

DOCUMENTO PARTICULAR DE INGENIERIA EN COMPUTACION ASFALTO DE CAFÉ

UNIVERSIDAD DE GUADALAJARA
 CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS
 DIVISIÓN INGENIERÍAS
 MAESTRA
 NORMA ELVA ESPINO ROJAS

Integrantes:

Luis Francisco Zárate Espinosa	216553182	luis.zarate5318@alumnos.udg.mx
Arellano Granados Angel Mariano	218123444	angel.arellano1234@alumnos.udg.mx
Galindo Diaz Jennifer Midori	217728482	jennifer.galindo@alumnos.udg.mx
Barajas Zavala Ulises	219129713	ulises.barajas1297@alumnos.udg.mx
Nudelstejer Gómez Ivan	218130122	ivan.nudelstejer1301@alumnos.udg.mx
Cruz Quintero Nancy Paola	217222228	nancy.cquintero@alumnos.udg.mx
Torres Solis Jaziel Edrei	221349798	jaziel.torres4979@alumnos.udg.mx
Dominguez Amezcu Marco Aurelio	216818534	aurelio.dominguez@alumnos.udg.mx
Avalos Viramontes Diego Cesar	218122014	diego.avalos1220@alumnos.udg.mx
Hernandez Hernandez Carlos Manuel	217239384	carlos.hernandez3938@alumnos.udg.mx
Gaspar Miramontes Andrea	221350664	andrea.gaspar5066@alumnos.udg.mx
Ramirez Martin Angel Omar	221350125	angel.ramirez5012@alumnos.udg.mx
Villalobos Lázaro Juan Manuel	216109185	juan.villalobos0918@alumnos.udg.mx
Rosales Gómez Moisés	217728067	Moises.romez@alumnos.udg.mx
Mayagoitia Franco Juan Pablo	217539159	juan.mayagoitia5391@alumnos.udg.mx
Trejo Reynoso Irene Marijose	217031732	irene.trejo3173@alumnos.udg.mx
Mercado Hernandez Angel Gabriel	221350141	angel.mercado5014@alumnos.udg.mx
Cruz Mortera Oliver Santiago	219108074	oliver.cruz1080@alumnos.udg.mx
Monrreal López Martin De Jesus	218145065	martin.monrreal1450@alumnos.udg.mx
Roque Gonzalez Juan Alonso	216671932	juan.roque6719@alumnos.udg.mx
Flores Gomez Carlos Ernesto	217085808	carlos.flores8580@alumnos.udg.mx
Pérez Peres Moisés Yaroslav Fedkevich	214662782	moises.perez6627@alumnos.udg.mx

Resumen

En este documento se hablará de la contribución de la carrera de ingeniería en Computación, es decir cómo se creó la página web funcional como base de datos, Esta se divide en dos partes, FrontEnd y BackEnd, en la cual se explicara a fondo cada una de las herramientas usadas para cada parte.

En el FrontEnd se utilizó html(Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) y CSS(Hojas de Estilo en Cascada del inglés Cascading Style Sheets), con estas dos se desarrolló la parte visual de la página web, a través de títulos, hipervínculos, etc. En el BackEnd se utilizó MySql(My significa la abreviatura de la hija

del cofundador y SQL Lenguaje de consulta estructurado del inglés Structured Query Language), este es el lenguaje de programación, PHP(Preprocesador de hipertexto del inglés Hypertext Preprocessor), con esta se conecta con el servidor, WAMP(Una abreviatura de Windows, Apache, MySQL y PHP), esta se encarga de la conexión a la base de datos, entre otros más.

Al final el resultado esperado es poder registrar por medio de tablas los datos obtenidos de los experimentos hechas por las otras carreras, y con base a estos resultados obtener el objetivo principal del proyecto que es observar si la mezcla de la propiedad obtenida del residuo de café (lignina) con el asfalto se obtiene más duración o tiempo de vida que el asfalto en sí.

I. INTRODUCCIÓN

Este documento tiene como objetivo el de presentar las implicaciones de los miembros de INCO en el realizamiento del proyecto de Asfalto de Café, pero de un forma técnica, nombrando el funcionamiento específico de los recursos utilizados a lo largo de la realización del proyecto, desde la formulación de una idea de aporte al proyecto hasta una larga y tendida explicación del funcionamiento de cada elemento dentro del aporte, así como fue evolucionando y progresando, así pues se mencionan los cambios y de que se trataron los cambios hasta la culminación de este, así como los conflictos con los que se encontraron los integrantes durante esta larga jornada.

Claro que lo primero que se necesita saber es el cómo y de donde salieron los recursos que utilizamos en este proyecto, y para entrar a ello se enlistan y explican su origen y sentido de existencia en la comunidad de desarrolladores y programadores y cómo entra dentro de nuestras necesidades para el realizamiento.

Después procederemos a explicar qué es lo que se implementó en la aportación, su funcionamiento en sí mismo y su funcionamiento en conjunto con todo lo demás.

II. Fundamentos Técnicos

A. Bases de datos

Las bases de datos son términos informáticos muy conocidos, ya que son ampliamente utilizados en el internet, desde que el mundo empezó a globalizarse gracias a el avance del internet, fue entonces cuando la información de los que la transitan se volvió tan importante y las formas de organizar toda esta información entrante avanzó junto con el internet, para guardar todo tipo de información dentro de servidores, de modo que se pueda ingresar a ella fácilmente.

Pero primero que nada la definición de bases de datos es

“Una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información.”

Las bases de datos tienen niveles de complejidad directamente relacionados con los volúmenes de datos que estén programados para recibir, ya que las bases de datos es solo un concepto que ya ha existido mucho antes de la tecnología, siempre ha existido la necesidad de categorizar grandes cantidades de información, de organizarla bajo diferentes parámetros, de acceder a ella para hacer con ella lo que en ese momento se necesite, y las ha habido de muchos tipos, los registros de libros en una biblioteca, registros de personas en una localidad, registros de ganancias en un negocio, y siempre ha sido un reto organizacional el recopilar de forma eficiente una cantidad de datos como lo podrían ser los libros de una biblioteca y donde se encuentran, ahora imagine el mantener un registro actualizado de los millones de usuarios que hay en internet sin la ayuda de los gestores de bases de datos, todo esto para marcar la diferencia entre las bases de datos y aquellos softwares que organizan bases de datos, ya que la utilidad de estos gestores no solo es la de recopilarlos, sino también el de consultar esos datos, el de poder modificar esos datos, ya que por lo general los datos pueden llegar a cambiar y habrá la necesidad de actualizar estos registros, pero lo más importante de esto que estos gestores permiten hacer todas estas operaciones con cantidades enormes, y cuando digo enormes hablo de millones de entradas de forma sistemática, y con sistemática queremos decir, que podemos tener un millón de entradas almacenadas en un servidor, y digamos que podemos verlas, pero al abrir estos registros nos aparecen un millón de datos desordenados, una real pesadilla, lo que mejor hacen estos gestores es el de organizar estos datos como van entrando, mediante tablas la mayoría de ellos, o al menos los más profesionales, tablas en las

