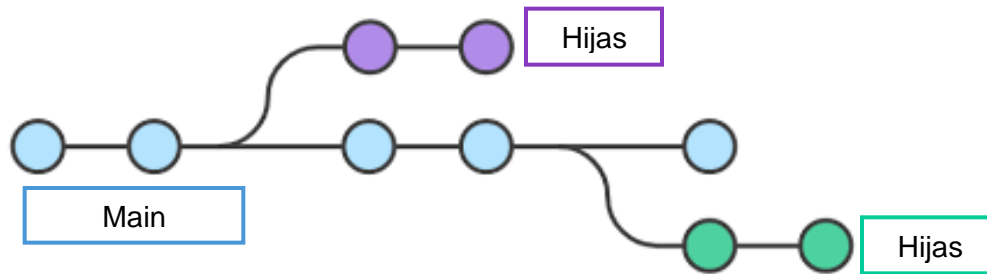
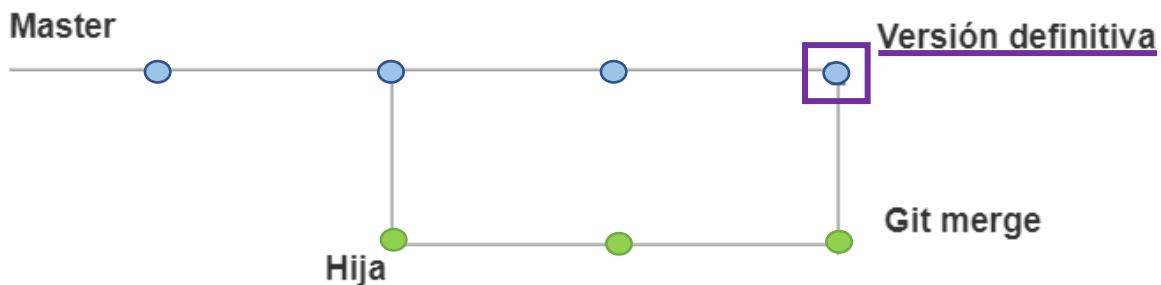


TRABAJANDO CON REPOSITORIOS EN GITHUB



Ramas: Existen dos tipos de ramas, Main y las hijas. Main es la rama principal, y sus hijas son ramas alternas que salen de la principal, dónde generalmente se ejecutan cambios que aún están en proceso de desarrollo o no han pasado por una revisión, esto para evitar problemas en Main.

Git Merge: Ocurrirá en la rama que estoy ubicada generando que el último commit de esa rama sea principal:



Conflicts: Los conflictos de fusión ocurren cuando fusionas ramas que tienen confirmaciones en competencia, y Git necesita tu ayuda para decidir qué cambios incorporar en la fusión final. Git a menudo puede resolver las diferencias entre ramas y fusionarlas automáticamente.

Pull Request: Los pull request permiten no solo llevar de forma más ordenada las tareas en la etapa del desarrollo, sino también crear propuestas o cambios que puedan ser integrados posteriormente a dicho proyecto.

Básicamente un pull request es una petición para integrar nuestras propuestas o cambios de código a un proyecto.

Examen 4

Audrey Gerardo Juárez

Fork: Un fork es una copia de un repositorio que administras. Los fork le permiten realizar cambios en un proyecto sin afectar el repositorio original. Puede obtener actualizaciones o enviar cambios al repositorio original con solicitudes de incorporación de cambios.

Rebase: El comando git rebase te permite cambiar fácilmente una serie de commits, modificando el historial de tu repositorio. Puede reordenar, editar commits juntas. Por lo general, usaría git rebase para: Editar mensajes de commits anteriores. Combine varias commits en una.

