



INSTITUTO TECNOLÓGICO DE IZTAPALAPA

INGENIERIA EN SISTEMAS COMPUTACIONALES

INFORME TÉCNICO DE RESIDENCIAS PROFESIONALES

TEMA:

Lumen (micro-framework, es decir una versión reducida de framework completo llamado [Laravel](#).)

LUGAR DE REALIZACIÓN:

Instituto tecnológico de Iztapalapa

ASESOR INTERNO:

Parra

Hernández

Abiel

Tomas

PRESENTA:

Anastasio Mendoza Erick – 171080406

33%

Carmona Guillen Samuel Jesús - 171080160

33%

Mena Gres Jesús Said – 171080101

33%

CIUDAD DE MÉXICO

06/2021

ÍNDICE GENERAL

Índice	iii
Resumen	iv
Introducción	5
Objetivo General.....	6
Objetivos Específicos.....	6
Justificación	7
CAPÍTULO 1. Marco de Referencia	9
1.1 Compilador	9
1.2 LLVM	10
1.3 MLIR	10
1.4 Lumen.....	11
1.5 Laravel.....	12
1.6 PHP	13
Metodología de trabajo	14
Desarrollo e implementación	16
Resultados.....	22
Conclusiones	25
Fuentes de información	26

RESUMEN

Se dará a conocer la importancia del compilador “**LUMEN**” el cual es bastante útil para crear micro servicios de páginas web, ya que tiene código Erlang y está optimizado para utilizar lenguaje PHP de tal forma que se complementan debido a sus bibliotecas y diseño para desarrollar sistemas, además Erlang se compone de una base de datos el cual nos facilitara herramientas para los proyectos que se generarán.

Debido al gran desempeño a la hora de utilizar un sistema complejo nos ayudara a tener un mejor resultado.

Palabras clave:

Código Erlang, Lenguaje PHP, LUMEN, microprocesadores.

INTRODUCCIÓN

Este documento será de utilidad para conocer los concepto, herramientas y propósitos de Lumen, ya que es un compilador que se utiliza para los procesos de MILR, por consiguiente, el mismo se divide en compilador, interprete y tiempos de ejecución, Ya que estaremos desarrollando aplicaciones para micro servicios y API's, tiene una implementación alternativa en Erlang/OTP.

Lumen no es solo un compilador, sino también un tiempo de ejecución. Está formado por dos partes:

Un compilador de Erlang para código nativo para un objetivo determinado (x86, ARM, WebAssembly)

Un tiempo de ejecución de Erlang, implementado en Rúst., que proporciona la funcionalidad principal necesaria para implementar OTP, para que podamos proseguir y tengamos el mismo concepto, tendremos que responder preguntas de índole básico, ya que en general la introducción a los compiladores puede ser de manera, poco practica y el usuario que está leyendo este documento debe contar con las habilidades cognitiva necesarias para una mejor comprensión de los temas que estaremos abarcando a lo largo de este documento.

OBJETIVO GENERAL

Servirá para hacer un compilador para Erlang - Elixir y otros lenguajes BEAM que se pueden compilar en diferentes entornos en lugar de ejecutarse en una máquina virtual.

OBJETIVOS ESPECÍFICOS

Objetivo Específico 1.

Desarrollar una solución marcada para interpretar microservicios de ayuda paulatina en PHP

Objetivo Específico 2.

Dar a conocer microservicios que ofrece lumen y ver la importancia (analizar) que este puede tomar con sus lenguajes aplicados en una página web

Objetivo Específico 3.

Explicar los microservicios de lumen e indicar las propiedades indispensables de este

Objetivo Específico 4.

Conocer más acerca de lo que nos puede ofrecer LUMEN y donde se puede aplicar.

JUSTIFICACION

Determinar cómo **LUMEN**, tiene un gran valor en el mundo de los compiladores, ya que es parte de la familia **BEAM**, donde se puede ejecutar en diferentes entornos de máquinas virtuales, donde los principales objetivos de este proyecto, son eliminar las bibliotecas que dejan sin código activo de tal forma que en el momento en el que contribuyamos con la materialización de ejecución y solución de problemas con este tipo de entorno de programación, será más fácil el escribirlo, tendrá una mayor compilación estática de código binario. Dándole una prioridad demasiado importante en el caso de los navegadores donde la mayoría de compiladores carece de efectividad en el transcurso de algún proyecto.

Cuando hablamos de un lenguaje de programación moderno con implicaciones a futuro debemos tener en cuenta lo que necesitamos para poder desarrollarlo y hacer el tipo de experimentación, ya que en ocasiones podríamos divagar un poco de la idea principal y en este caso **LUMEN** no busca hacer eso sino acoplarse a los nuevos navegadores para ocupar el tan llamativos *WebAssembly* (aquí fue traducción de Word), que es un lenguaje de programación de bajo nivel como el famoso lenguaje ensamblador, entonces teniendo en cuenta cada una de las ventajas y desventajas que se tienen en dicho lenguaje de programación podemos darnos una idea de cómo se puede ocupar con **C**, **C++** y **Rúst**.

“WebAssembly tiene grandes implicaciones para la plataforma web — provee una forma de correr código escrito en múltiples lenguajes en la web a una velocidad casi nativa, con aplicaciones cliente corriendo en la web que anteriormente no podrían haberlo hecho.”

Lumen tiene la capacidad para trabajar de la mano con cada uno de estos lenguajes de programación y como anteriormente ya habíamos mencionado es capaz de acoplarse a diferentes compiladores para que al final del día podamos realizar un trabajo en un tiempo más corto y se pueda aprovechar el código que se está programando de manera óptima, eficaz, proactiva, con saber esto y que BEAM encaja demasiado bien con los patrones que se tienen en la bibliotecas (React), ya que son cadenas árbol cada una que es responsable de lo que pase con nuestra aplicación, sin embargo si una llega a fallar hace que las demás partes de la aplicación se bloqueen. y de ahí poder visualizar la facilidad y efectividad de este Al igual se da a conocer un servicio específico como lo es el Erlang el cual nos da las herramientas fundamentales como una base de datos integrada y también su lenguaje en nativo lo cual como se mencionó nos da una mejor compatibilidad con el lenguaje de PHP para su mejor funcionamiento.

Mientras que el objetivo de los compiladores es obtener una traducción del programa fuente a otro lenguaje, los intérpretes tienen como objeto la obtención de los resultados del programa. Para ello deben realizar dos tareas: analizar su entrada y llevar a cabo las acciones especificadas por ella. La parte de análisis puede realizarse de manera idéntica a como se lleva a cabo en los compiladores. Es la parte de síntesis la que se diferencia sustancialmente. En el caso de la interpretación, se parte del árbol de sintaxis abstracta y se recorre, junto con los datos de entrada, para obtener los resultados.

Dar conocer la importancia del compilador “**LUMEN**” el cual es bastante útil para crear micro servicios de páginas web, ya que tiene código Erlang y está optimizado para utilizar lenguaje PHP de tal forma que se complementan debido a sus bibliotecas y diseño para desarrollar sistemas, además Erlang se compone de una base de datos el cual nos facilitara herramientas para los proyectos que se generarán.