que en una cantidad de columnas que decida el que lo programe podrá guardar tantos datos necesite en cada registro, y cada una de estas columnas recibirá un tipo de dato que se necesite, de esta forma los datos se hacen más digeribles y es más fácil localizar un sector de una población de datos que tenga algo en común, por lo tanto con los gestores es fácil filtrar datos de una gran población.

Los gestores son muchos y cada uno tiene implicaciones para necesidades específicas, tienen aptitudes mejores para cierto tipo de funciones.

INCO en este proyecto se usó MYSQL, un sistema de gestión de bases de datos más extendido en la comunidad, cualquier persona que programe terminará utilizando MYSQL, no solo porque sea de código abierto sino que es básico para el funcionamiento de aplicaciones, como mencione antes, cualquier aplicación que se programe, tarde o temprano necesitará guardar datos, por ello es tan vital el hecho de tener bases de datos, por lo que MYSQL es la opción básica a elegir, este gestor es relacional, relacional en términos de bases de datos siempre se relaciona, valga la redundancia a SQL, y se refiere a el uso de tablas con filas y columnas para relacionar los datos con el nombre de sus columnas, que es lo que me refería con sistemático, y de lo que se trata las bases de datos, aunque no siempre sea así, las bases de datos NOSQL, o no relacionales, ya que no usan el concepto de SQL, son las que guardan los datos sin relación a algo, que puede entrar en otras formas y necesidades de guardar datos en registros, que quiere decir esto, que estas formas no relacionales guardan otros propósitos a la hora de relacionar datos, por lo tanto no es que haya una forma equivocada de almacenar datos, sino que la información en internet es tanta que existen muchas formas y gestores de bases de datos para cada pequeña necesidad de almacenamiento que requieran los administradores; El hecho de la implementación de este gestor de base de datos en el proyecto fue la simplicidad de su funcionamiento al ser el más utilizado con las

funciones más generalizadas, ya que como había dicho los datos se almacenan con muchas necesidades diferentes y capacidades diferentes.

Algunos otros gestores de información son:

- PostgreSQL: Son gestores de base de datos poderosos que administran muy bien grandes cantidades de datos, suelen ser utilizadas en intranets y sistemas de gran calibre.
- Microsoft SQL Server: es una base de datos más potente que access desarrollada por Microsoft. Se utiliza para manejar grandes cantidades de datos.
- Access: Es una base de datos desarrollada por Microsoft.

Normalmente estos gestores son utilizados por compañías que están en contacto de muchos usuarios, por lo que SQL por sí solo no resistiría el volumen de entrada de datos, y estos han tomado aún más importancia desde que estos datos que reciben todas estas empresas de redes sociales puede capitalizar los datos, lo que hace importante que la tecnología de ese sector se haga más potente para controlar los volúmenes de entrada de datos.

Entre las capacidades de los gestores también se encuentran tipos de los mismos más allá de relacional y no relacional, aunque los estos conceptos son los más importantes, existen muchas formas de guardar documentos de muchos tipos en diferentes lugares, alguno de los enfoques podrían ser el hacer búsquedas rápidas de información, el de aplicar métodos numéricos o estadísticos a una población de datos, o el de aplicar algoritmos con relación a ciertos parámetros de los datos, siempre depende de qué datos se guardan y con qué propósito se guardan.

Los gestores de bases de datos no son en sí aplicaciones como lo podrías llegar a imaginar, al menos no todas, si te imaginabas que abres una aplicación como excel para gestionar una base de

datos, te equivocas pero no del todo, estos gestores en su parte básica son como las partes más lógicas de las computadoras, son lenguajes, es una forma más de hablarle a una computadora para que haga algo útil, y tienen sus propios comandos o palabras clave para realizar ciertas modificaciones, creaciones o eliminaciones de tablas o registros dentro de por ejemplo una base de datos manejada por sql, como lo puede ser el SELECT FROM para seleccionar una tabla o una columna de una tabla, o los muchos tipos de datos que se pueden configurar dentro de una tabla, para ello existe un campo especial del que se hace vital en las bases de datos, es el campo de ID, un dato primario identificativo, un dato relacionado con un registro que al ser primario se refiere a que ningún otro registro de la base de datos tiene un ID igual por lo tanto puede utilizarse para redireccionar ese registro completo a algún lugar, un dato que no suele ingresar por “consola” sino que se genera automáticamente y ayuda a resolver problemas lógicos dentro de los fetchs o querys en las bases de datos, que también son palabras comando dentro de SQL, que basicamente se refiere a las búsquedas de datos según parámetros que como lo pueden ser datos específicos en columnas específicas, como un (poner pseudocódigo en sql), esto tomará más peso en las implementaciones en páginas web,y se explicará más adelante.

EN CONCLUSIÓN: podemos estar seguros que las bases de datos están detrás de cada espacio en internet, son estructuras vitales para el funcionamiento de espacios únicos dentro de la red y de aplicaciones que hacen nuestra vida más fácil y que no imaginamos vivir sin sus servicios, la información es más importante hoy que nunca, ya que es un elemento vital de la globalización y del comercio en el mundo, por lo tanto queda recalculada el almacenamiento de datos y su gestionamiento eficiente para garantizar el mejor aprovechamiento de los mismos, por lo tanto es una tecnología que

debe seguir mejorando ya que sus aplicaciones son remunerantes.

III. Servidores

Los servidores son las bases que tienen a flote la vasta comunidad de internet, ya que si alguna vez te has preguntado de dónde salen estos espacios donde se almacena todos estos recursos que podemos ver en las millones de páginas web donde se comunican millones de computadores, un entorno donde van y vienen datos, de plataforma en plataforma, pues estos son la base de todo.

