DIGITALHOUSE

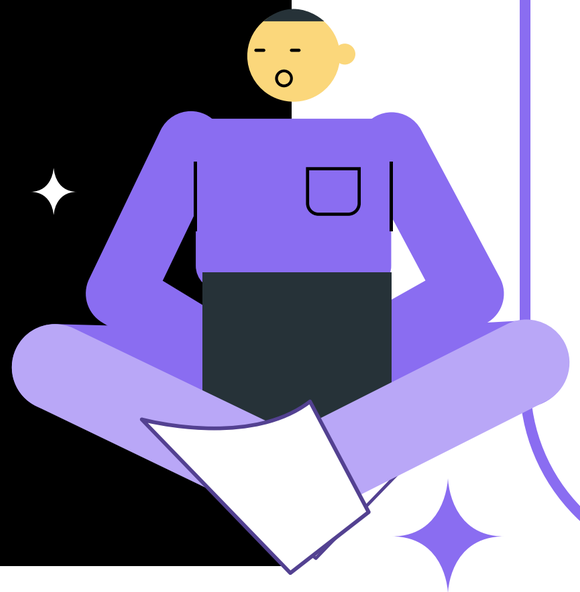
Repaso

Certified Tech Developer

The Ultimate Degree

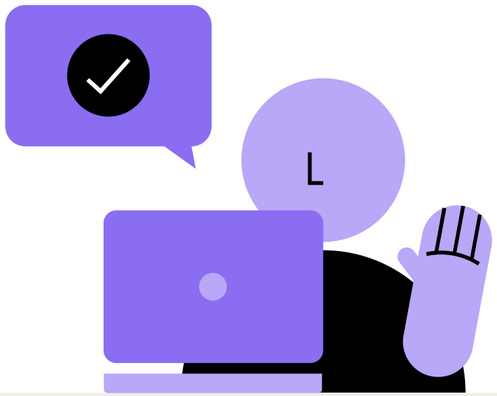
01 OAuth 2.0

OAuth 2.0 es un estándar abierto para la autorización de APIs que nos permite compartir información con otras aplicaciones sin tener que compartir los datos del usuario. Actualmente, es utilizado por grandes empresas, como Google, Facebook, Twitter, entre otras.



¿Qué problema resuelve?

Pensemos en la siguiente situación: si tenemos un back end que expone una API y un front end que la consume, en principio no tenemos grandes problemas para gestionar los usuarios y sus credenciales, ya que somos dueños del 100% del sistema. Sin embargo, llega un momento en el que queremos agregar una funcionalidad que dependa de un tercero, por ejemplo, que nuestra aplicación escriba y publique contenido en la cuenta de Twitter de los usuarios. Para hacer esto, deberíamos pedirle el nombre de usuario y contraseña... Pensemos, ¿quién nos daría esta información? Nadie.

