

# Bootcamp Java Developer

Fase 1 - Java Analyst  
Módulo 1

# Scripting

## Scripting

Ya conocemos casi todas las características y modos de trabajo de GitHub. Cualquier grupo o proyecto medianamente grande necesitará, además, **personalizar o integrar GitHub con servicios externos**.

Por suerte, GitHub es bastante *hackeable* en muchos sentidos. **Veremos cómo se usan los enganches (*hooks*) de GitHub y las API para conseguir hacer lo que queremos.**



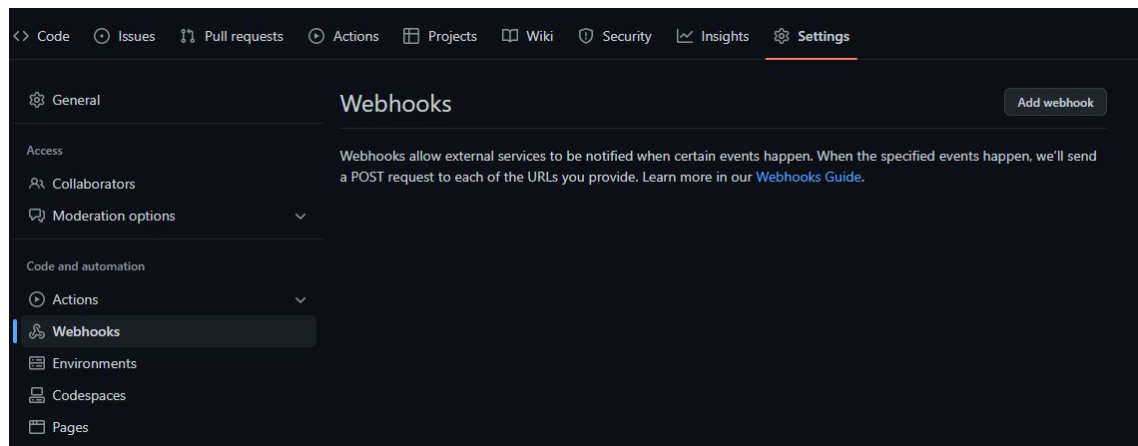
Las secciones ***Hooks*** y ***Services*** de la **página de Administración del repositorio** en Github tienen herramientas que sirven para hacer más simple la interacción entre Github y sistemas externos.



# Servicios

Los enganches y servicios se configuran desde la sección **Settings** del repositorio.

En la opción **Webhooks**:

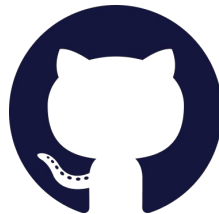


Existen docenas de servicios que se pueden elegir. Muchos de ellos para integrarse en otros sistemas de código abierto o comerciales. Algunos son servicios de integración continua, gestores de incidencias y fallos, salas de charla y sistemas de documentación.

Veremos cómo levantar uno sencillo: el **enganche con el correo electrónico**. Si elegimos “email” en la opción **Add Service** veremos una pantalla de configuración similar a **Configuración de servicio de correo**.

En este caso, si se pulsa el botón **Add service**, la dirección de correo especificada recibirá un correo cada vez que alguien envíe cambios (push) al repositorio.

Los servicios pueden dispararse con muchos otros tipos de eventos, aunque la mayoría, solo se usan para los **eventos de envío de cambios (push)** y hacer algo con los datos del **hook**.



Para integrar algún sistema concreto con GitHub, se debe observar si hay algún servicio de integración ya creado.

Por ejemplo, si utilizas **Jenkins** para ejecutar pruebas de tu código, **puedes activar su servicio de integración** que se disparará cada vez que alguien altere el repositorio.



## Hooks

Si se necesita algo más concreto o se desea integrar con un servicio o sitio no incluido en la lista, se puede usar el **sistema de enganches más genérico**.

Los enganches de GitHub son bastante simples. **Se indica una URL y GitHub enviará una petición HTTP a dicha URL cada vez que suceda el evento que quieras.**

Normalmente, esto funcionará si puedes configurar un pequeño servicio web para escuchar las peticiones de GitHub y luego hacer algo con los datos que se envían.

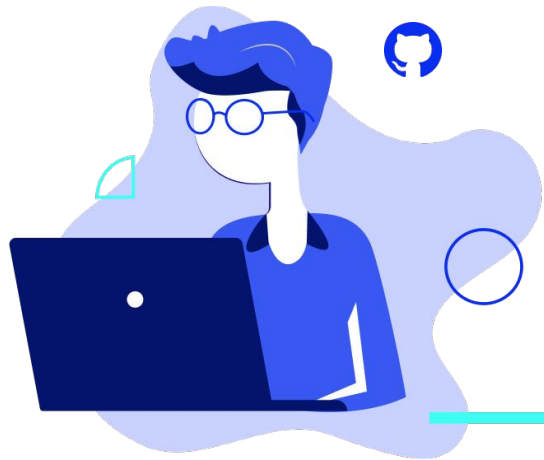
Para **activar un enganche**, pulsa en el botón **Add webhook** de la **sección Services and Hooks**. Esto mostrará una página como **Configuración de enganches web**.

La configuración de un enganche web es bastante simple. Casi siempre basta con **incluir una URL y una clave secreta, y pulsar en Add webhook**.



Hay algunas opciones sobre **qué eventos quieres que disparen el envío de datos** (de forma predeterminada el único evento considerado es el evento push, que se dispara cuando alguien sube algo a cualquier rama del repositorio).

Veamos un pequeño ejemplo de servicio web para manejar un enganche web. Usaremos el entorno **Sinatra de Ruby**, puesto que es conciso y podrás entender con facilidad qué estamos haciendo.





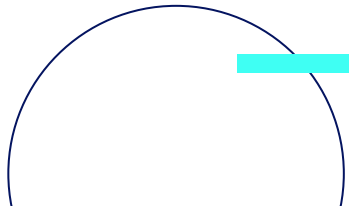
# La API de GitHub

Servicios y enganches sirven para recibir notificaciones “push” sobre eventos que suceden en tus repositorios.

Pero, ¿qué pasa si necesitas más información acerca de estos eventos?, ¿y si necesitas automatizar algo como añadir colaboradores o etiquetar 213 incidencias?

Aquí es donde entra en juego la **API de GitHub**. GitHub tiene montones de llamadas de API para hacer casi cualquier cosa que puedes hacer vía web, de forma **automatizada**.

En esta sección aprenderemos cómo autenticar y conectar a la API, cómo comentar una incidencia y cómo cambiar el estado de un Pull Request mediante la API.



## Uso básico

Lo más básico que se puede hacer es una petición **GET** a una llamada que no necesite autenticación. Por ejemplo, información de solo lectura de un proyecto de código abierto.

Si queremos conocer información acerca del usuario “schacon”, podemos ejecutar algo como:

GET <https://api.github.com/users/schacon>

Hay muchísimas llamadas como esta para obtener información sobre organizaciones, proyectos, incidencias, commits, es decir, todo lo que se puede ver públicamente en la web de GitHub.

Es posible usar la API para otras cosas como ver un archivo Markdown cualquiera o encontrar una plantilla de .gitignore.

## [Sobre la API REST](#)



**¡Sigamos  
trabajando!**