Un servidor o servidor, en el lenguaje informático, es un ordenador y sus programas, que están al servicio de otros ordenadores. Estos son los que “sirven” a otros ordenadores, por lo mismo se llaman servidores, porque sirven cosas y están al servicio de otros ordenadores, como por ejemplo cuando quieres ver un video de Youtube, tu computadora lo pide al sitio y el sitio está en un servidor, cuando vas a ver una película en Netflix, las películas salen de un servidor, si recibes un correo electrónico, lo recibes de un servidor de correos electrónicos, y un poco de esto se hace en este proyecto, para probar las primeras versiones de nuestra página web no se hizo directamente en el internet, sino dentro de un servidor local, ya que nuestra computadora es en sí un servidor, que puede correr cosas de forma local, que solamente nosotros podemos ver, de esa forma podemos monitorear los cambios que hacemos de forma local sin que los cambios erróneos se vean más allá, por lo que este servidor local que hay en todas las computadoras nos permite hacer muchas cosas, el problema con que nuestras computadoras personales sean servidores es que no están diseñadas para ello, sino que los servidores son ordenadores especialmente diseñados y programados para servir como tal, una cosa importante de los servidores que siempre deben estar encendidos, ya que si llegaran a apagarse entonces dejarían de ofrecer sus servicios, de proporcionar los datos que almacenan o servir

como plataformas de red, mientras que nosotros no podemos tener nuestras computadoras encendidas todo el tiempo para que funcionen como servidores, así que ahí entra la gran diferencia de este tipo de ordenadores.

Y como estos ordenadores se encargan de esta gran carga los servidores pueden llegar a ser estas salas llenas de ordenadores que funcionan como ordenadores, y muchos de ellos están configurados para funcionar mejor para ciertas funciones estos no simplemente muestran datos, sino que deben estar preparados para mostrar datos, y co preparados me refiero a que deben ser configurados para tal o cual tarea, un ejemplo de esto es el servidor que aloja nuestra página web, que está preparado para interpretar código php, por lo tanto puede entender nuestro código php y llevar a cabo sus instrucciones una vez que se está en él, cosa que no todos los servidores tienen., a esta parte de los servidores se les llama el software, este software debe ser acorde al servicio que están supuestos a dar los servidores, y el hardware es solo el ordenador, que entre mejores componentes tenga mas rápido sera el tiempo de respuesta y la posibilidad de que se sobrecarguen de trabajo/solicitudes de los clientes/computadoras.

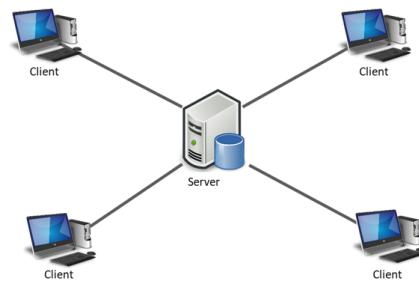


El modelo o arquitectura que siguen los servidores es el de cliente-servidor, es decir el cliente pide y el servidor proporciona los recursos o servicios, se refiere a la arquitectura de una red informática en la que muchos procesadores remotos, conocidos como “clientes”, que en cualquier caso podría ser mi computadora o la de cualquier persona, solicitan y

reciben servicio de un sistema centralizado, es decir, de una computadora host(anfitrión) a la que se le llama servidor.

Una ventaja importante del modelo cliente-servidor es que su arquitectura centralizada ayuda a facilitar la protección de los datos con controles de acceso impuestos por las políticas de seguridad. Además, no importa si los clientes y el servidor se basan en el mismo sistema operativo porque los datos se transfieren a través de protocolos cliente-servidor que son independientes de la plataforma.

Una desventaja importante del modelo cliente-servidor es que si demasiados clientes solicitan simultáneamente datos del servidor, pueden sobrecargarse, por ello es que cuando los servidores no son lo suficientemente potentes para recibir una cantidad determinada de solicitudes de servicio se sobrecargan y páginas completas que se alojan en servidores puede denegar su servicio, ya que se encuentran una congestión de red.



Como antes se mencionó los servidores pueden y están preparados para dar cierto tipo de servicios, por ello es que existen tipos de servidores que se dedican exclusivamente a tareas específicas, de modo que se ha denominado en ellos tipos de servidores, a continuación se enlistan:

A. Servidores de Mail:

Estos servidores sirven básicamente como una oficina de correos, almacenan correos, envían correos y también reciben correos, además de contar con otras aplicaciones que tengan que ver con correos electrónicos; estos tipos de servidores son de los más antiguos ya que los correos electrónicos

son uno de los avances más tempranos del inicio del internet, por lo mismo los servidores dedicados a ello son pioneros.

Son programados para manejar este tipo de información y al mismo tiempo también tiene sus subtipos que pueden entre sus funciones, retener correos hasta que el usuario los abre, hacer previsualizaciones del contenido de los correos electrónicos para que no sea necesario descargar su información a menos que el usuario así decida.

B. Servidores de Base de datos

Un servidor de base de datos, también conocido como database server o RDBMS (Relational DataBase Management Systems) en caso de bases de datos relacionales, es un tipo de software de servidor que permiten la organización de la información mediante el uso de tablas, índices y registros.

C. Servidores web

Un servidor web se ocupa de guardar la información en formato HTML de los sitios, donde se incluye texto, imágenes, videos y todo tipo de datos. Mediante un explorador web, los usuarios pueden visualizar todo esto en sus pantallas.

Y fue el que utilizamos para subir nuestra página web a internet, de forma que cualquier persona interesada en ella, y los integrantes de nuestro equipo sean capaces de ingresar en él y visualizar lo que hicimos, tanto el html dentro de nuestro proyecto, el CSS que se refleja en el HTML y el PHP que hace que las funcionalidades de la página web que creamos cobren vida y el usuario, ya sea un miembro de nuestro equipo o un extraño que se haya topado con nuestra página y quiera saber mas de nuestro proyecto pueda interactuar con las opciones o que el integrante de nuestro equipo quiera ingresar datos en nuestra base de datos.

D. Servidores Cloud

Este tipo de servidores no es que tengan una función de muestra especial, simplemente guardan grandes cantidades de información de cualquier tipo, y seguramente ya has oído hablar de estos servicios de CLOUD o la nube antes, donde por lo general últimamente se hacen respaldos de información de ordenadores o teléfonos, y es que la razón de este tipo de servidores sean un tipo es que son contratados por usuarios y por empresas para sus usuarios para guardar su información de forma segura, ya que se les considera como cajas fuertes de información en el internet.