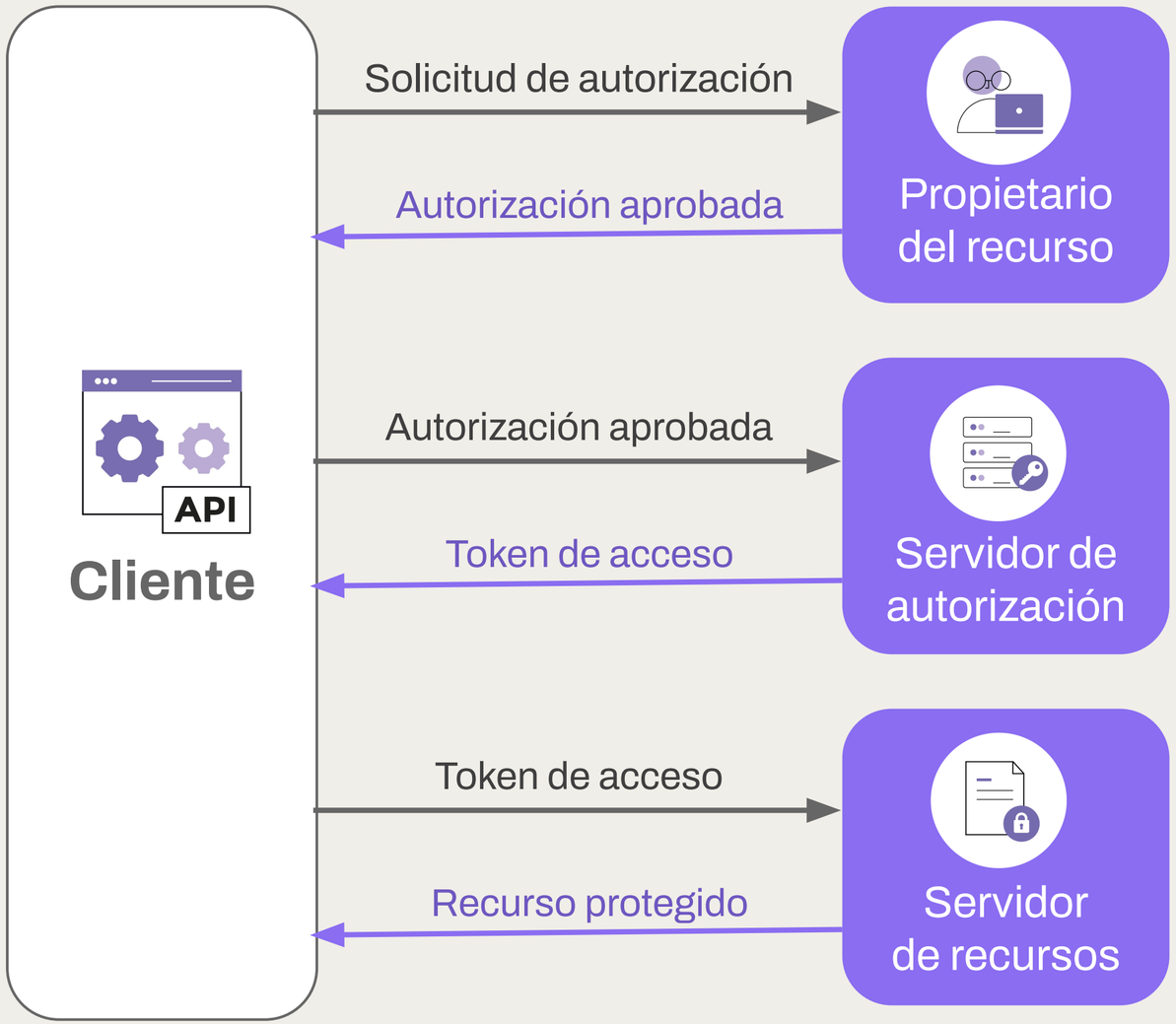


Con OAuth 2.0, el usuario puede autorizar a nuestra aplicación a publicar contenido en Twitter sin tener que compartir su contraseña.

¿Cómo lo hace?

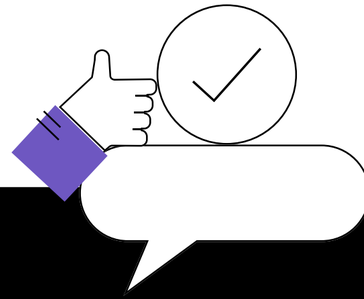
OAuth 2.0 permite al cliente solicitar acceso a recursos con la aprobación del usuario mediante la generación de un token de acceso.

De este modo, OAuth 2.0 permite que una aplicación acceda a recursos proporcionados por otra aplicación, sin solicitar el nombre de usuario y la contraseña del usuario.



Mediante la generación de un token de acceso —una cadena que denota un ámbito específico, un tiempo de vida y otros atributos de acceso—, OAuth 2.0 permite al cliente solicitar acceso a recursos controlados por el propietario del recurso y alojados por el servidor de recursos, y recibir un conjunto diferente de credenciales del propietario del recurso.

