**Capacitación shamir:**

Se descargaron las siguientes tecnologías :

Navicat , vagrant virtualbox, GitBash

Se estableció usar el repositorio gitbucket.

Se cambia el uso de xaamp con el servidor apache por homesteady.

**Navicat :** Navicat  es un administrador gráfico de base de datos y un software de desarrollo producido por PremiumSoft CyberTech Ltd. para [MySQL](https://es.wikipedia.org/wiki/MySQL" \o "MySQL), [MariaDB](https://es.wikipedia.org/wiki/MariaDB" \o "MariaDB), [Oracle](https://es.wikipedia.org/wiki/Oracle_Database), [SQLite](https://es.wikipedia.org/wiki/SQLite" \o "SQLite), [PostgreSQL](https://es.wikipedia.org/wiki/PostgreSQL" \o "PostgreSQL) y [Microsoft SQL Server](https://es.wikipedia.org/wiki/Microsoft_SQL_Server). Cuenta con un Explorador como interfaz gráfica de usuario soportando múltiples conexiones para bases de datos locales y remotas. Su diseño está pensado para satisfacer las diferentes necesidades de un amplio sector del publico; desde administradores y programadores de bases de datos a diferentes empresas que dan soporte y o comparten información con clientes o socios.

**Vagrant:** es una herramienta para la creación y configuración de entornos de desarrollo virtualizados.

**Virtualbox :** Es una herramienta para la simulación y virtualización de sistemas operativos .

**Git :** Es una herramienta para el control de versionamientos de un archivo permite un ambiente virtual y confiable para el desarrollo de una aplicación.

**Comandos básicos de git :**

* git fetch:

Descarga los cambios realizados en el repositorio remoto.

* git merge *<nombre\_rama>*:

Impacta en la rama en la que te encuentras parado, los cambios realizados en la rama “nombre\_rama”.

* git pull:

Unifica los comandos *fetch* y *merge* en un único comando.

* git commit -am "<mensaje>":

Confirma los cambios realizados. El “mensaje” generalmente se usa para asociar al *commit* una breve descripción de los cambios realizados.

* git push origin *<nombre\_rama>*:

Sube la rama “nombre\_rama” al servidor remoto.

* git status:

Muestra el estado actual de la rama, como los cambios que hay sin commitear.

* git add *<nombre\_archivo>*:

Comienza a trackear el archivo “nombre\_archivo”.

* git checkout -b *<nombre\_rama\_nueva>*:

Crea una rama a partir de la que te encuentres parado con el nombre “nombre\_rama\_nueva”, y luego salta sobre la rama nueva, por lo que quedas parado en ésta última.

* git checkout -t origin/*<nombre\_rama>*:

Si existe una rama remota de nombre “nombre\_rama”, al ejecutar este comando se crea una rama local con el nombre “nombre\_rama” para hacer un seguimiento de la rama remota con el mismo nombre.

* git branch:

Lista todas las ramas locales.

* git branch -a:

Lista todas las ramas locales y remotas.

* git branch -d *<nombre\_rama>*:

Elimina la rama local con el nombre “nombre\_rama”.

* git push origin :*<nombre\_rama>*:

Elimina la rama remote con el nombre “nombre\_rama”.

* git remote prune origin:

Actualiza tu repositorio remoto en caso que algún otro desarrollador haya eliminado alguna rama remota.

* git reset --hard HEAD:

Elimina los cambios realizados que aún no se hayan hecho *commit*.

* git revert *<hash\_commit>*:

Revierte el *commit* realizado, identificado por el “hash\_commit”.

**Laravel :**  es un [framework](https://es.wikipedia.org/wiki/Framework" \o "Framework) de [código abierto](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto) para desarrollar aplicaciones y servicios web con [PHP](https://es.wikipedia.org/wiki/PHP) 5. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.[2](https://es.wikipedia.org/wiki/Laravel#cite_note-introduccion-2)

Fue desarrollado por Taylor otwett.

El entorno de desarrollo se mueve entre Linux y Windows.