IV. Funcionamiento y naturaleza de los datos recolectados

A. Distintos tipos de usuarios:

Dentro de los usuarios que se concibió un poco desde del principio hay 2, los cuales tienen diferentes implicaciones o propósitos dentro de la página web, la página web sirve como punto de encuentro de los integrantes de nuestro equipo, ya sean parte del equipo de desarrollo de INCO o carreras más enfocadas a la experimentación, pero claro que esta no fue su objetivo final, ya que la página principal tiene como objetivo dar una breve vista de nuestro proyecto, con palabras clave lo que hicimos, así que toma un enfoque más público que el que tenía al principio que servía como una pagina de encuentro y recibo de datos a una pagina que si hace ello y además nos da a conocer.

Entonces tenemos 2 tipos de usuarios, los que solo verán nuestro proyecto desde una perspectiva general y los que verán, editarán y borran datos de la base de datos, esto hace que la página web sea muy versátil haciendo dos cosas a la vez.

Aun así hay otro usuario secreto en todo esto, y es el que tiene acceso a la base de datos por medio de phpMyadmin, que tiene un acceso más crudo a los datos y a las tablas de las bases de datos, veámoslo como grados de prioridad o de permisos, el público en general tiene prioridad uno, y puede ver lo que

está presentado en la pagina, mas sin embargo no tiene forma alguno de ver datos recabados ni interactuar con ellos de ninguna forma, por otro lado tenemos a los que tiene acceso a un usuario y una clave de acceso, tendrán acceso al apartado de experimentación, donde podran interactuar con las tablas y ver los datos ya recabados, teniendo un grado de prioridad 2, y por ultimo los administradores tendran el maximo grado de prioridad, ya que ellos son los que determinan estos grados anteriores, y los que dan acceso a ciertas partes de la base de datos, ya que los de acceso 2 no ven ni tienen acceso a todas las tablas de la base de datos como lo puede ser la tabla de usuarios, esto porque no hay forma de registrar usuarios desde la pagina web, solo mediante la inyección directa por los administradores a la base de datos, esto quiere decir que no hay una implementación de una pagina externa de registro de nuevos usuarios, no es una opción disponible porque rompería con el orden de jerarquía que lleva la pagina web.

así que en conclusión:

1. Público general
2. Usuarios
3. Administradores

B. Por qué la base de datos es accesible desde un sitio web

Esto tiene 2 cuestiones a responder, el porqué y el cómo, el porqué es muy fácil, las bases de datos por sí mismas son espacios muy técnicos, que funcionan con lenguajes gestores de bases de datos, con muchos términos en su sintaxis de conocimiento específico de quienes trabajan con ellas, en pocas palabras las bases de datos no las puede manejar o navegar entre ellas una persona sin conocimiento de ello, por ello la importancia de conectarlas con ambientes depurados de estos términos y simplificados para cualquier cosa que se desee hacer con ellas, si es que solo se quiere visualizar el contenido de sus tablas o si también se quiere actualizar algún campo de sus tablas, o en últimos

términos se desea eliminar algún campo o hasta el registro completo, son acciones que se traducen en lenguaje SQL en este caso y llevan una serie específica de comandos que de nos er escritos correctamente la base de datos no entenderá nuestra solicitud y no hará los cambios pertinentes, por ello la incorporación de ellas en las páginas web.

Otro porque sería la necesidad de recabar información de formularios, cosa que la base de datos no puede hacer dentro de su lenguaje, y que las páginas web están especializadas en hacer, ademas de que resulta lógico tener un almacén de los datos que entran ya que serán de suma importancia para la toma de decisiones posteriores, en muchos sitios en internet los datos recabados de millones de personas ya alimentan potentes algoritmos con objetivos corporativos.

El cómo es otra cuestión no muy diferente, lo primero que necesita una página web para hacer uso de este espacio de almacenamiento y a sus múltiples tablas es conectarse por medio de métodos ya implementados en la estructura de muchos lenguajes de programación orientados a la creación y manejo de páginas web, en nuestro caso usamos métodos relacionados al lenguaje de sql dentro de php, donde se ingresaba una serie de datos necesarios para establecer una conexión con la base de datos y una vez con esta conexión en vivo con la base de datos es posible usarla para alimentar sentencias condicionales o alimentar la base de datos con formularios que se recaben con formularios de html y enviarlos a ciertos campos en ciertas tablas por medio de php, las posibilidades no tienen límites.

El limitar el acceso a ciertas personas dentro de la página web permite que haya un control de los datos ingresados, y que la base de datos no se llene de datos indeseados, como lo podría ser en nuestro caso, datos erróneos o inconclusos en tablas dedicadas a ciertos experimentos, de modo que pudiera resultar en una pérdida completa del

progreso, ya que no sabremos cuales son los datos verídicos y los datos erróneos, lo que haría un mal manejo de nuestra página web y una baja seguridad en la entrada de datos.

C. Hosting

Con relación a los servidores, entra un término muy utilizado en la web, el hosting, la traducción de host del inglés es anfitrión, el dueño del sitio en la red, y es importante porque es el sitio donde se aloja nuestro sitio web y todos los recursos que necesita para funcionar correctamente, que pueden ser muchas cosas dependiendo de las funciones que brinde el sitio, en nuestro caso, la base de datos, que es parte esencial de la página, como ya lo hemos mencionado antes.

El host(anfitrión) que dará hogar a nuestra página web y los recursos que necesita, está haciendo hosting, que en sí es el dueño de un servidor, y muchas veces su propio administrador, es el que presta o vende este espacio dentro de su servidor para que en él se hallen los sitios web, este servidor por lo general debe tener algunas características para que sea apto para el eficiente movimiento del internet, como lo son calidad de rendimiento, para no dejar tirados los recursos necesarios de funcionamiento de las páginas web alojadas en él y con él un tiempo de carga decente, capaz de evitar códigos con errores que puedan dejar desprevenida a la página a ataques informáticos.

La potencia del hosting determinara que tantos datos pueden entrar en su base de datos y que tantos usuarios serán capaces de acceder al sitio web al mismo tiempo sin sufrir problemas por sobretaballo, por ello cuando se escoge el hosting se debe revisar que tenga la máxima compatibilidad posible con los requerimientos computacionales de la página y la cantidad de usuarios que se esperan que ingresen, de otro modo los problemas de rendimiento se acumularan haciendo que la página no funcione correctamente o hasta que sea

imposible de ingresar a ella, ya que los servidores tienen problema para mostrar páginas saturadas de solicitudes.

