plantillas que nos ayudan a desarrollar la IU más rápido.

La demanda de desarrolladores web es muy alta y va en aumento. Con una buena formación encontrar trabajo en este sector no será difícil en comparación con otras áreas y además, con unos salarios muy interesantes.

Por eso te recomendamos nuestro máster en Máster en Desarrollo Web Fullstack donde aprenderás con los mejores profesores y podrás optar a prácticas en empresas punteras del sector.

**MOTOR DE BASE DE DATOS A UTILIZAR**

**METODOLOGIA DE DESARROLLO A UTILIZAR**

**ARQUITECTRURA DE DESARROLLO (HARDWARE Y SOFTWARE)**

SO: Linux

Servidor Web: Apache o N