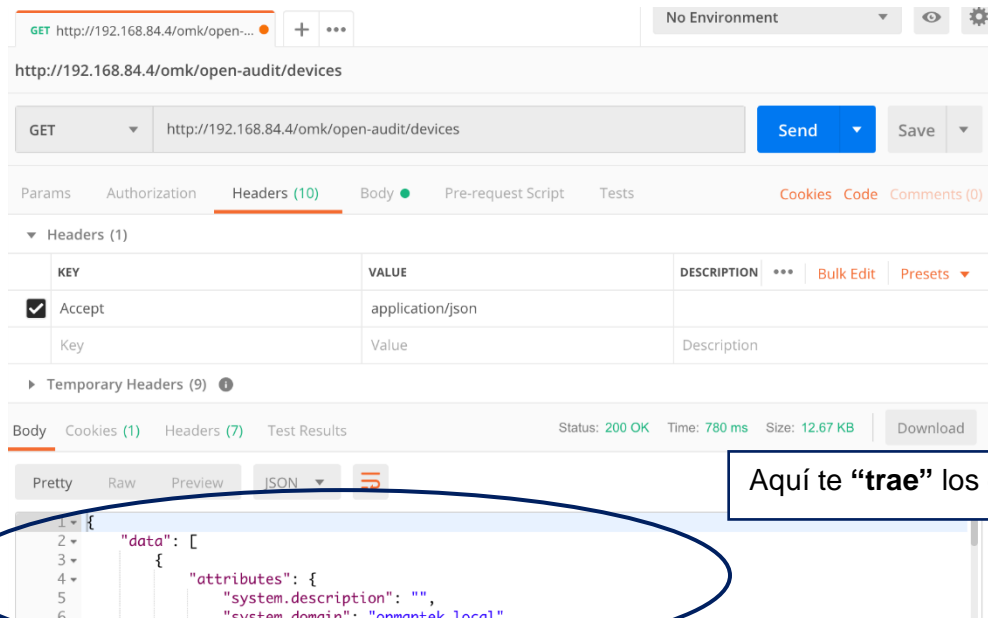
Stash: Si ha guardado cambios que aún no está listo para confirmar, puede ocultar los cambios para más adelante. Cuando oculta los cambios, los cambios se eliminan temporalmente de los archivos y puede optar por restaurarlos o descartarlos más adelante.

Clean: Git clean es hasta cierto punto un comando de 'deshacer'. Git clean puede considerarse complementario a otros comandos como git reset y git checkout. Mientras que estos otros comandos funcionan en archivos agregados previamente al índice de seguimiento de Git, el comando git clean funciona en archivos sin seguimiento. Los archivos sin seguimiento son archivos que se han creado dentro del directorio de trabajo de su repositorio pero que aún no se han agregado al índice de seguimiento del repositorio mediante el comando git add.

Cherry-pick: Puede seleccionar una confirmación en una rama para crear una copia de la confirmación con los mismos cambios en otra rama. Si confirma cambios en la rama incorrecta o desea realizar los mismos cambios en otra rama, puede seleccionar la confirmación para aplicar los cambios a otra rama.

TRABAJANDO CON REPOSITORIOS EN GITHUB

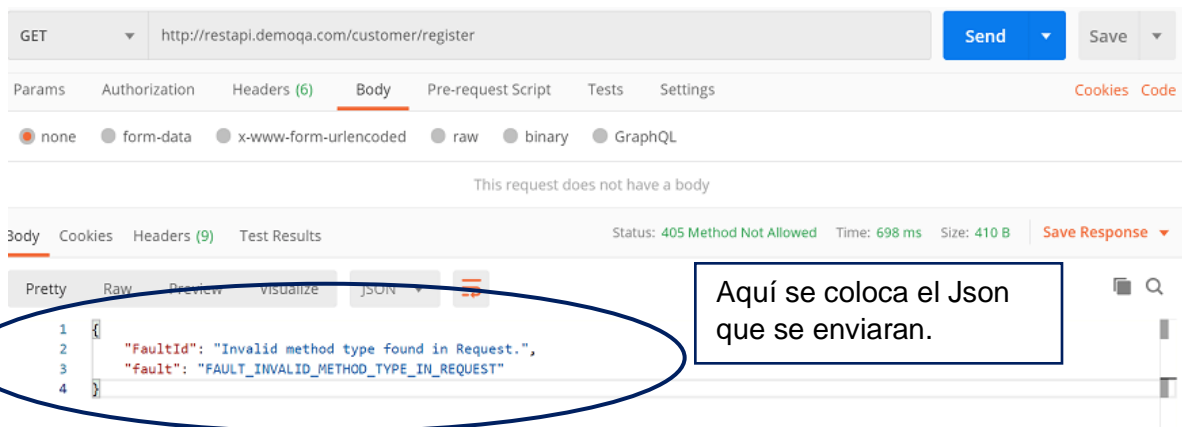
GET: El método GET se usa para recuperar información del servidor dado usando un URI dado. Las solicitudes que utilizan GET solo deben recuperar datos y no deben tener ningún otro efecto sobre los datos.