V. Páginas web

A. ¿Qué es una página web?

Se conoce como página Web, página electrónica o página digital a un documento digital de carácter multimediático (es decir, capaz de incluir audio, video, texto y sus combinaciones), adaptado a los estándares de la World Wide Web (WWW) y a la que se puede acceder a través de un navegador Web y una conexión activa a Internet. Se trata del formato básico de contenidos en la red.

Las páginas Web se encuentran programadas en un formato HTML o XHTML, y se caracterizan por su relación entre unas y otras a través de hipervínculos: enlaces hacia contenidos diversos que permiten una lectura compleja, simultánea y diversa, muy distinta a la que podemos hallar en los libros y revistas

Las páginas Web cumplen básicamente con la tarea de brindar información de cualquier índole y en cualquier estilo o grado de formalidad.

Algunas, al mismo tiempo, permiten distintos grados de interacción entre usuarios o con alguna institución, como son las páginas de foros, servicios de citas o redes sociales, las páginas de compra y venta de bienes, las páginas de consulta o de contacto con empresas, instituciones gubernamentales o con ONGs, e incluso las páginas de soporte técnico especializado.

En principio, las funciones de una página Web son tan amplias como la demanda de los usuarios y la oferta de sus creadores .

B. Tipos de Páginas Web

Existen dos tipos de páginas web, conforme al modo en que se genera su contenido

Páginas Web Estáticas: Operan mediante la descarga de un fichero programado en código HTML, en el que están todas las instrucciones para que el navegador reconstruya la página Web, accediendo a las ubicaciones de sus elementos y siguiendo un orden preconcebido, rígido, que no permite la interacción con el usuario. Este tipo de páginas son meramente informativas, documentales, no interactiva

Páginas Web Dinámicas: A diferencia de las anteriores, las páginas Web dinámicas se generan en el momento mismo del acceso del usuario, empleando para ello algún lenguaje interpretado (como el PHP), lo cual le permite recibir solicitudes del usuario, procesarlas en bases de datos y ofrecer una respuesta acorde a sus requerimientos.

VI. Frontend

A. ¿Qué es Frontend?

Un sistema de front-end es parte de un sistema al que el usuario accede directamente para recibir o utilizar el sistema anfitrión. Permite a los usuarios que accedan y puedan solicitar en el sistema de información.



Se utiliza principalmente para enviar preguntas, solicitudes o recibir datos desde el sistema anfitrión, como típicamente los de front-end tienen más las capacidades de controlar la lógica computacional de un negocio muy limitado, base de datos y funciones del sistema.

Se muestran varios tipos de contenido y hace la entrada el usuario que muestre disponible para el backend, como la estructura básica del sitio web,

como navegador el frontend textos, gráficos, videos y otros medios.

Se desarrolla, principalmente, a través de tres lenguajes: HTML (HyperText Markup Language), CSS (Cascading Style Sheets) y JS (Javascript). Cada uno de estos lenguajes se usa para desarrollar diferentes partes del front-end. Vamos a explicar en qué consiste cada uno, y cuáles son sus funciones. Básicamente, son tres lenguajes que se dividen las tres tareas básicas del desarrollo: arquitectura, apariencia, e interacción.

Incluso con una base de datos donde la mayoría de los procesos se ejecutan en segundo plano y son invisibles para el usuario, existe una interfaz gráfica de usuario, como la tabla de salida después de una consulta de datos. Los sistemas de aplicaciones basados en la web casi siempre se dividen en un front-end y un back-end. En un sistema de gestión de contenidos, el front-end es la interfaz en la que los visitantes pueden ver el contenido publicado.

VII. HTML

A. ¿Qué es HTML?

El HTML define el contenido del lenguaje. Se trata de un conjunto de etiquetas que sirven de texto y/o otros elementos que comprueban una página web, imágenes, listas, videos, etc. lo que permite hacer una página web, se consiguen los resultados profesionales gracias a nuestras capacidades para el diseño y nuestra vena artística, así como a la incorporación de otros lenguajes para definir el formato con el que se tienen que presentar las webs.

Este lenguaje se escribe en un documento de texto, por el cual se requiere un editor de textos para escribir una página web.

Hasta el momento ya tenemos una idea sobre lo que hace HTML pero aún no sabemos cómo funcionan las etiquetas. Para esto vamos a imaginar que tienen que hacer un ensayo sobre algún tema,

consideraremos que el ensayo será nuestra página web y la manera en la que va ordenado un ensayo es por un título, la introducción, el desarrollo y la conclusión, si se tiene en cuenta que esto nos dice cómo debe ir nuestro ensayo podemos considerar que éstas serían las etiquetas porque nos dicen la estructura que lleva, es decir, la forma en que se acomoda el texto en nuestro ensayo.

Para conocer el código HTML que utiliza una página web, hay que seleccionar Ver código fuente en nuestro navegador (como Internet Explorer o Mozilla Firefox). Al elegir esta opción, se abrirá el editor de texto con el código HTML de la página que se está visualizando.

VIII. CSS

CSS es un acrónimo para decir en pocas palabras CASCADING STYLESHEETS, o en español, Hojas de estilo en cascada, este es un lenguaje de programación basado en el diseño web o diseño gráfico en la web, y tiene fuertes vínculos en html, ya que este lenguaje es capaz de definir muchas opciones gráficas profundas dentro de lo visual de una pagina web, así que su enfoque es meramente de frontweb, lo fascinante de CSS es lo profunda que puede llegar a ser la personalización de pequeños y grandes elementos dentro de una página web, con esto me refiero a que características como lo son colores, bordes, márgenes, fondos, layouts, tamaños, fuentes, etc. pueden ser completamente independientes o no, dependiendo del propósito de su diseño, en elementos como títulos, párrafos, enlaces, listas, tablas, formularios, botones; Con esto quiero inferir que puedes tener 3 elementos en un documento html con un color diferente, una fuente diferente, con bordado, y un centenar am,s de características que hace que los sitios web sean muy versátiles, le den rienda libre al diseño online y a la creatividad a los diseñadores.

Como ya se expuso, dentro del código de CSS se puede seleccionar un elemento y darle un estilo específico, pero otra pregunta válida sería el cómo funciona realmente este lenguaje.

