LARAVEL

Video #2

Requisitos:

* PHP
* Xaamp
* Composer modo global

Comandos:

* *composer global require laravel/installer* (instalador de laravel)
* *laravel new blog* (crear nuevo proyecto en Laravel con el nombre "blog")
* *cd blog* (entrar al directorio “blog”)
* *php artisan serve* (correr el proyecto)

Video #3

Configuración:

Config/App/

**'timezone'** => **'America/Guayaquil'**,

**'locale'** => **'es'**,

**'faker\_locale'** => **'es\_EC'**,

.env

**DB\_DATABASE**=**inconcerto**

Video #4

Si tienes un repositorio ya creado en Github (y trabajaste en el equipo y quieres vincular a un repo de GitHub), puedes usar la terminal o la aplicación de GIT y ejecutar los comandos:

* Abres un Git bash en el directorio blog
* ls -a (dir /D si estás en Windows, dir /A para ver todos los ocultos, dir /AD para ver directorios ocultos)
* si no aparece el .git : git init
* ls -a (dir /D si estás en Windows, ya aparece)
* git checkout -b "*Una rama*"
* git branch (para verificar que estás en la rama que vas a trabajar)
* git remote add *origin* (url de nuestro repo)
* git remote -v (aparece que tenemos fetch para actualizacion y push para subir datos)
* git status (ver el estado de los directorios)
* git add . (añade)
* git commit -m “Mensaje”
* git push *origin* *“Una rama”* (hacemos el push hacia la rama master de nuestro repo)
* Actualiza la pestaña de GitHub para evidenciar los cambios

Si tienes un repositorio ya creado en Github (que te has clonado, y lo seguirás trabajando), puedes usar la terminal o la aplicación de GIT y ejecutar los comandos:

* Abres un Git bash en el directorio blog
* Git clone (Url del repo)
* Ls -a (dir /D si estás en Windows, ya aparece, debe aparecer uno de .git)
* Git checkout -b n-“*rama*” (el n- no aplica si la rama es “dev”)
* Git Branch (Aseguraremos de que estás en esa rama)

Trabajamos todo lo que tengamos que trabajar, y luego:

* Git add .
* Git commit -m “#n *nombre de la tarea*” (donde n es el número de la tarea)
* Git push origin n-“*rama*” (donde n es el número de la tarea)

Te devolverá un link para que hagas pull request, le clickeas. Te redirecciona al Github, y lo siguiente que debemos hacer es conectar con una tarea:

* Click en “*conect with an Issue*” y elegimos la tarea a la que queremos conectar
* Click en crear un Pull Request

Luego debemos esperar a la retroalimentación, ver el video desde el min (43:41), y de no haber ningún cambio

* Click en “Merge”

Seguridad en ramas

* Settings -> Branches -> escoger “dev” como default -> actualizar -> añadir reglas -> paterno escribimos: master -> enmarcar pull request review before merge -> crear regla
* Settings -> Branches -> escoger “dev” como default -> actualizar -> añadir reglas -> paterno escribimos: dev -> enmarcar pull request review before merge -> crear regla

**Crear una etiqueta:**

Zenhub -> Issues -> Labels -> New Label

**Crear tareas:**

Zenhub -> New Issue -> Configurar como se requiera (*Titulo Tarea*)

**Crear un Milestone**

Zenhub -> Boton desplegable junto a New Issue -> Title: Sprint “n” -> crear

**Commits**

Comenzar siempre con un numeral y un número “n (*Titulo Tarea*)”

**Pull Requests**

Luego de hacer el git push, das click al enclace que lanza el git en la termianl