The screenshot shows a REST client interface with a GET request to `http://192.168.84.4/omk/open-audit/devices`. The response status is 200 OK. The response body is displayed in JSON format, showing a list of devices with attributes like system.description and svstem.domain. A blue oval highlights the JSON response, and a callout box points to it with the text "Aquí te 'trae' los datos."

```
{
  "data": [
    {
      "attributes": {
        "system.description": "",
        "svstem.domain": "oomantek.local"
      }
    }
  ]
}
```

POST: Una solicitud POST se utiliza para enviar datos al servidor, por ejemplo, información del cliente, carga de archivos, etc. mediante formularios HTML.



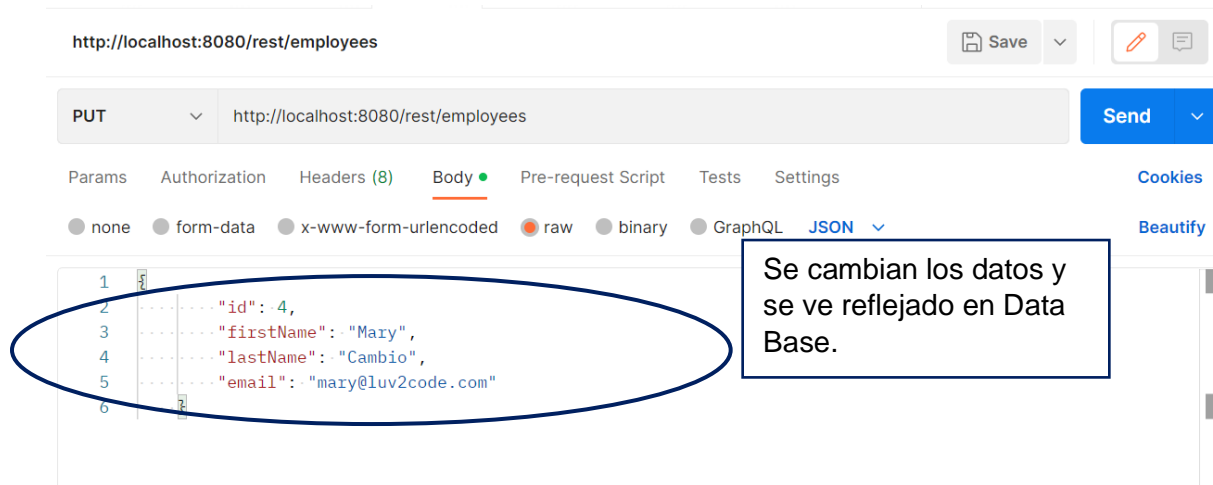
The screenshot shows a REST client interface with a GET request to `http://restapi.demoqa.com/customer/register`. The response status is 405 Method Not Allowed. The response body is displayed in JSON format, showing an error message. A blue oval highlights the JSON response, and a callout box points to it with the text "Aquí se coloca el Json que se enviaran."

```
{
  "FaultId": "Invalid method type found in Request.",
  "fault": "FAULT_INVALID_METHOD_TYPE_IN_REQUEST"
}
```

Examen 4

Audrey Gerardo Juárez

PUT: Se utiliza para enviar datos a un servidor para crear/actualizar un recurso. Reemplaza todas las representaciones actuales del recurso de destino con el contenido cargado.



DELETE: El método DELETE elimina el recurso especificado.