Debido al gran desempeño a la hora de utilizar un sistema complejo nos ayudara a tener un mejor resultado.

MARCO TEORICO

Tema: Lumen (micro-framework, es decir una versión reducida de framework completo llamado Laravel.)

1.1 Compilador

Un compilador es un pequeño programa informático, que se encarga de traducir (compilar) el código fuente de cualquier aplicación que se esté desarrollando. En pocas palabras, es un software que se encarga de traducir el programa hecho en lenguaje de programación, a un lenguaje de máquina que pueda ser comprendido por el equipo y pueda ser procesado o ejecutado por este.

Un concepto un poco más elaborado es el siguiente: Un compilador es un programa que convierte o traduce el código fuente de un programa hecho en lenguaje de alto nivel, a un lenguaje de bajo nivel (lenguaje de máquina).

La importancia de los compiladores radica mucho en que, sin estos programas, no existiría ninguna aplicación informática, ya que son la base y/o parte más importante de la programación en cualquier plataforma.

Existen varios tipos de compiladores, los cuales se adecuan a los requerimientos de los programadores. Podemos destacar los compiladores cruzados, compiladores optimizadores, compiladores de una sola pasada y de varias pasadas, compiladores JIT y los conocidos intérpretes, que traducen el código del programa en tiempo real.

1.2 LLVM

El Proyecto LLVM es una colección de tecnologías de cadena de herramientas y compiladores modulares y reutilizables. A pesar de su nombre, LLVM tiene poco que ver con las máquinas virtuales tradicionales.

LLVM se ha convertido en un proyecto general que consta de una serie de subproyectos, muchos de los cuales están siendo utilizados en la producción por una amplia variedad de proyectos comerciales y de código abierto, además de ser ampliamente utilizados en la investigación académica . El código del proyecto LLVM tiene la licencia "Licencia Apache 2.0 con excepciones LLVM".

Además de los subproyectos oficiales de LLVM, hay una amplia variedad de otros proyectos que utilizan componentes de LLVM para diversas tareas . A través de estos proyectos externos, puede usar LLVM para compilar Ruby, Python, Haskell, Rúst., D, PHP, Pure, Lúa y varios otros lenguajes. Una de las principales fortalezas de LLVM es su versatilidad, flexibilidad y capacidad de reutilización, razón por la cual se está utilizando para una variedad tan amplia de tareas diferentes: todo, desde hacer compilaciones JIT ligeras de lenguajes integrados como Lúa hasta compilar código Fortran para super ordenadores.

También podemos definir que es una infraestructura para desarrollar compiladores, escrita a su vez en el lenguaje de programación C++, que está diseñada para optimizar el tiempo de compilación, el tiempo de enlazado, el tiempo de ejecución y el "tiempo ocioso" en cualquier lenguaje de programación que el usuario quiera definir. Implementado originalmente para compilar C y C++, el diseño agnóstico de LLVM con respecto al lenguaje, y el éxito del proyecto han engendrado una amplia variedad de lenguajes

1.3 MLIR

El proyecto MLIR es un enfoque novedoso para construir una infraestructura de compilador extensible y reutilizable. MLIR tiene como objetivo abordar la fragmentación del software, mejorar la compilación para hardware heterogéneo, reducir significativamente el costo de construir compiladores específicos de dominio y ayudar a conectar compiladores existentes.

MLIR está diseñado para ser un IR híbrido que puede soportar múltiples requisitos diferentes en una infraestructura unificada. Por ejemplo, esto incluye:

- La capacidad de representar gráficos de flujo de datos (como en Tensor Flow), incluidas las formas dinámicas, el ecosistema operativo extensible por el usuario, las variables de Tensor Flow, etc.
 - Las optimizaciones y transformaciones se realizan normalmente en tales gráficos (por ejemplo, en Grappler).
 - Capacidad para representar operaciones específicas de un objetivo, por ejemplo, operaciones de alto nivel específicas de un acelerador.
 - Cuantización y otras transformaciones de gráficos realizadas en un gráfico de aprendizaje profundo.
- Entre otras.

MLIR es un IR común que también admite operaciones específicas de hardware. Por lo tanto, cualquier inversión en la infraestructura que rodea a MLIR (por ejemplo, el compilador transmite ese trabajo) debería generar buenos rendimientos; muchos objetivos pueden utilizar esa infraestructura y se beneficiarán de ella.

El proyecto MLIR define una representación intermedia (IR) común que unifica la infraestructura necesaria para ejecutar modelos de aprendizaje automático de alto rendimiento en Tensor Flow y en marcos de trabajo de AA similares. Este proyecto incluirá la aplicación de técnicas de HPC, junto con la integración de algoritmos de búsqueda, como el aprendizaje por refuerzo. El objetivo de MLIR es reducir el costo para incorporar hardware nuevo y mejorar la usabilidad para los usuarios de Tensor Flow.

1.4 LUMEN

Sobre lo investigado en lumen podemos destacar que este es una gran herramienta para desarrollar un compilador usado de forma para base de datos sencillas y avanzadas dentro de páginas web el cual es muy compatible con el lenguaje php lo cual lo hacer más interactivo y una forma más entera para desarrollo de páginas web

Bases teóricas:

- ¿Qué es un micro servicio?
- ¿Para qué sirve una API?
- ¿Dónde encuentro un compilador?
- ¿Cuándo puedo ocupar el compilador?

Concepto clave:

Micro servicios, Lenguaje PHP, FRAMEWORK

Lumen es un proyecto para hacer un compilador para **Erlang / Elixir** y otros lenguajes BEAM que se pueden compilar en diferentes entornos en lugar de ejecutarse en una máquina virtual.

Los microservicios son tanto un estilo de arquitectura como un modo de programar software. Con los microservicios, las aplicaciones se dividen en sus elementos más pequeños e independientes entre sí. A diferencia del enfoque tradicional y monolítico de las aplicaciones, en el que todo se compila en una sola pieza, los microservicios son elementos independientes que funcionan en conjunto para llevar a cabo las mismas tareas.

Lumen es el hermano pequeño de Laravel: un micro-framework rápido y ligero para escribir API's RESTful. Con sólo un poco de código, puede usar Lumen para crear una API RESTful segura y extremadamente rápida.

LLVM requiere Cmake, un compilador C/C++ y Python se recomienda que también se instale Ninja y CCache para que la compilación sea significativamente más rápida especialmente en reconstrucciones posteriores

Una vez teniendo en cuenta estas preguntas básicas podremos determinar cuándo si, y cuando no ocupar este tipo de tecnología (programación)

Debido a la gran demanda de actividades de los diferentes sectores donde se encuentra organizado el área de cómputo, por consiguiente, esta etapa es obtener una representación de la entrada que nos permita realizar la síntesis o la interpretación con comodidad. La representación que nosotros utilizaremos es la que se llama árbol de sintaxis abstracta.

1.5 LARAVEL

Laravel es un popular framework de PHP. Permite el desarrollo de aplicaciones web totalmente personalizadas de elevada calidad.

Laravel es un framework PHP. Es uno de los framework más utilizados y de mayor comunidad en el mundo de Internet.