CSS puede abrirse dentro de un documento html con las llaves del nombre “`<style></style>`”, para empezar a describir a los elementos de una página, este método se le llama método embebido, y los estilos descritos dentro de las llaves de estilo solo cambian el estilo de la página en la que se están ingresando está programando estos estilos, sin embargo existen formas más eficientes de aplicar estilos de cascada dentro de las páginas web, este método es el CSS externo donde se crea un archivo .css, que es la forma más utilizada, para que muchos documentos de html puedan hacer uso de un diseño generalizado, importando el código de este .css a tu archivo .html, de modo que se reutiliza el código de css para definir el estilo de un sitio web completo, de modo que todo se vea uniforme.

Con este lenguaje los programadores de CSS han podido realizar páginas profesionales y atractivas para los usuarios que las visitan, que es el máximo objetivo del lenguaje.



Actualmente CSS se encuentra en la versión 3, donde se actualizaron funciones ya existentes y se agregaron funciones como transparencias, sombras, degradados, bordes redondeados, transformaciones y animaciones.

IX. GIT



Lo más importante de estos dos es git, git es un software de control de versiones para desarrolladores, básicamente la idea de git es tener un diario de un código y sus actualizaciones, cuando se hicieron estas actualizaciones, en que parte del código se hicieron las actualizaciones, y quien hizo estas actualizaciones, así es, git es un software que permite a muchos programadores colaborar en un mismo proyecto, cosa que dentro del mundo es esencial, git le permite a todos trabajar dentro de un mismo proyecto sin pisarse mientras trabajan, quiero decir, sin que haya conflictos en el código que se maneja, ya que trabaja con versiones, si hay algo que no compila, que tiene demasiados errores que simplemente no gusta, siempre se puede volver a la versión anterior del código donde todo funcionaba perfectamente bien y trabajar con esa versión, ya sea usándola como lo es o modificar desde una versión funcional.

Para hacer posible todo esto git hace uso de comandos y funcionalidades con la que el creador puede controlar el flujo de trabajo y las versiones o actualizaciones que se hacen dentro del código.

Lo primero que se debe hacer cuando se está trabajando con git crear un repositorio, el repositorio de git es como tener un espacio asignado que git estará supervisando, donde desde nuestra propia computadora revisará si el archivo que tenemos en un repositorio tienen algún cambio y donde exactamente.

Sin adentrarnos demasiado en lo lejos que ha ido git, ya hay muchas funcionalidades, extensiones y aplicaciones que hacen el manejo de git aun mas

facil y profesional dentro de nuestras computadoras, como lo pueden ser las extensiones de git y github en editores de código como Visual Studio Code, o la propia aplicación de Github.

Como decía, repositorio es un proyecto que está siendo seguido por el control de versiones de git, cada cambio que se hace y se sube en él es registrado con todos los datos antes mencionado, que se cambio, cuando se cambio y quien lo cambio, y claro que estas actualizaciones siguen un hilo, de cual versión fue primero, cuál después, y en git este concepto se recreó con las ramas o “branches”, las cuales son líneas de trabajo, y claro que sería extraño darle tanto énfasis a la implementación de ramas de trabajo si solo hay una, pero como podrás imaginar el flujo de trabajo con git es mas complejo que solo una linea de versiones que se apila como lo podría llegar a ser con las aplicaciones que descargamos en línea, pero es que tambien tiene relacion con ello, el hecho de que se llaman ramas es que hay muchas salidas dentro de las ramas de un árbol, muchas bifurcaciones, y es que la rama principal que por lo general se llama “master” o “main” son las versiones del proyecto que si salen al publico, pero de esa rama principal pueden salir otras ramas, y estas ramas secundarias para que funcionan?, principalmente las ramas secundarias funcionan como espacios de trabajo aislados de las versiones funcionales principales; por ejemplo, puedes estar tratando de programar una nueva funcionalidad en un tu proyecto, pero no quieres que si llegara a salir mas esta serie de versiones aisladas de la principal, repercutieron en el funcionamiento de tu proyecto principal, por lo tanto abres una rama para ello y creas varias versiones hasta que finalmente funciona tu funcionalidad y entonces puede juntar esta rama con la principal, y dado que no funcione simplemente puedes quitarla sin problema alguno, permitiendo a programadores o equipos de programadores desarrollar partes de un proyecto sin cambiar realmente el proyecto en sí hasta que así se decida.

Cada “versión” dentro de git se sube manualmente por los programadores involucrados, no es tan simple como presionar un botón de guardar, estas versiones son llamados “commits” en git, y el primero de todos que se hace un proyecto es cuando se sube el proyecto en sí, por medio de estos commits se suben nuevas actualizaciones del código, donde no solo se refiere a agregar, sino también a cambiar código y a borrar código, o los tres a la vez, estos commits se preparan asegurándose que todos los cambios realizados entran dentro de esta nueva actualización, además que el programador le puede dar un nombre y texto a su commit, indicando que fue brevemente lo que actualizar y mejorar la comunicación general de los commits.

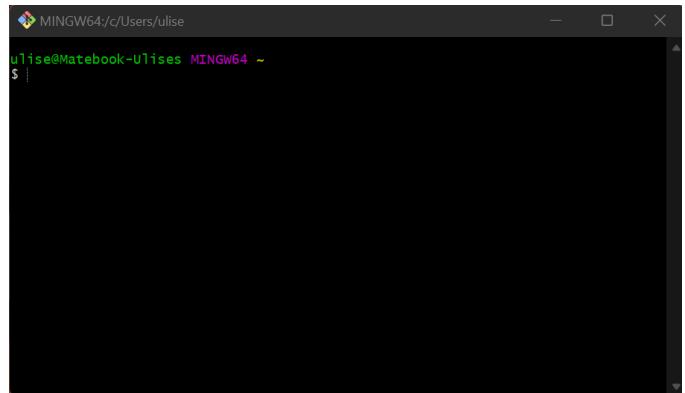
Existen otras ramas llamadas “Forks” que son proyectos diferentes que se derivan de otro proyecto, aunque se derivan de un proyecto y toman los recursos del mismo cuando uno hace un fork quiere decir que este nuevo proyecto que se deriva tiene un enfoque diferente, por lo tanto es otro proyecto. Como si fuera una herencia de clase en c++.

Cuando estas ramas se unen a la rama principal lo hacen mediante el comando “git merge”, que fusiona las ramas, sin embargo esto no siempre podría funcionar de maravilla, ya que siempre puede haber problemas de compatibilidad con las versiones y dar algunos problemas.