Los **tokens de acceso** son emitidos a los clientes de terceros por un servidor de autorización con la aprobación del propietario del recurso. El cliente utiliza el token de acceso para acceder a los recursos protegidos alojados en el servidor de recursos.

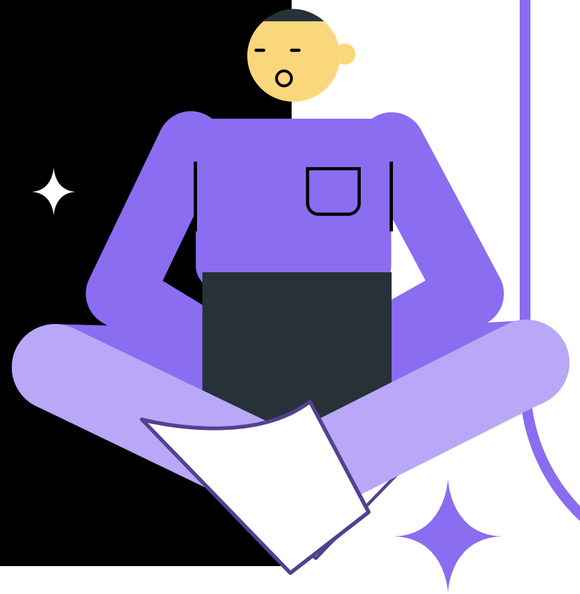


Es un recurso muy presente en nuestro día a día. Posiblemente, ustedes (propietarios del recurso) han accedido a una aplicación (cliente) en la que le han concedido acceso a la información existente en otra aplicación (servidor de recursos), sin que esta nueva aplicación a la que ha accedido recientemente pueda ver toda la información, solo aquella a la que ustedes le han permitido acceder. Se autentifican directamente con un servidor de confianza del servicio (servidor de autorización), que emite las credenciales específicas de la delegación del servicio (token de acceso).

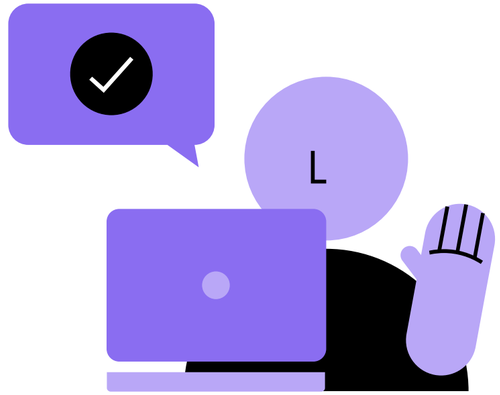
De este modo, **OAuth 2.0** permite que una aplicación acceda a recursos proporcionados por otra aplicación sin solicitar el nombre y la contraseña del usuario.

02 OpenID Connect

Como vimos, con OAuth 2.0 realizamos el proceso de autorización que se realiza gracias a la incorporación de **OpenID Connect**, lo que permite llevar a cabo un inicio de sesión único mediante OAuth. OpenID Connect introduce el concepto de **token ID**. Este es un token de seguridad que permite al cliente verificar la identidad del usuario. El token ID también obtiene información básica del perfil del usuario.



03 OpenID Connect Discovery



Entonces, OpenID Connect es una simple capa de identidad que se coloca sobre el protocolo OAuth 2.0 y permite a los clientes verificar la identidad del usuario final basándose en la autenticación realizada por un servidor de autorización. Además, permite obtener información de perfil básica sobre el usuario final de una manera interoperable, tipo REST.

Esta especificación define un mecanismo para que un tercero de confianza de OpenID Connect descubra el proveedor de OpenID del usuario final y obtenga la información necesaria para interactuar con él, incluyendo sus endpoints de OAuth 2.0. En este punto, entra en juego un servicio de OpenID Connect: **Discovery**. Este facilita el cambio entre diferentes proveedores de OpenID.

DIGITALHOUSE

¡Muchas gracias!