Como framework resulta bastante moderno y ofrece muchas utilidades potentes a los desarrolladores, que permiten agilizar el desarrollo de las aplicaciones web.

Laravel pone énfasis en la calidad del código, la facilidad de mantenimiento y escalabilidad, lo que permite realizar proyectos desde pequeños a grandes o muy grandes. Además, permite y facilita el trabajo en equipo y promueve las mejores prácticas.

El framework Laravel trabaja con una arquitectura de carpetas avanzada, de modo que promueve la separación de los archivos con un orden correcto y definido, que guiará a todos los integrantes del equipo de trabajo y será un estándar a lo largo de los distintos proyectos. Por supuesto, dispone también de una arquitectura de clases también muy adecuada, que promueve la separación del código por responsabilidades.

Laravel es un framework más nuevo que muchos de sus competidores. Ha venido desde atrás con una progresión muy fuerte, consolidándose en los últimos años como el framework PHP más utilizado.

Esta progresión se debe a varios factores. Obviamente, es fundamental la cantidad de funcionalidad que nos ofrece ya lista, sin embargo, este motivo no es el más importante, dado que muchos otros framework ofrecen un soporte también muy completo. Sí es verdad que la cantidad de módulos ha ido incrementándose continuamente y a día de hoy es posible que supere a otras alternativas populares, pero su adopción por parte de la comunidad ha sido muy relevante incluso antes de ello.

1.6 PHP

PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML.

PHP se utiliza para generar páginas web dinámicas. Recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc.

Lo que distingue a PHP de JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente como si fuera una página web estática. El cliente recibirá los resultados que el servidor devuelve después de interpretar el código PHP, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido.

PHP es el lenguaje más popular para el desarrollo backend, es decir, es la tecnología más usada en lo que respecta a lenguajes del lado del servidor.

El motivo de este éxito no es solo la facilidad con la que PHP permite desarrollar páginas, sino también formar parte del stack de tecnologías más habitual en Internet: LAMP (Linux, Apache, MySQL y PHP).

METODOLOGIA

Se realizó una video conferencia en equipo donde nos pudimos dar cuenta que una metodología ágil completa es Extreme Programming en la cual Se trata de un entorno de desarrollo cuyo objetivo es promover la aplicación de prácticas de ingeniería apropiadas para la creación de software.

Aunque SCRUM le quito la popularidad este sigue siendo óptimo para la creación de software por lo que ofrece:

El diseño del programa suele ser simple y basado en la funcionalidad del sistema y se lleva a cabo durante la planificación y finalización de la **iteración**.

La programación del **software** se hace siempre en pareja, lo que se llama programar a dos manos. Se asegura con este método que al menos un programador conozca y controle una actividad de otro donde pueda ser revisado. La ventaja es que se produce mejor código que en base a un programador, aunque la dificultad de la misma sea mayor.

El código es de todos, con el desarrollo de las pruebas automáticas y la programación a dos manos se incluye también la posibilidad de que cualquiera pueda añadir y retocar parte del código, aunque eso sí, deba ser un estilo común y cuyo resultado sea como si sólo lo hubiera hecho una persona.

El **Extreme Programming** tiene como gran ventaja el de la programación organizada y planificada para que no haya errores durante todo el proceso. Los programadores suelen estar satisfechos con esta metodología. Es muy recomendable efectuarlo en proyectos a corto plazo.

5 VALORES FUNDAMENTALES DE Extreme Programming

1. Comunicación

El objetivo es **romper las barreras entre negocio y desarrollo**. Para ello, la programación XP promueve que todos los requisitos sean comunicados y trabajados con el equipo y no mediante documentación.

2. Simplicidad

Esta metodología pone el foco en **codificar las necesidades de hoy, no las de un futuro**. Además, también se simplifica el diseño para agilizar el desarrollo y facilitar el mantenimiento. Para conservar la simplicidad hay que mantener la refactorización del código, así podremos preservar el código simple a medida que va creciendo.

3. Feedback

Una de las mayores ventajas de que el cliente esté integrado en el proyecto es que su

opinión sobre el estado de este lo podemos conocer en tiempo real.

4. Respeto

El respeto mutuo es fundamental para que un equipo pueda trabajar de forma eficiente y ofrecer un buen rendimiento. Implica desde que un desarrollador no realice modificaciones que puedan tener un impacto negativo en el trabajo de un compañero hasta la forma de llegar al cliente.

5. Valentía

Diseñar y programar para hoy y no para mañana implica valentía en la metodología XP, así como reconocer los errores tan pronto como se detecten.

DESARROLLO E IMPLEMENTACIÓN

El intento realizar la clonación y compilación del proyecto dado por el profesor, pero nos encontramos con dificultades al intentar instalar RUST.

```

C:\Users\lfvnb\Downloads\rustup-init (3).exe
C:\Users\lfvnb\.cargo
This can be modified with the CARGO_HOME environment variable.
The cargo, rustc, rustup and other commands will be added to
Cargo's bin directory, located at:
C:\Users\lfvnb\.cargo\bin
This path will then be added to your PATH environment variable by
modifying the HKEY_CURRENT_USER/Environment/PATH registry key.
You can uninstall at any time with rustup self uninstall and
these changes will be reverted.
Current installation options:

  default host triple: x86_64-pc-windows-msvc
  default toolchain:   stable (default)
  profile:             default
  modify PATH variable: yes

1) Proceed with installation (default)
2) Customize installation
3) Cancel installation
>1
info: profile set to 'default'
info: default host triple is x86_64-pc-windows-msvc
warning: Updating existing toolchain, profile choice will be ignored
info: syncing channel updates for 'stable-x86_64-pc-windows-msvc'
info: default toolchain set to 'stable-x86_64-pc-windows-msvc'
stable-x86_64-pc-windows-msvc unchanged - rustc 1.53.0 (53cb7b09b 2021-06-17)

Rust is installed now. Great!

To get started you may need to restart your current shell.
This would reload its PATH environment variable to include
Cargo's bin directory (C:\Users\lfvnb\AppData\Local\Temp\rustup-init-x86_64-pc-windows-msvc\cargo-install0unLXD\).
Press the Enter key to continue.
  