Una característica importante de git es la integridad de datos, con esto me refiero a que git hace un checksum(suma de comprobación) de cada archivo dentro de los commits para asegurarse que todos los colaboradores tienen los mismos datos,

Este software fue creado por Linus Torvalds como su segundo gran proyecto mientras él y colaboradores creaban el Kernel de Linux, y como había muchos colaboradores en el proyecto y el software que utilizaban se hizo de pago, Linus

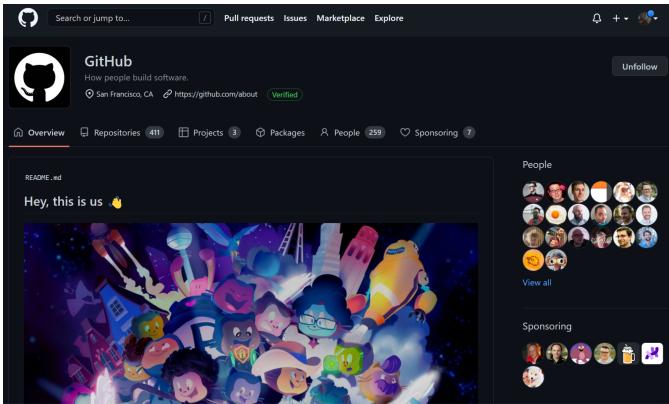
decidió crear el propio para seguir trabajando en otro gran proyecto, así que git, el software controlador de versiones más usado en la comunidad es consecuencia de otro proyecto.



X. GITHUB



Github es un sitio en el internet muy especial para los desarrolladores y programadores que desean ser parte de algo más grande, es básicamente una red social para programadores, en la que los desarrolladores pueden subir sus proyectos o repositorios y hacerlos públicos para otros desarrolladores puedan también ser parte del proyecto aportando su pequeño grano de arena a él, dentro de github se han desarrollado grandes proyectos como Wordpress en el que todavía es posible aportar por medio de git, haciendo una rama y luego de ser posible fusionarla a una nueva versión del mismo, todos estos conceptos vistos en el apartado de git.



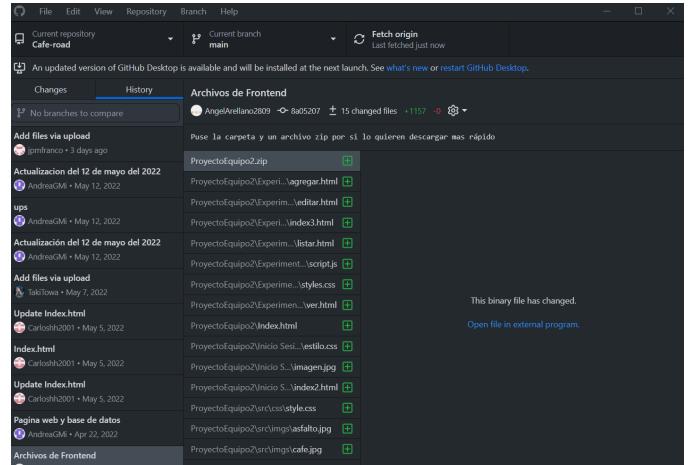
Esta fue comprada por Microsoft en 2018, y aunque al principio servía como forma de comunicación entre colaboradores, ha evolucionado para ser una red social que te impulsa a descubrir proyectos de desarrolladores alrededor del mundo, de conocerlos a ellos y a su ideas y de trabajar en sus ideas junto a ellos, una red social muy peculiar que sigue con los valores que un dia revolucionaron la tecnología, los valores de colaboración y unidad, ya que muchos proyectos conocidos han sido fruto del trabajo de miles de personas trabajando en conjunto sin un fin de lucro, sin un salario, usando sus tiempos libres para ayudar al progreso colectivo.

Además de esto cuenta con una aplicación para escritorio que sirve como facilitador de git, como un IDE que facilita los comandos de git en fáciles botones, además que dentro del mismo se puede escribir código, confirmar cambios y hacer commits en tiempo récord, y fue la que usaron muchos miembros de INCO para comunicarse en este proyecto.



La aplicación de escritorio puede clonar el repositorio o encontrarlo entre tus archivos personales para traerlo al escritorio de GitHub Desktop y empezar a hacer cambios, además se

pueden gestionar las ramas, seleccionarlas para trabajar en ellas y fusionarlas de ser necesario, lo que puede ser genial para los principiantes en github que no están tan familiarizados con la consola de git/gitbash.



XI. Backend

Es la lógica detrás de todo lo que hace que la página web funcione correctamente. El backend de una solución determina qué tan bien funciona la aplicación y qué experiencia, buena o mala, tendrá el usuario al usarla.

El trabajo en esta sección asume algo completamente diferente a la interfaz de usuario, ya que requiere el dominio de otros términos de programación, y el lenguaje requiere lógica, en el que el campo también resuelve el problema óptimo: codificación de recursos, seguridad del sitio y otros factores.

Dentro del desarrollo de aplicaciones hay dos partes esenciales, el que ve la persona que los usa y la parte que hace funcionar a la aplicación, si alguna vez te preguntaste cómo es que esa aplicación hace lo que hace, siempre hay un miembro de backend en ello, como bien se mencionó el backend se encarga de la parte lógica de la página o de la aplicación, de básicamente el funcionamiento íntegro de las funciones que ofrece, del backend depende que una página no se caiga, falle, o haga lo que está

diseñada para hacer y es donde se aplica las estructuras de datos y códigos que no están directamente relacionados con el como se ve sino con el como funciona, entre esos lenguajes se encuentra php, que fue utilizado en este proyecto, java, python, javascript, cuál lenguaje que maneje información con un propósito funcional y racional, en pocas palabras el esqueleto que hace que la tecnología funcione.

En esta ocasión el equipo de backend designado en este proyecto se encargó de las bases de datos conectadas a las páginas web diseñadas por front.

Es importante decir que los dos(backend y frontend) son partes importantes para el diseño funcional y eficiente de páginas web y aplicaciones y que se creó esta división para hacer más específico las áreas de trabajo para diseñadores e ingenieros.

XII. PHP

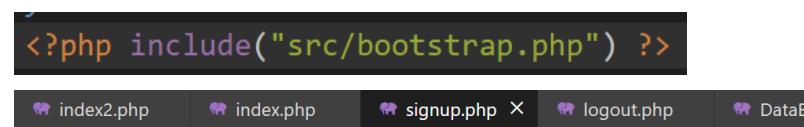
PHP es un lenguaje de programación para desarrollar aplicaciones y crear sitios web que cada día atrae más seguidores. Fácil de usar y en constante evolución, es una apuesta segura para cualquiera que quiera trabajar en proyectos calificados sin problemas.

