

¿Qué es un desarrollador Back-end?



Front End

- Markup and web languages such as HTML, CSS and Javascript
- Asynchronous requests and Ajax
- Specialized web editing software
- Image editing
- Accessibility
- Cross-browser issues
- Search engine optimisation



Back End

- Programming and scripting such as Python, Ruby and/or Perl
- Server architecture
- Database administration
- Scalability
- Security
- Data transformation
- Backup

Los usuarios no acceden directamente al backend de una aplicación o sistema informático.

La tecnología, como los servidores, las bases de datos y las API, permite que funcionen funciones como la interfaz de usuario.

Los desarrolladores de backend construyen y mantienen esa tecnología. Puede que los usuarios ni siquiera sean conscientes del trabajo del desarrollador de backend, pero sin él, el sitio web o la aplicación no funcionarían.

Desarrollador backend frente a desarrollador frontend

Los desarrolladores web frontend trabajan en las partes del sitio web que el usuario ve y con las que interactúa. Esto también se llama a veces desarrollo del lado del cliente o del lado del navegador.

Los desarrolladores backend (o del lado del servidor) proporcionan la infraestructura para respaldar eso.

Aunque los desarrolladores de frontend y backend suelen trabajar juntos, tienen un arsenal diferente de lenguajes y habilidades de programación.

Por ejemplo, los lenguajes con los que trabajan principalmente los desarrolladores web frontend son HTML, CSS y JavaScript. Mientras codifican, tienen que pensar en la experiencia del usuario. ¿Cómo se ven las imágenes? ¿Es el sitio accesible? ¿Funciona bien en los principales navegadores?

Los desarrolladores de backend utilizan lenguajes de programación y scripting como Python, Ruby y PHP. Se centran en cuestiones como si el sitio es seguro o cómo se gestionan los datos.

Hay otro tipo de desarrollador llamado desarrollador full-stack. Full-stack significa que haces tanto desarrollo backend como frontend.

Puede parecer que conocer todas las facetas del desarrollo web es mejor que limitarse a una, pero muchos trabajos buscan especialistas.

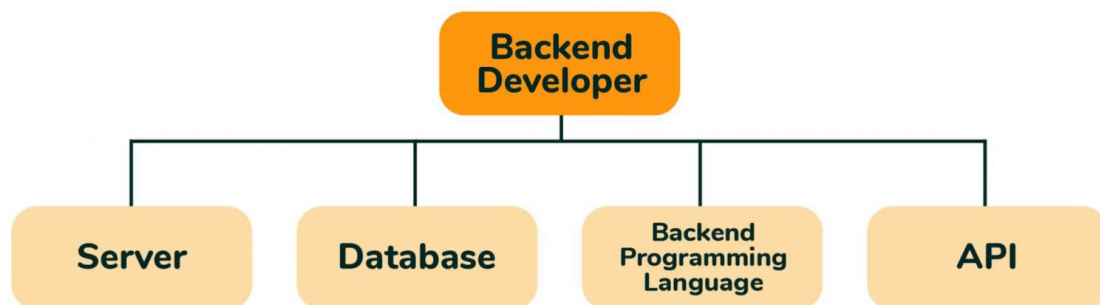
¿Qué hace un desarrollador backend?

Un desarrollador web backend trabaja en el servidor, la base de datos y las APIs que hay detrás de una aplicación web.

La descripción del trabajo es diferente a la de un desarrollador frontend, pero los dos trabajan juntos para crear un trabajo que sea funcional y fácil de usar.

Por ejemplo, un desarrollador de frontend crea una página en un sitio web con un botón en el que un cliente puede hacer clic para recuperar datos sobre sus pedidos anteriores.

El desarrollador backend escribiría el código que permite que el botón obtenga los datos correctos de la base de datos. Esos datos se devuelven al frontend, donde el desarrollador del frontend determina cómo se muestran al visitante.



 InterviewBit

Descripción del trabajo de desarrollador backend

Cada trabajo de desarrollador backend será un poco diferente. En general, los desarrolladores de backend se encargarán de tareas como:

- Desarrollar infraestructura de backend para sitios y aplicaciones web
- Mantener el backend existente
- Solucionar problemas y arreglar errores
- Gestionar entornos de alojamiento
- Pruebas de control de calidad
- Trabajar con desarrolladores de frontend para transferir datos de forma eficiente a las aplicaciones web del lado del cliente
- Colaborar con clientes y departamentos como la gestión de productos, el marketing, las ventas o el servicio de atención al cliente para desarrollar soluciones
- Actualizar la documentación sobre la arquitectura, los procesos y los sistemas

¿Por qué convertirse en desarrollador backend?

Aprender a ser un desarrollador backend te llevará tiempo y compromiso. Puede que te preguntes si merece la pena.