```

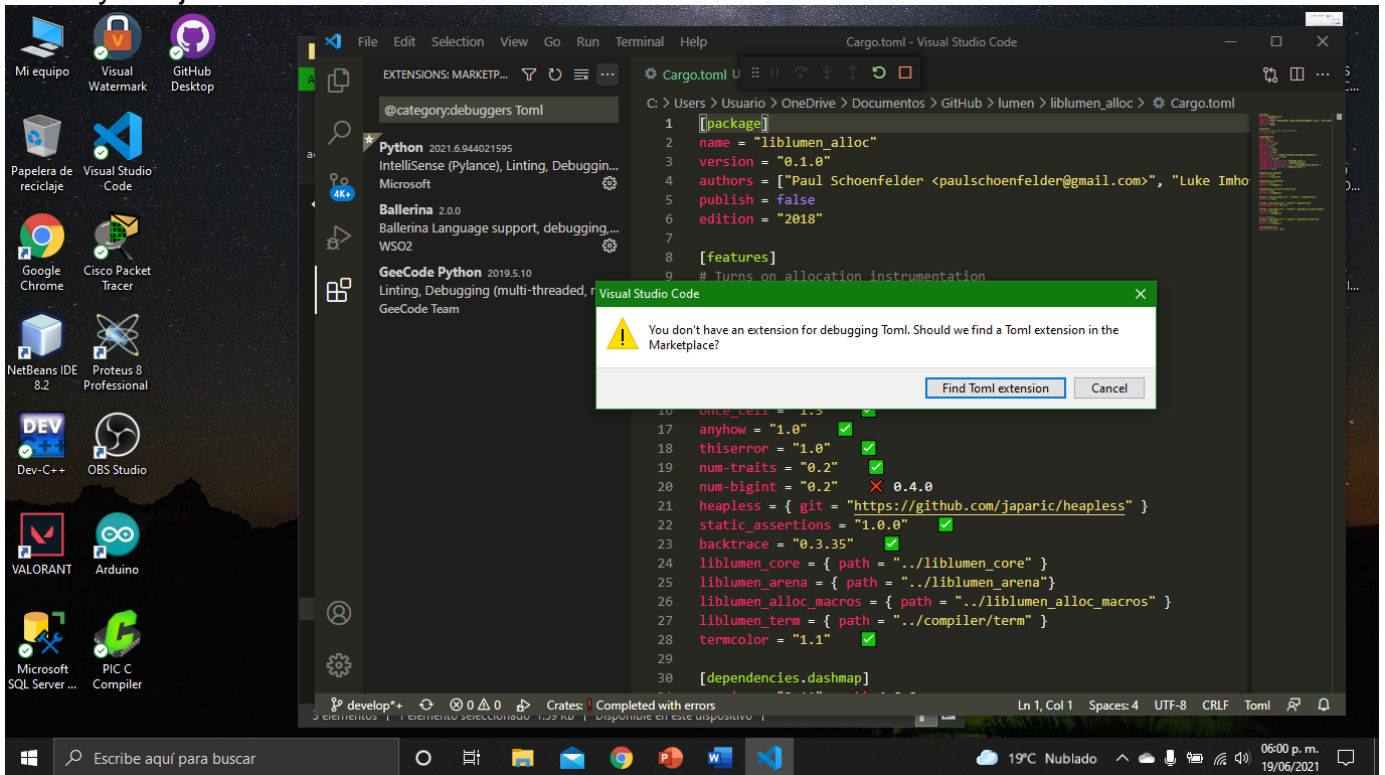
```

To learn more, run the command again with --verbose.
warning: build failed, waiting for other jobs to finish...
error: linker `link.exe` not found
  |
  = note: El sistema no puede encontrar el archivo especificado. (os error 2)
note: the msvc targets depend on the msvc linker but `link.exe` was not found
note: please ensure that VS 2013, VS 2015, VS 2017 or VS 2019 was installed with the Visual C++ option
error: aborting due to previous error
error: linker `link.exe` not found
  |
  = note: El sistema no puede encontrar el archivo especificado. (os error 2)
note: the msvc targets depend on the msvc linker but `link.exe` was not found
note: please ensure that VS 2013, VS 2015, VS 2017 or VS 2019 was installed with the Visual C++ option
error: aborting due to previous error
error: failed to compile `cargo-make v0.34.0`, intermediate artifacts can be found at `C:\Users\USER\AppData\Local\Temp\rustup-init-x86_64-pc-windows-msvc\cargo-install0unLXD\`
Caused by:
  build failed
C:\WINDOWS\system32>
  
```

En estos intentos de instalación a todos los integrantes del equipo nos marcaba este error por lo

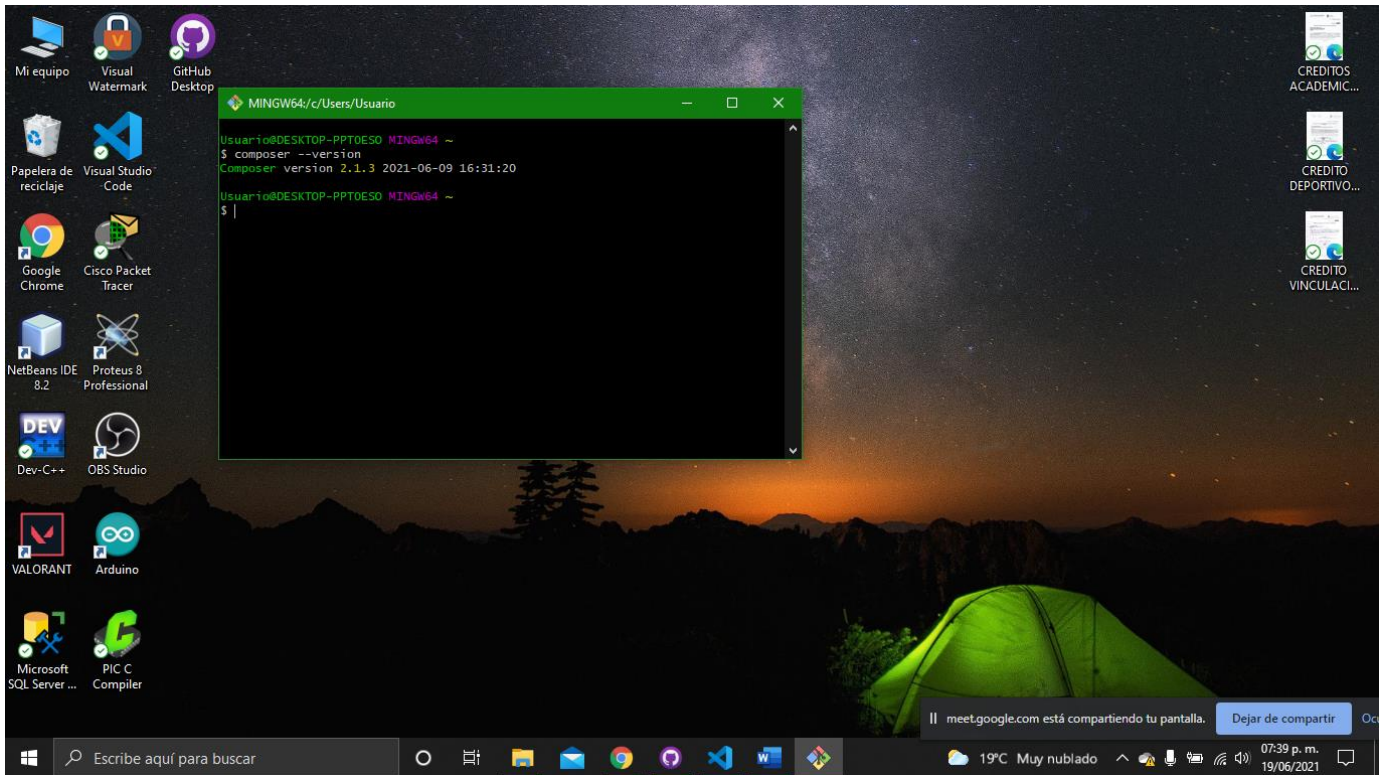
cual decidimos investigar otro método por el cual realizar al practica

Y cuando se realizó la instalación al fin, en Visual Studio, al intentar ver cada archivo y que generaba, nos pedía instalar alguna extensión, pero al instalar todas las que arrojaba, seguía sin correr y arrojando el mismo error de extensión.

