# **¿Qué es Django?**

Django (*gdh/ˈdʒæŋɡoʊ/jang-goh*) es un framework para aplicaciones web gratuito y open source, escrito en Python. Un framework web es un conjunto de componentes que te ayudan a desarrollar sitios web más fácil y rápidamente.

Cuando construyes un sitio web, siempre necesitas un conjunto de componentes similares: una manera de manejar la autenticación de usuarios (registrarse, iniciar sesión, cerrar sesión), un panel de administración para tu sitio web, formularios, una forma de subir archivos, etc.

Por suerte para ti, hace tiempo atrás varias personas notaron que los desarrolladores web enfrentan problemas similares cuando construyen un sitio nuevo, por eso se unieron y crearon frameworks (Django es uno de ellos) que te ofrecen componentes listos para usarse.

Los frameworks existen para ahorrarte tener que reinventar la rueda y ayudarte a aliviar la carga cuando construyes un sitio nuevo.

## **¿Por qué necesitas un framework?**

Para entender para qué es Django, necesitamos mirar más de cerca a los servidores. Lo primero que el servidor necesita saber es, que quieres que te sirva una página web.

Imagina un buzón (puerto) el cual es monitoreado por cartas entrantes (peticiones). Esto es realizado por un servidor web. El servidor web lee la carta, y envía una respuesta con una página web. Pero cuando quieres enviar algo, tienes que tener algún contenido. Y Django es quien que te ayuda a crear el contenido.

## **¿Qué sucede cuando alguien solicita una página web de tu servidor?**

Cuando llega una petición a un servidor web, es pasada a Django quien intenta averiguar lo que realmente es solicitado. Toma primero una dirección de página web y trata de averiguar qué hacer. Esta parte es realizada por **urlresolver** de Django (ten en cuenta que la dirección de un sitio web es llamada URL - Uniform Resource Locator; así que el nombre *urlresolver* tiene sentido). Este no es muy inteligente; toma una lista de patrones y trata de hacer coincidir la URL. Django comprueba los patrones de arriba hacia abajo y si algo coincide entonces Django le pasa la solicitud a la función asociada (que se llama *vista*).

Imagina a una cartera llevando una carta. Ella está caminando por la calle y comprueba cada número de casa con el que está en la carta. Si coincide, ella deja la carta allí. ¡Así es como funciona el urlresolver!

En la función de *vista* se hacen todas las cosas interesantes: podemos mirar a una base de datos para buscar alguna información. ¿Tal vez el usuario pidió cambiar algo en los datos? Como una carta diciendo "Por favor cambia la descripción de mi trabajo." La *vista* puede comprobar si tienes permiso para hacerlo, actualizar la descripción de tu trabajo y devolver un mensaje: "¡Hecho!". Entonces la *vista* genera una respuesta y Django puede enviarla al navegador web del usuario.

Por supuesto, la descripción anterior está un poco simplificada, pero no necesitas saber todas las cosas técnicas aún. Tener una idea general es suficiente.

Así que en lugar de entrar demasiado en los detalles, simplemente comenzaremos creando algo con Django y aprenderemos todas las partes importantes en el camino!