El desarrollo backend puede ser un campo profesional lucrativo y satisfactorio. He aquí algunas razones.

Demanda de desarrolladores backend

El desarrollo web es un campo de rápido crecimiento.

En todas las especialidades de desarrollo web, se espera un crecimiento del 8% de los puestos de trabajo en la próxima década. Eso supone unos 13.400 puestos de trabajo al año, un crecimiento mucho más rápido que la media.

Satisfacción de la carrera de desarrollador backend

El dinero y las oportunidades laborales son estupendos, pero tú quieres un trabajo que realmente disfrutes.

La buena noticia es que la mayoría de los desarrolladores de backend están contentos con lo que hacen.

Los desarrolladores backend califican su satisfacción profesional como mejor que la media.

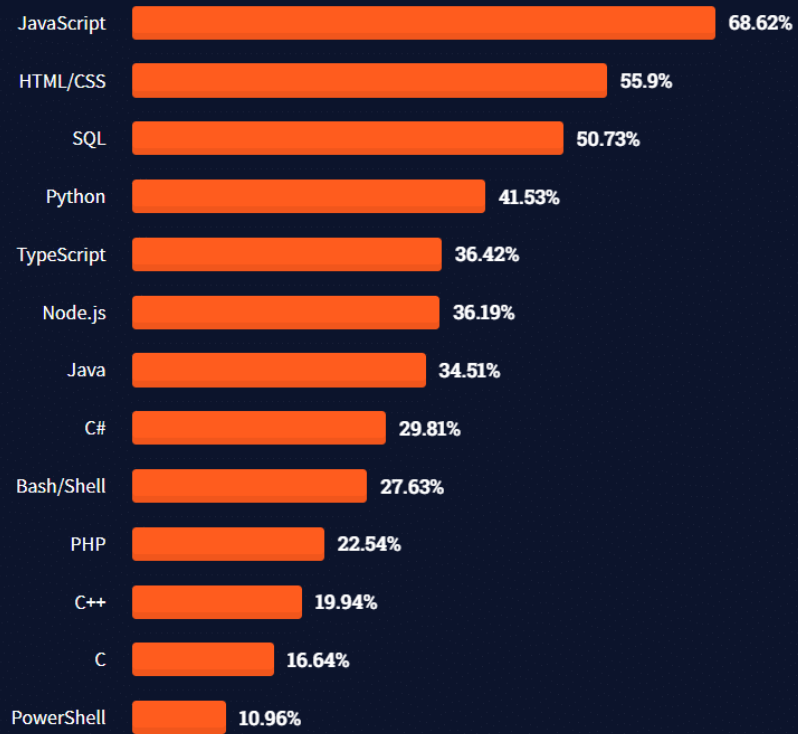
(En caso de que te lo preguntes, una encuesta mostró que están ligeramente más satisfechos que los desarrolladores frontend y igual de satisfechos como los desarrolladores de full-stack)

Habilidades y lenguajes de programación de los desarrolladores backend

Cada trabajo de desarrollador backend requiere un conjunto de habilidades ligeramente diferente.

Las habilidades y los lenguajes de programación que se indican a continuación son los más demandados por los desarrolladores de backend y te convertirán en un candidato competitivo.

También es una buena idea tener un conocimiento básico de las habilidades básicas del frontend, como HTML y CSS. Aunque no los utilices para el trabajo de backend, necesitarás tener los conocimientos suficientes para comunicarte con tus colegas desarrolladores frontend sobre ellos.



¿Qué es Spring?



Como definición podemos decir que Spring es un framework de código abierto para la creación de aplicaciones empresariales Java, con soporte para Groovy y Kotlin. Tiene una estructura modular y una gran flexibilidad para implementar diferentes tipos de arquitectura según las necesidades de la aplicación.

¿Qué es un framework?

Spring se considera un framework, pero este es un concepto que en ocasiones se confunde, así que vamos a ver qué es un framework.

Una librería es un conjunto de clases, de funciones y de utilidades que nos permiten realizar algunos procesos. Un buen ejemplo de librería sería una librería matemática, a la cual le podemos dar muchos datos y nos puede calcular, por ejemplo, la desviación típica, o le podríamos plantear una integral y la podría resolver.

A diferencia de una librería, un framework es:

- Un conjunto de artefactos software, es decir, que puede incluir una librería, de conceptos y de metodologías.
- Nos provee de un mecanismo genérico para resolver uno o más problemas de un tipo determinado.
- Es extensible a través de código escrito por los usuarios.
- Ofrece facilidad para el desarrollo y despliegue.

Si tuviéramos que desarrollar una aplicación web, podríamos utilizar un framework que nos facilite la tarea, que nos aporte soluciones a ese desarrollo. Uno de ellos podría ser, por ejemplo Spring MVC, que nos permitiría crear fácilmente una aplicación web, ya que nos aislaría de determinados problemas, como el hecho de crear servlet o registrar las peticiones, así nos podríamos dedicar a lo que realmente importa.