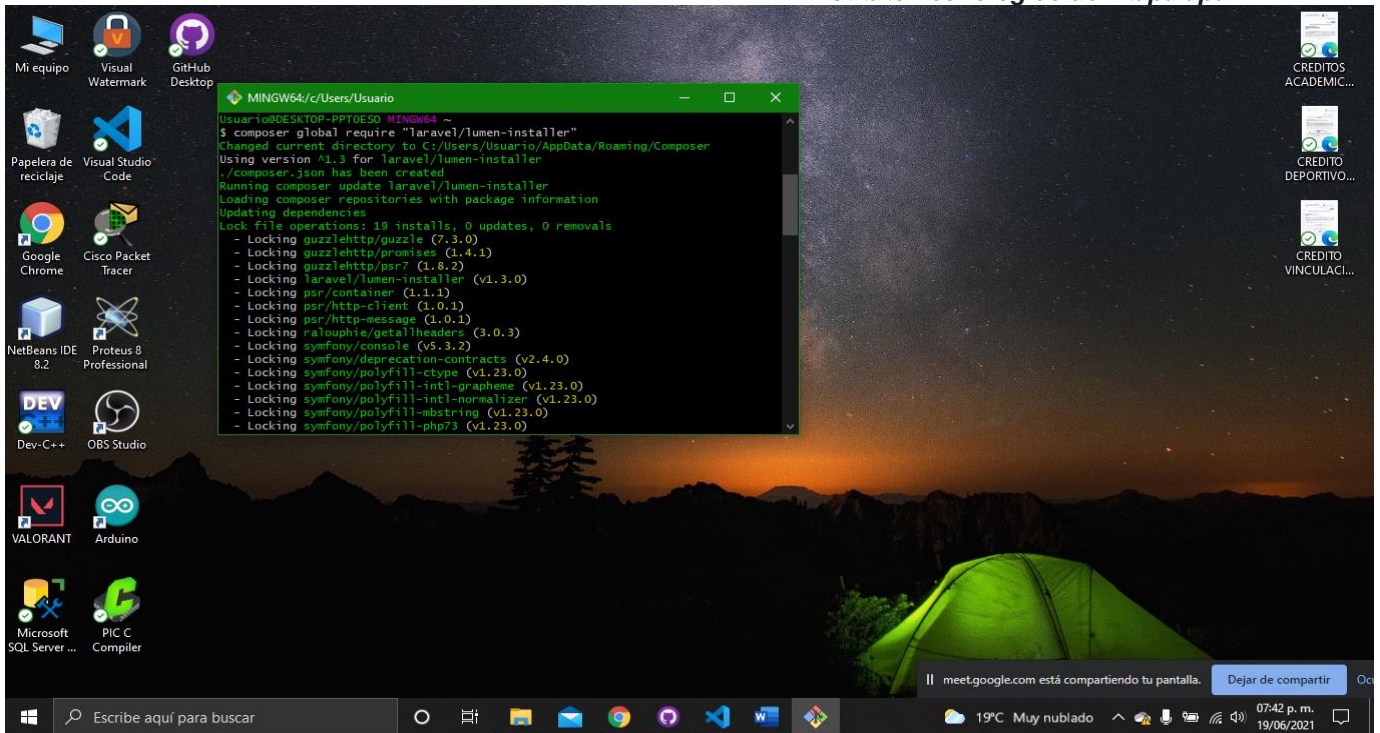


Por lo que se tomó la decisión de realizar PHP con LUMEN.

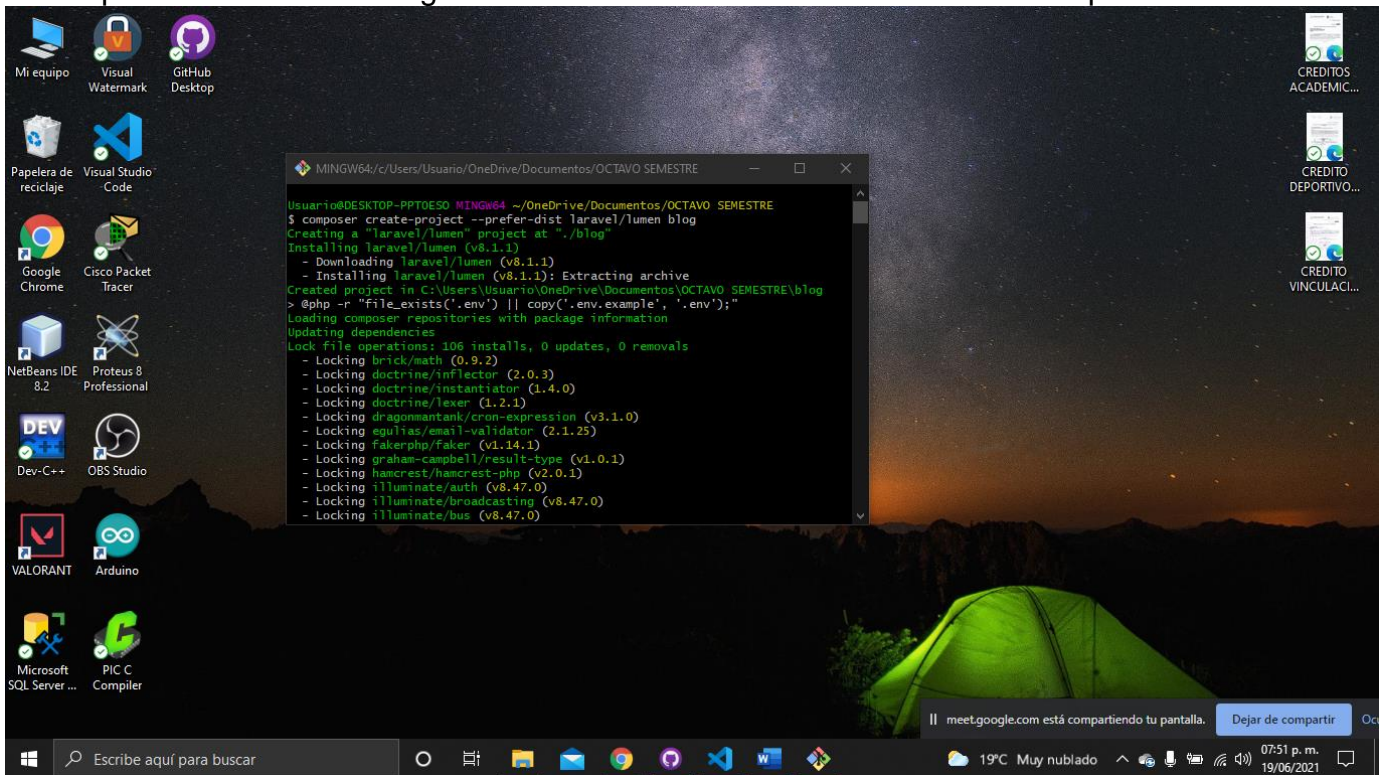
Al querer hacer la práctica de otro modo tuvimos que instalar varias herramientas para la facilitación de esta una de estas y que se ve en la captura es la de GIT BASH el cual nos ayuda a correr en consola de una forma diferente y más fácil al abrir carpetas como se podrá observa a continuación



En esta parte ocupamos comando directos de la página de LUMEN los cuales nos ayudaba a instalarlos de una forma rápida y eficiente ya que su instalación no duro casi nada.

