**TDG guía instalación.**

En esta sección se hablará sobre 2 opciones para instalar el software proyectogrado en una máquina local para proceder a su desarrollo y testeo.

1. **XAMPP (recomendada)** : instalar XAMPP desde la dirección <https://www.apachefriends.org/download.html> usando la versión adecuada para el equipo de trabajo, después de haber instalado el software y haber configurado el php ini del mismo para poder trabajar con bases de datos MySQL se procede a guardar la carpeta del proyecto en la carpeta htdocs (puede guardarse en una subcarpeta y en niveles inferiores de carpetas de ser necesario), después de esto se debe tener instalado en el equipo el “composer” (en caso contrario descargarlo https://getcomposer.org/download/ y seguir las instrucciones de instalación) este permite la correcta funcionalidad de los paquetes de laravel además de la instalación de nuevos paquetes para el proyecto.
2. **PHP directo:** tener instalado y configurado el lenguaje de php directamente en la maquina, además de configurar el “composer” como se mencionaba anteriormente también se debe modificar el php ini para no solo correr el MySQL, sino además también los servicios anexos a laravel, además de la configuración del servidor local

**TDG Comprensión proyecto**

El proyecto se divide en varias subcarpetas que se generan automáticamente por laravel para mantener un orden en el software. Cada carpeta tiene una función especifica

**app:** carpeta donde se alojan la mayoría de los archivos relacionados al desarrollo del proyecto, en esta misma se alojan carpetas como la de “Console” que permite modificar las tareas del kernel, “Exceptions” para ejercer un control sobre las excepciones o errores que se puedan presentar y sean previstos, “Http” carpeta donde de alojan los controladores de las diferentes tareas del software (en estos archivos se alojan las funciones que serán llamadas por las API), al igual que una carpeta “Middleware” donde se alojan todas las funciones de control de usuarios básicas de laravel, “Imports” donde se alojan las funciones para importar archivos al software, “Models” donde se alojan los diferentes modelos de almacenamiento de datos para el software y como serán utilizados en la base de datos, “Providers” donde se almacenan las funciones que involucran directamente al software y a los paquetes descargados.

**bootstrap:** carpeta que almacena los archivos de Bootstrap para poder darle un comportamiento responsive al renderizado web del software

**config:**  En esta carpeta se almacenan los archivos de permisos y usos de los diferentes paquetes y tareas que vienen por defecto incluidas en laravel

**database:** Esta carpeta almacena 3 subcarpetas para controlar el funcionamiento de la base de datos, en “factories” se almacenan los datos de default para poder hacer registros en la base de datos de los diferentes modelos en caso de ser necesarios, “migrations” en donde se alojan las tareas de modificion y/o creación de la base de datos de manera automática, “seeders” carpeta en la cual se almacena la información para poder abrir un espacion inicial en la base de datos

**public:** En esta se alojan carpetas como “css” donde se almacenan los estilos para el renderizado, “images” para almacenar las imágenes que se van a usar en el mismo, “js” donde se almacenan los scripts de javascript que se ejecutan para contralar el funcionamiento del front en el software, además de varios archivos de configuración que se relacionan con la parte visual y renderizado en web

**resources:** Esta carpeta también posee carpetas como las anteriores, pero son ya directamente relacionados a las funciones y vistas en las que se van a renderizar los modelos de la información mencionada anteriormente, además de la carpeta “lang” donde se almacenan diferentes textos para renderizar en el software, “sass” para diferentes configuraciones de estilos y “views” donde se alojan las diferentes carpetas alusivas a las funciones y modelos que renderizan la forma como se ve la información.

**storage:** carpeta en la cual se almacenan la información de los registros que se hacen en el software, también llamados logs

**test:** carpeta en la cual se almacenan los archivos para hacer los tests de las funcionalidades

**vendor:** carpeta exclusiva de laravel para almacenar los paquetes descargados y demás funciones propias

esto es una breve descripción de las funciones y contenidos de las carpetas del proyecto, además de estas se encuentran una variedad de archivos como los “composer” que como se menciono antes son el controlador de paquetes para el software, los archivos ml para guardar información, además de los archivos con las excepciones de seguirle el rastreo al repositorio, el archivo con mayor relevancia de todos estos es el “.env” debido a que es de este que el software saca la configuración para poderse ejecutar, anexo también se encuentra un ejemplo del mismo para darse una guía para como configurarlo. Además de este archivo se incluyen un ProyectoGrado.xlsx el cual es una version Excel de la vista de la base de datos con sus diferentes tablas, en el cual se evidencia que los títulos en rojo se refiere a información necesaria para el correcto funcionamiento del software, además de una copia de la base de datos de prueba proyectogrado.sql, un archivo schedules.cvs el cual es la plantilla para subir la información directamente a la base de datos en caso de ser necesario para la programación, el archivo ProgramaHorarios.cvs que es la plantilla para subir la información mediante el software a la base de datos.

**Tener muy presente que los archivos mencionados en el último párrafo deben ser tenidos en cuenta a la hora de ejecutar cualquier acción en el software debido a que el desconocimiento de estos podría afectar no solo problemas en su ejecución sino errores en su funcionamiento.**

**Conclusiones**

* Eliminar la relación en la tabla de subjects al programa en específico para así eliminar la redundancia en la misma tabla, una posible solución será crear una tabla intermedia en la cual se lleve la conexión entre un programa y las materias asociadas
* Modificar la funcionalidad de subir CVS totalmente, debido a que esta trae asociada un modelo que se presta para errores en la subida de información a la base de datos, además de entorpecer el proceso, usando funciones internas del mismo PHP como array\_map y str\_getcvs para así limitar la dependencia de paquetes externos y aumentar la eficiencia del programa
* Modificar la arquitectura de la información eliminando la tabla de teachers que realmente no tiene sentido debido a que el rol se puede asignar al usuario, además de habilitar la opción de tener múltiples roles, además de asignar tareas adecuadas a cada rol
* La importancia de una buena arquitectura de datos se ve reflejada en el tiempo de uso y mejoras que se van proponiendo en pos de mejorar la funcionalidad y la eficiencia en lado de