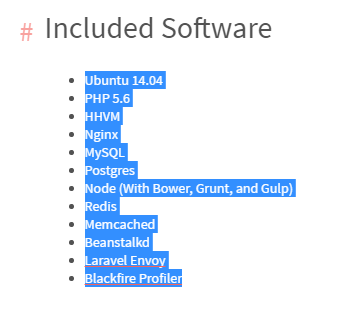
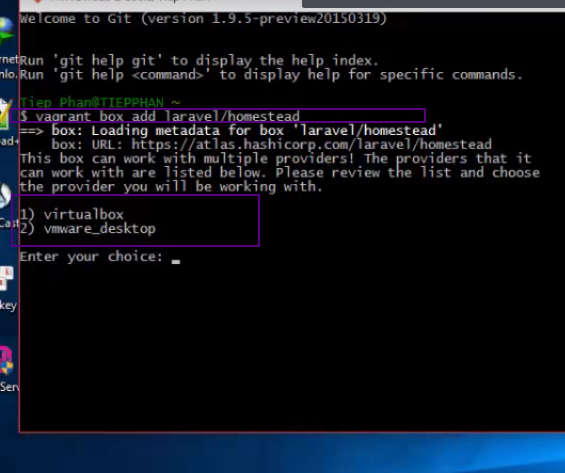
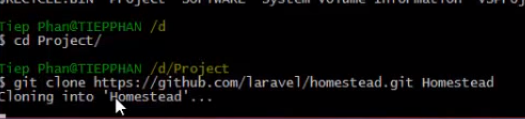
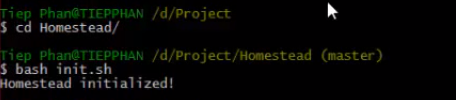
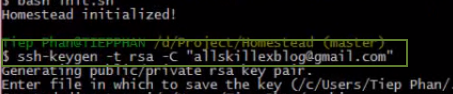
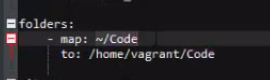
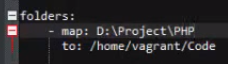
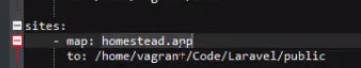
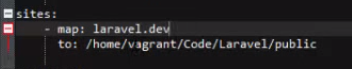
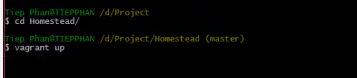
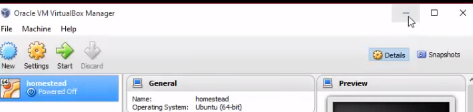
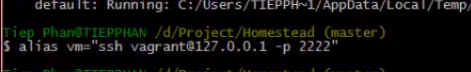
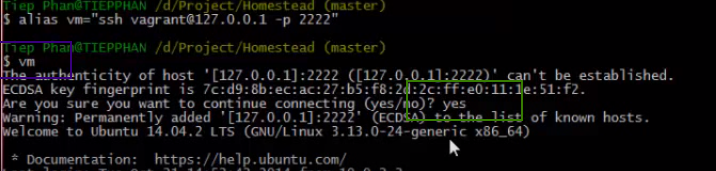
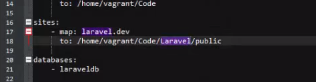
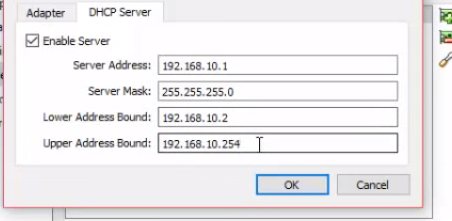
**Laravel Homestead**

Laravel homestead es el adecuado y recomendado entorno de trabajo para el desarrollo de aplicaciones “laravel”.

Con homestead podemos proveer a laravel de maquinas virtualizadas y un soporte de servidor web sin la necesita de la instalación de un recurso adicional.

**Instalación del entorno de trabajo:**

Guía paso a paso:

* Descargamos el git de la pagina http: git-scm.com
* Descargamos en vagrant de http: vagrantup.com
* Descargamos virtual box de http: virtualbox.org.
* Instalamos los archivos dentro de nuestra pc .
* Software dentro de vagrant :
* 
* Dentro del git bash excribimos el comando vagrant box add laravel/homestead.
* A continuación seleccionamos virtual box.
* 
* Instalamos el repositorio dentro de nuestro proyecto , ubicamos la carpeta y usamos el comando git clone.
* 
* Después dentro de la carpeta del archivo , dentro de su carpeta homestead iniciamos el servicio .
* 
* En la ruta C:\Users\nomb\_usuario\.homestead En el archivo Homestead.yaml verificamos la siguiente información
* Generamos una llave ssh 
* Mapeamos a la carpeta del proyecto
* 
* 
* Cambiamos los sitios
* 
* 
* En databases cambiamos a laravel db.
* Nos vamos a la siguiente ruta C:\Windows\System32\drivers\etc
* En el archivo hosts verificamos la ip de laravel.
* 
* Debe ser la misma en el archivo host y el archivo Homestead.yml
* Una vez que nos aseguramos de eso levantamos el servicio de vagrant dentro del proyecto.
* 
* Una vez hecho eso aparecerá dentro del virtual box el homestead.
* 
* Se hace la conexión ssh a la pc con el siguiente comando.
* 
* Se establece una conexión con Ubuntu
* 
* Se escriben las siguientes líneas para llegar al composer.
* 
* Creamos el proyecto y lo nombramos
* 
* Este nombre debe ser igual en este punto del achivo homestead.yml
* 
* Debe ponerse a instalar laravel dentro del proyecto.
* Procedemos a configurar el virtual box .
* Vamos a preferences 🡪 network 🡪HostOnly networks 🡪 Details , cambiamos la dirección ipv4 por la correspondiente en el archivo host de laravel.dev
* Cambiamos de esta forma la información dhcp
* 

**Una vez creado pasos para iniciar el servidor**

Dentro de la carpeta Homestead ponemos vagrant up.

En la misma carpeta ponemos $ alias vm="ssh vagrant@127.0.0.1 -p 2222"

Usamos vm para iniciar Ubuntu.

Esto seria todo.

Con exit cerramos la sesión Ubuntu.

Para apagar el servidor lo hice directamente del virtual box.

**Comandos de git**

Git init me permite instanciar la función git dentro de una carpeta para subirla al repositorio.

Git Add. Coge todo los documentos de esa carpeta y los prepara para subirlos al repositorio.

Git status Me permite verificar todos los archivos que van a subirse.

Git commit –am”name commit” Me permite hacer un comit de los archivos (si hago git status debe aparecer vacio “nothing to commit”).

Git log

Git remote add origin // Direccion para subir archivos , me permite subir los commit.

Git push –u origin –all

Me realiza un push de todo lo que hay en el proyecto