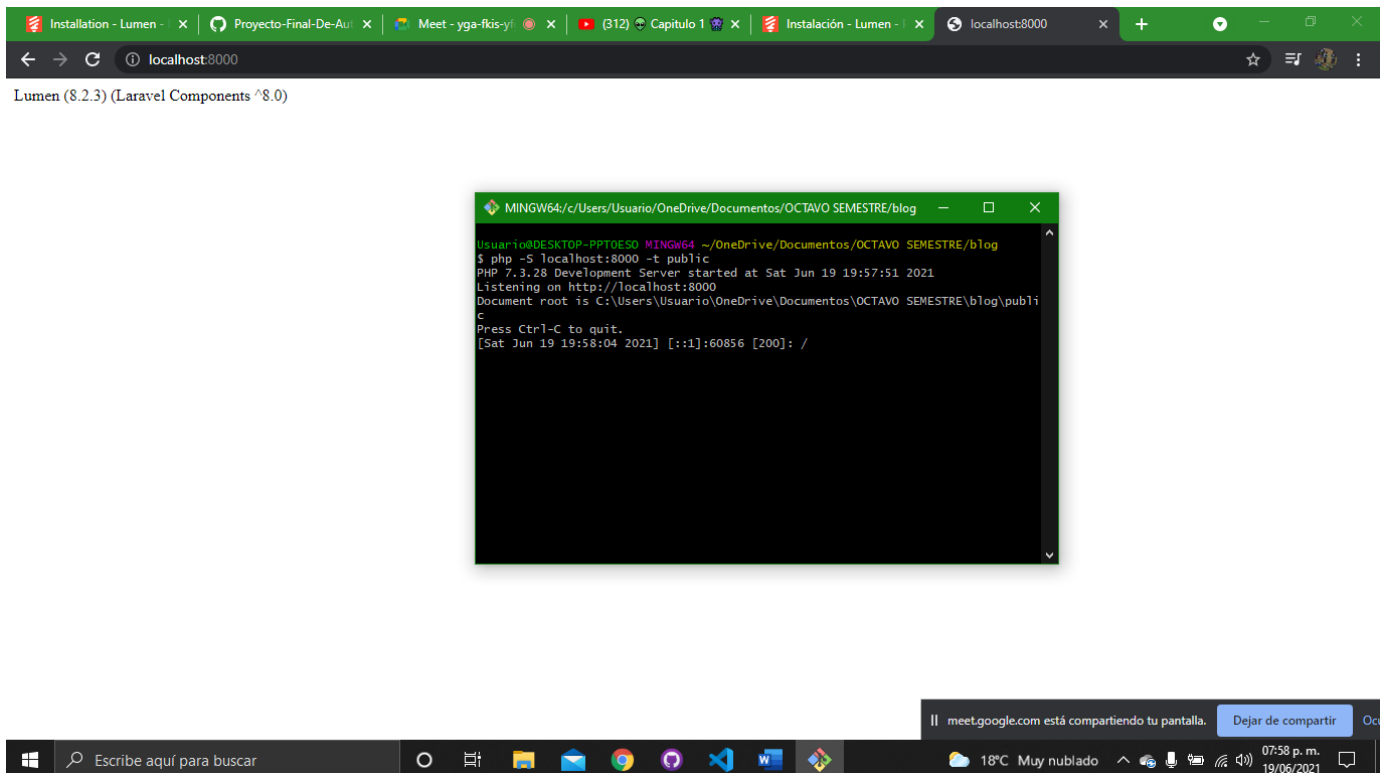


En esta parte fue donde hicimos uso de la aplicación de GIT BASH, para abrir los comandos de consola directo en una carpeta sin necesidad de poner la ruta como se haría normalmente, gracias a esto pudimos instalar el blog de LUMEN de una forma eficiente en una carpeta recién creada



Una vez instalado el BLOG donde estará todos los archivos de LUMEN que se pueden modificar para ser utilizarlos de mejor forma volvemos a nuestro CMD donde volvemos abrir la carpeta

donde se instaló el blog e ingresamos este comando `php -S localhost:8000 -t public` El cual nos dará acceso a lo que está dentro de la BD de LUMEN y también nos dará un enlace para abrir como PHP, en el cual sin modificarlo nos dará lo que tiene por defecto en los comandos que es lo que se muestra en la página de atrás.



Comando que se muestra en la parte de arriba abierto desde visual estudio para poder tener una fácil manipulación de estos códigos

```

1 <?php
2
3 /** @var \Laravel\Lumen\Routing\Router $router */
4
5 /*
6  * Application Routes
7  *
8  * Here is where you can register all of the routes for an application.
9  * It is a breeze. Simply tell Lumen the URIs it should respond to
10  * and give it the Closure to call when that URI is requested.
11  */
12
13 $router->get('/', function () use ($router) {
14     return $router->app->version();
15 });
16
17
18
19

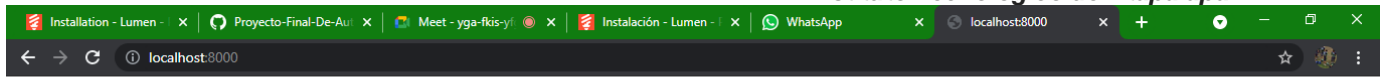
```

