1. Ejecute el programa **xamp** e inicie los servicios de **Apache y MySQL** (Los servicios pueden tardar un poco en iniciar).
2. Vaya a la ruta **C:/xampp/htdocs y entre en la carpeta donde tiene guardado el proyecto.**
3. Pulse *click* derecho en cualquier parte dentro de esa carpeta y seleccione la opción *Git Bash Here* para abrir una consola de **git** que automáticamente se posicionara en la ruta de la carpeta del proyecto. Este paso debe hacerlo dos veces para que pueda tener dos consolas.
4. En una de las consolas ejecute el siguiente comando para ingresar a la carpeta del Back-End:

* cd gems\_backend

1. En la consola con la que ejecuto el paso 3, ejecute el siguiente comando para cambiar a la rama dev perteneciente a este repositorio puesto que no se debería trabajar sobre la rama master:

* git checkout dev

1. Una vez posicionado en la rama dev como se explico en el paso 4, ejecute el siguiente comando para iniciar el servidor y posteriormente esperar el mensaje Database is connected:

* npm run start:dev

1. En la segunda consola que abrió en el paso 2, ejecute el siguiente comando para ingresar a la carpeta gems\_web-frontend:

* cd gems\_web-frontend

1. Después de entrar a la carpeta del frontend como se esxplico en el paso 6, ejecute el siguiente comando para iniciar el servidor del Front-End:

* npm start

y listo, espere a que abra el navegador y podrá ver el proyecto en marcha.

**Nota:**

1. Cuando haga algún registro desde el Front-End, puede confirmar que si se guardaron los cambios visitando la ruta de phpmyadmin: <http://localhost/phpmyadmin/>

Selecciona el nombre de la base de datos correspondiente al proyecto (gems:db), elija la tabla donde deberían estar los datos y vera una lista de registros realizados.

1. Si ingreso un dato desde el Front-End y quiere confirmar al mismo tiempo si se almaceno en la tabla de la BD, debe recargar la página de phpmyadmin ya que nos cambios no se visualizan en tiempo real.
2. Puede usar el comando *history* en la consola de *Git Bash* si se le olvida el nombre de algún comando que haya usado recientemente.
3. Siempre realice un pull antes de ejecutar el proyecto o antes de enviar cambios al repositorio remoto.
4. Para realizar un pull en git (actualizar el repositorio local con el remoto que se encuentra en GitHub) utilice el comando:
   * Git pull origin nombre\_rama

Donde nombre\_rama será el nombre de la rama local que quiere actualizar. En caso de que le aparezca un error indicando que debe realizar un merge, utilice el siguiente comando:

* git reset --hard && git pull origin nombre\_rama

1. Para realizar un push (enviar cambios al repositorio remote que se encuentra en GitHub):
   * git push origin nombre\_rama

Donde nombre\_rama sera la rama a la que se quieren enviar los cambios. Si al usar este comando no se le permite enviar el push, utilice el comando:

* + git push origin nombre\_rama -f