Qué es una aplicación empresarial

Una aplicación empresarial normalmente es:

- Una gran aplicación, es decir, suele ser bastante amplia y orientada a un ámbito comercial o industrial.
- Compleja, es decir, con bastantes funcionalidades, con muchos requisitos y que será utilizada por muchas personas, por lo que debería ser escalable, que

pueda crecer con el tiempo, es decir, que sí hoy tenemos mil usuarios y dentro de un año tenemos un millón, que no haya que rehacer la aplicación de nuevo.

- Distribuida, es decir, que incluso la podamos tener deslocalizada en distintos servidores.
- Crítica, es decir, que no permita o no tolere fallos e incluso, si tuviéramos algún tipo de fallo, que sea capaz de reponerse.
- Orientada a desplegarse dentro de redes corporativas o, el esquema más usual a día de hoy, en internet a través de la nube.
- Centrada en los datos, por lo que todas las funcionalidades se suelen desarrollar en torno a los mismos.
- Intuitiva, de un uso fácil para evitar el rechazo de los usuarios.
- Suele tener, además, unos grandes requisitos de seguridad y de mantenibilidad.

Todas estas serían las características que definirían a una aplicación empresarial.

Spring funciona sobre JVM

Spring funciona con Java, aunque también tiene soporte para otras tecnologías.

Inicialmente se diseñó para trabajar con Java SE y algunas especificaciones o APIs de Java EE, pero a día de hoy trabaja con el JDK 8 y JDK 9. También podemos trabajar con Groovy y tiene soporte para Kotlin.

Spring tiene estructura modular

Spring ya no es solo un framework para la inyección de dependencias, sino que tiene toda una familia de proyectos que abarcan muchos ámbitos: el ámbito de desarrollo de aplicaciones web, aplicaciones web reactivas, seguridad, servicios web, microservicios, Android, etcétera.

Además, dentro de alguno de esos proyectos, podemos encontrar que tiene una estructura modular, es decir, que está orientada a poder tener distintos módulos que agrupan diversas funcionalidades, desde el contenedor de inversión de control, la programación orientada a aspectos, el acceso a datos, etcétera.

Spring es flexible

Spring nos permite desarrollar todo tipo de aplicaciones diferentes:

Aplicaciones de escritorio, aplicaciones de línea de comando, aplicaciones web clásicas, web reactivas, microservicios...

Aplicaciones que acceden a base de datos vía SQL directamente, a través de algún tipo de ORM, bases de datos NoSQL...

Aplicaciones con esquemas de seguridad clásica donde almacenamos en nuestra base de datos los elementos de seguridad, clave o credenciales, pero también con otros sistemas.

Aplicaciones pequeñas, medianas y grandes, aplicaciones que tienden a ser escalables, aplicaciones que vayamos a gestionar a través de contenedores, aplicaciones que necesitemos desplegar en la nube.

Crear Nuestro Primer proyecto en Spring

Link video Crear proyecto con java Spring: <https://youtu.be/qdPEZjhzjIU>

¿Qué se puede hacer con Spring?



Cada una de estas características de Spring fueron desarrolladas en una serie de proyectos independientes, donde cada uno es utilizado para diferentes fines. Entre los principales proyectos se encuentran:

Spring Boot: Facilita la creación y configuración inicial de proyectos de Spring para generar aplicaciones de fácil y rápida puesta en marcha.

Spring Data: Utilizado para la administración, manejo y comunicación con bases de datos, tanto relacionales como no-relacionales.

Spring Security: Utilizado para las cuestiones de seguridad que puede necesitar todo proyecto.

Spring Web Services: Utilizado para facilitar el desarrollo de Web Services SOAP.

Spring Framework vs Spring Boot

Spring Boot es una extensión de Spring Framework y se encuentra dentro de su lista de proyectos. Fue creado con la finalidad de facilitar la creación de aplicaciones web listas para salir a producción, es decir, bajo el concepto "Just Run" (solo ejecutar).

Anteriormente, realizar las configuraciones iniciales para llevar a cabo una aplicación en Spring llevaba mucho tiempo a los desarrolladores. Esto, se realizaba mediante una configuración manual de un archivo xml y de un servidor de aplicaciones web, consumiendo gran parte del tiempo de desarrollo del proyecto en realizar configuraciones. Dada esta problemática y con la finalidad de resolverla, fue desarrollado Spring Boot, que requiere una configuración mínima y que puede ser integrado con otros proyectos de Spring o librerías externas. En la siguiente imagen se pueden observar las características incorporadas y quitadas de Spring Framework que lograron la implementación de Spring Boot.