Jugando con este pequeño código logramos poner lo que queríamos en la pantalla ya que este texto que utilizamos en esa línea de código se guarda en la BD de datos el cual de ahí jalara el texto y lo mostrara como en el resultado de abajo.

```

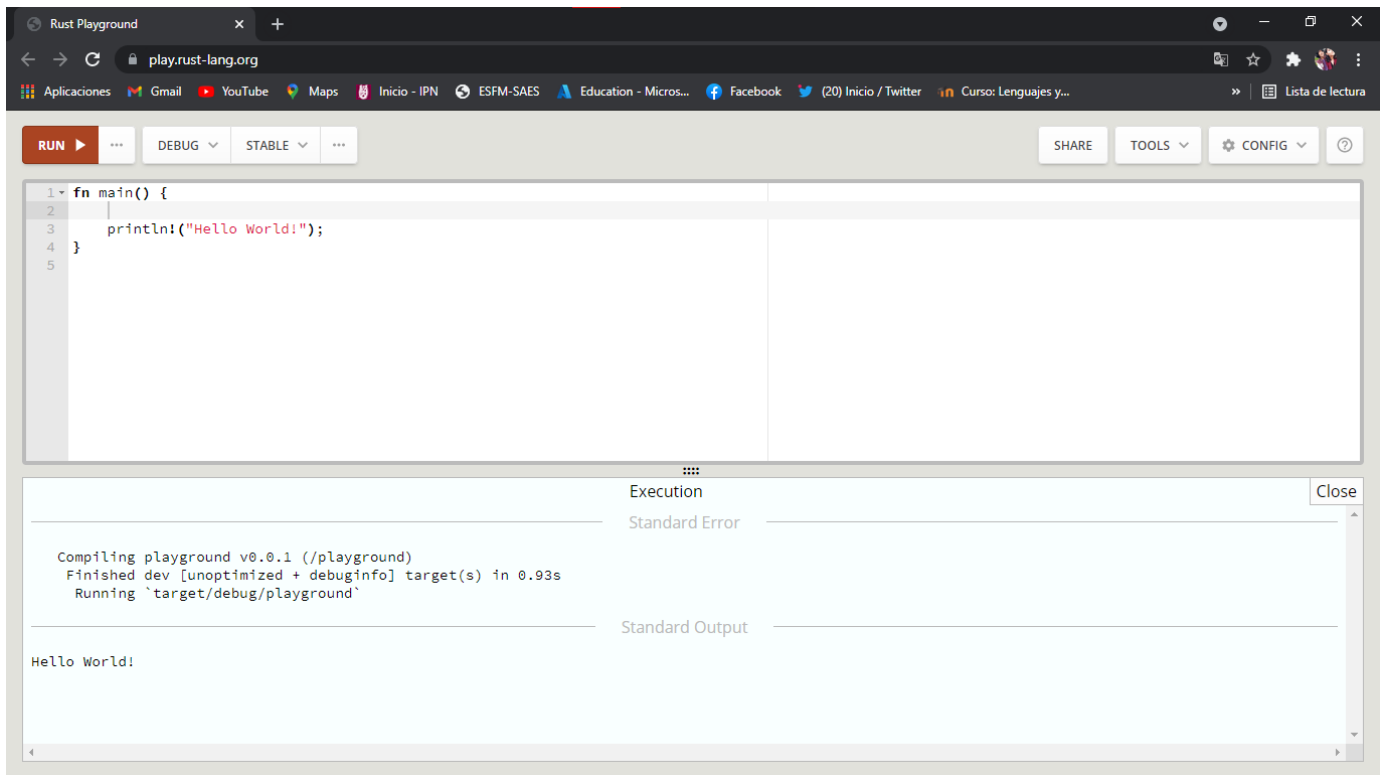
1 <?php
2
3 /** @var \Laravel\Lumen\Routing\Router $router */
4
5 /*
6  * Application Routes
7  *
8  * Here is where you can register all of the routes for an application.
9  * It is a breeze. Simply tell Lumen the URIs it should respond to
10  * and give it the Closure to call when that URI is requested.
11  */
12
13 $router->get('/', function () use ($router) {
14     return [
15         'Equipo: Los Rico Pollo <3 '
16     ];
17 });
18
19
20
21
22

```



RESULTADOS

Se hicieron tres simulaciones de las cuales 2 fueron para el primer repositorio de GitHub donde se plantea una programación más dedicada al entorno de la arquitectura de C ocupando Rúst., nos dio demasiados problemas al momento de instalar las librerías, por lo tanto, al momento de querer inicializar un código en VS, nos manda error de repositorios y aun así cuando se instalaron varias extensiones, nos marcaba el mismo error. De tal manera se hizo un hola mundo en Rúst. para entender un poco más sobre este tipo de programación.



The screenshot shows the Rust Playground web interface in a browser. The code editor contains the following Rust code:

```
1 fn main() {  
2  
3     println!("Hello World!");  
4 }  
5
```

Below the code editor, the execution results are displayed in a panel titled "Execution". The panel is divided into two sections: "Standard Error" and "Standard Output".

Standard Error:

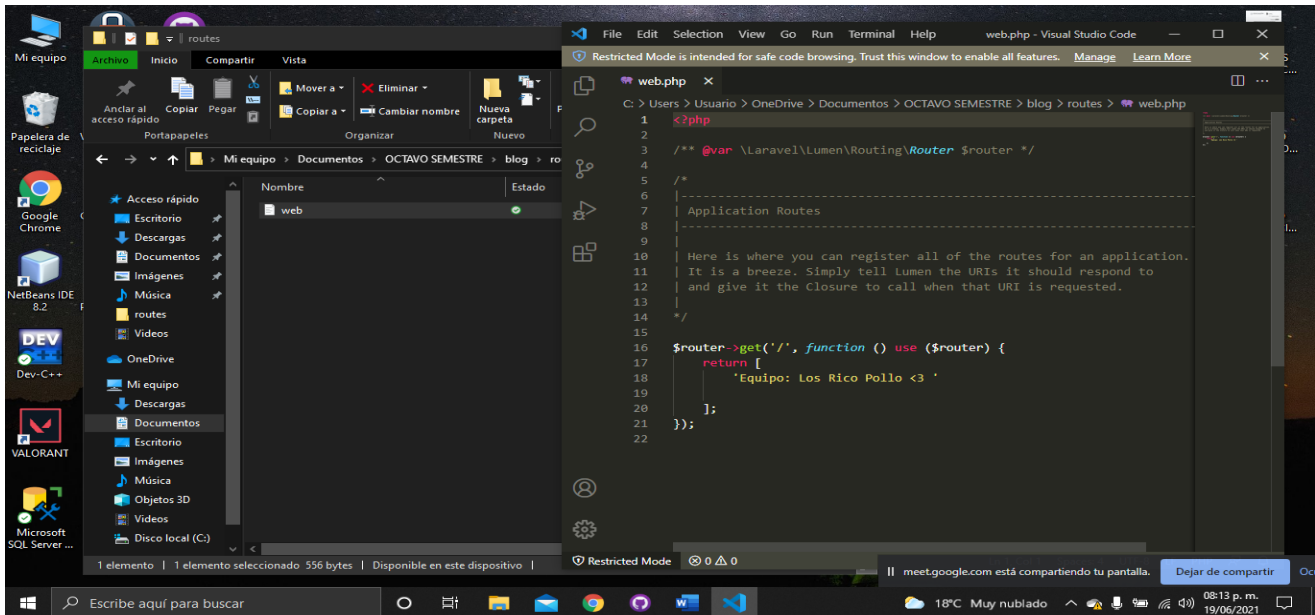
```
Compiling playground v0.0.1 (/playground)  
Finished dev [unoptimized + debuginfo] target(s) in 0.93s  
Running `target/debug/playground`
```

Standard Output:

```
Hello World!
```

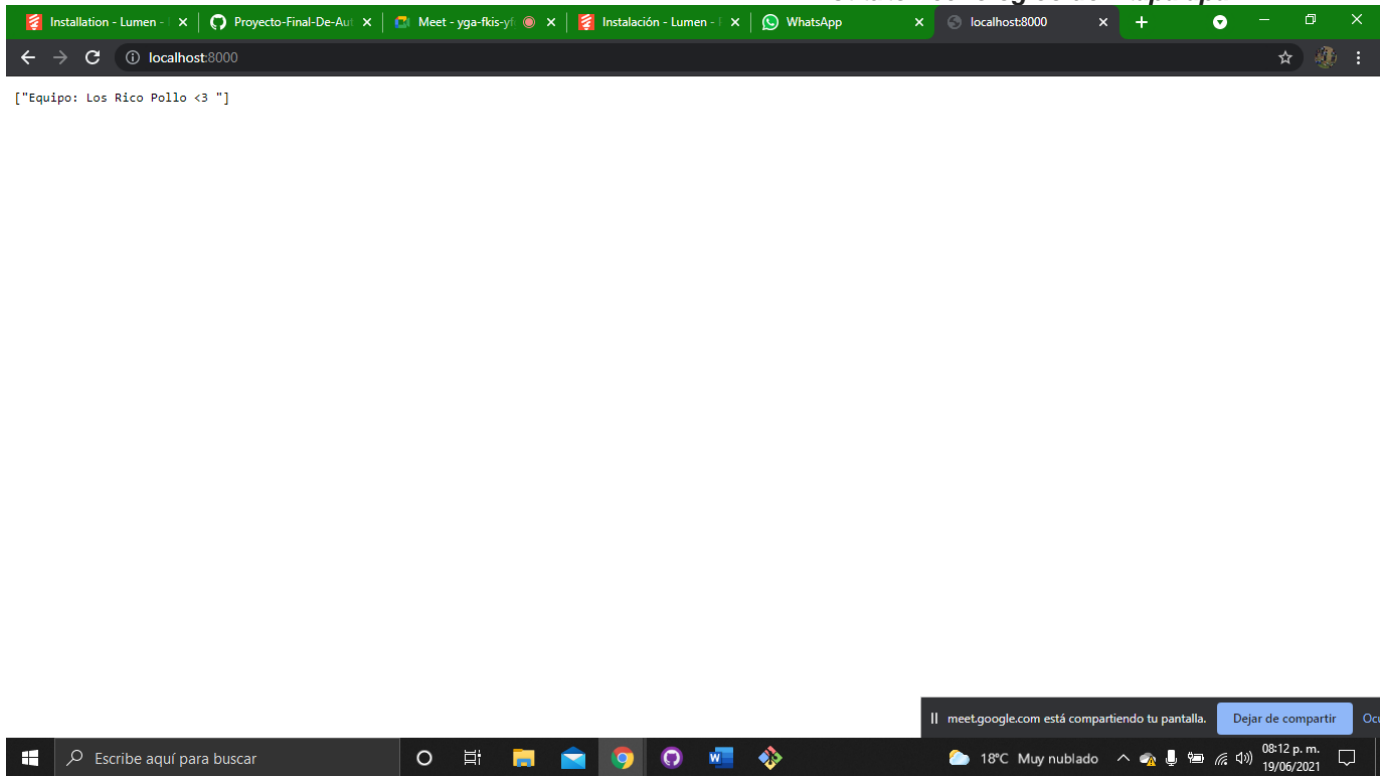
El cual tiene un main para poder ser inicializado, y un println que se ejecutara como una macro que imprime texto en la consola.

Como tal para la simulación de Lumen en conjunto con php, se debe tener principalmente un conocimiento básico de la estructura de codificación, ya que de tal manera podremos trabajar mejor con las diferentes secciones de código y, por lo tanto, dar un resultado gráfico.



Lo que podemos observar de manera paulatina, son líneas de código dedicadas a:

- La página que se utilizará.
- La función para que pueda darnos el host.
- Retorno para que se haga un bucle.
- El texto que aparecerá en nuestra pantalla, (página web).



En esta imagen se puede ver de manera gráfica el resultado de la programación anterior mencionada, ya que el mensaje o texto aparece en nuestra pantalla, sitio web.

CONCLUSIONES

- ✚ Es de gran importancia saber todas las bases fundamentales de LUMEN para la elaboración de un compilador y conocer sus aplicaciones a grandes rasgos de este mismo.
- ✚ Es necesario indagar a fondo para poder entender ampliamente a que se dedica o realiza LUMEN, puesto que hoy en día junto con Laravel funcionan ampliamente para la realización rápida y eficaz de aplicaciones web.
- ✚ Es fundamental saber las cosas que se pueden crear con diferentes aplicaciones como por ejemplo nuestro tema que es LUMEN ya que este como tal no es usado para crear microprocesadores complejos ya que a este se le denomina como el hermano menor de Laravel el motivo es porque laravel es más completo, pero lo que tiene lumen y por lo cual no se queda atrás es porque este también es un tanto completo lo único que este desecha cosas para que sea fácil de manipular y más fácil pero se puede llegar a convertir en laravel al momento de meter los complementos que lumen desecha.
- ✚ Si se realiza la investigación de manera secuencial, se pueden saber demasiado temas de interés, ya que el tema que nos toco es demasiado útil para la mayoría de desarrolladores que se dedican a resolver problemas complejos para hacer determinadas cosas e incluso para su resolución al 100%.
- ✚ Las diferentes opciones para realizar ejercicios que nos ofrece Lumen, son demasiado interesantes porque a pesar de ser algo “tediosa” la instalación y tener que prestar demasiada atención a los comandos que utilizaremos en el cmd o en alguna consola diferente a la que viene por defecto en win10, incluso si ocupamos un sistema operativo diferente ya sea una partición de Linux (Ubuntu), hay que ser cuidadosos al escribir porque como ya sabemos una palabra mal escrita se puede traducir en que no se instale o que se está instalando algo diferente a lo que querías tener.
- ✚ Incluso con todos los errores que se llegaron a tener se aprendió bastante de un nuevo entorno de programación el cual está basado íntegramente en C y C++, por ende, no es difícil aprender.
- ✚ En cuanto a los conocimientos que ya están cimentados de la investigación ardua, los tres nos dimos a la tarea de aprender cosas por nuestra cuenta, por lo que si uno tenía la duda de algún concepto otro podía explicarlo y así seguir avanzando en el trabajo final.

FUENTES DE INFORMACIÓN

Noguera Bulmaro. (octubre 24, 2012). ¿Qué es un compilador? junio 19, 2021, de culturacion.com Sitio web: <https://culturacion.com/que-es-un-compilador/>

Sin autor. (Sin fecha). La infraestructura del compilador LLVM. junio 19, 2021, de LLVM Sitio web: <https://llvm.org/>

Sin autor. (junio 11, 2021). LLVM. junio 19, 2021, de Wikipedia Sitio web: <https://es.wikipedia.org/wiki/LLVM>

Hugo. (Sin fecha). Descripción general de la representación intermedia de varios niveles. junio 19, 2021, de MLIR Sitio web: <https://mlir.llvm.org/>

Gonzales Enrique. (Sin fecha). ¿Qué es PHP? y ¿Para qué sirve? Un potente lenguaje de programación para crear páginas web. (CU00803B). junio 19, 2021, de aprenderaprogramar.com Sitio web: https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:ique-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70&Itemid=193

Sin autor. (Sin fecha). PHP. junio 19, 2021, de desarrolloweb.com Sitio web: <https://desarrolloweb.com/home/php>

Sin autor. (Sin fecha). Laravel. junio 19, 2021, de desarrolloweb.com Sitio web: <https://desarrolloweb.com/home/laravel>

McDade Jack. (Sin fecha). Instalación. junio 19, 2021, de lumen Sitio web: <https://lumen.laravel.com/docs/5.8>