PHP generalmente se define como un lenguaje del lado del servidor.

Esto significa que es aplicable en la programación que ocurre en el servidor web responsable de ejecutar la aplicación o, más a menudo, en una página web. Este trabajo preliminar le permite cargar elementos de la página antes de mostrarla, por ejemplo, a los usuarios que visitan una página web.

En el servidor se ejecuta código PHP que, mediante la lectura de los comandos, puede activar todos los elementos funcionales y la apariencia visual de la página web.

PHP fue de gran ayuda para la programación lógica de la página web, ya que con abrir unas simples llaves “<?p ?>” dentro de un archivo “html” y lo digo entre comillas ya que el código html y php va junto, o funciona junto sin problemas, pero si hay código php en un archivo .html deja de ser un .html para ser un archivo .php.



Se decidió usar php, aunque no en un principio, ya que permite juntar todo en un uno sin necesidad de aprender demasiadas cosas sobre el código, es decir, php es un lenguaje tan extendido aunque sea tan viejo, por así decirlo, ya que ha tenido múltiples actualizaciones, ya que se pueden hacer muchas cosas de forma muy simple, y que se combina el front con el back en los archivos php.

Como se mencionaba anteriormente php es un lenguaje de parte del servidor, que se codea y trabaja conjuntamente con la base de datos, y fue nuestro puente con las base de datos, php contiene una parte muy lógica dentro del funcionamiento de las opciones de la página, es decir, su utilidad le da a la página una versatilidad de funcionamiento condicional en la entrada y salida de datos, un ejemplo de esto es la implementación de login y sign up, la capacidad de darle a una pagina web la posibilidad de jalar datos de usuarios en la página para compararlos con los de la entrada de un login, hablo de iniciar sesión, una funcionalidad básica dentro de las páginas web, que te permite identificarte en la página y atribuirse los movimientos y darte acceso a datos ingresados con anterioridad entre muchas otros privilegios de estar registrado en una página web, y es eso lo que se hace en un login, que también es un ejemplo de implementación de la parte lógica con php, de no tener ninguna coincidencia pues la página por medio de mensajes programas en las condicionales que entren en dado caso, la página te avisara de el

tipo de salida que tuvieron tus datos de entrada y lo que puedes hacer con ello, muy seguramente una de las salidas es que no estés registrado y entonces se hace otro importante uso del php, el ingreso de registros desde una página web, a este punto es importante mencionar que php trabaja con comandos de SQL ajustados a php, osea que hay un interesante intersección de lenguajes y tecnologías funcionando en conjunto cuando se ingresan datos para registrarse, cosa que php hace muy bien, con esta intersección de lenguajes me refiero a esto:

- Primero que nada y sin meterme demasiado en html, se abren unas llaves de formulario que habilitan un espacio donde se pueden escribir datos.
- Este formulario manda los datos a una función de php al registrar que se presiona un botón con un atributo llamado “submit” que está dentro del formulario de html
- la función de php atrapa esos datos que manda el formulario de html y estos datos tienen nombre, son como variables, los nombre se establecen en el formulario de html, y estas variables
- Mediante una conexión con php a la base de datos que se guarda en una variable de php llamada conn, de connection y se usan un comando “sql_connect” que es una función reservada de php que tiene directa relación con lenguaje de SQL y que recibe argumentos como el nombre de la base de datos, el servidor donde se encuentra(host), el usuario que lo maneja y una contraseña, a cambio regresa la conexión con esta y la disponibilidad de usar métodos de SQL en el que podríamos llamar como objeto de tipo SQL; todo esto está en otro archivo de php que se importa en todos los archivos que hacen uso de la base de datos, de modo que no es necesario volver a escribir este código cada vez que se necesite sino que se manda llamar cada vez que se necesite.

- Al final y con el “objeto” con php puede meter las variables obtenidas en la tabla y columnas a las que correspondan las mismas y crear un nuevo registro en la base de datos

Todo esto para explicar como php funciona en conjunto como un esqueleto que mueve los datos de la página a la base de datos o viceversa, así como usar estos datos de entrada y salida para diferentes propósitos según convenga al programador.

XIII. WAMP

WAMP se ejecuta como un proceso en Apache y se comunica con la base de datos MySQL, lo que ayuda a crear un sitio web dinámico. Además, WAMP es un conjunto de servicios que permite mostrar tu sitio web localmente (sin estar conectado a Internet). Como resultado, facilita el trabajo de un buen número de desarrolladores. Puedes probar nuevas funciones y plugins como mejor te parezca, sin riesgo. Vale la pena recordar que, una vez creado el sitio web, es necesario elegir un servicio de alojamiento web para ponerlo online (para el alojamiento hablaremos en un momento). La pila de software WAMP se puede descargar desde la página de descargas del proyecto WampServer. Para los usuarios de Microsoft Windows este es un paquete de fácil instalación, con un panel de control que inicia Apache Web Services, PHP y MySQL en la computadora local.

Por lo tanto, WAMP y otros tipos de software con nombres y funciones similares son una buena manera de desarrollar sitios web en una computadora local sin transferir archivos a un sitio web activo. Además, los principiantes a menudo lo usan con fines de aprendizaje y prueba.

A. Apache

Se utiliza un servidor web para mostrar un sitio web. Tan pronto como solicites que se muestre una página, el servidor extraerá el contenido y se lo

mostrará. Uno de los mayores desafíos es poder atender a varios usuarios simultáneamente, ya que cada uno solicitará la visualización de una página diferente. Un servidor web también es capaz de leer y procesar diferentes lenguajes de programación como PHP, Python, HTML, JavaScript y otros.

El servidor web Apache tiene módulos que agregan funciones al software, como MPM (para manejar varios modos de procesamiento) o “mod_ssl” (para activar el soporte para SSL V3 y TLS).

Hay otros módulos Apache, como módulos de seguridad, caché o reescritura de URL. Además, puedes crear y configurar el servidor utilizando un archivo .htaccess, que se tiene en cuenta en todos los alojamientos compartidos.

B. Instalación de WAMP

Como anteriormente se mencionaba, WAMP es un software que nos permite trabajar proyectos desde nuestro propio ordenador sin la necesidad de tener conexión a internet, pero como todo software, primero debemos tenerlo instalado correctamente, a continuación, explicaremos como se instala nuestra herramienta principal de trabajo:

1. Primero descarga WAMP del sitio oficial. No es necesario que completes la información de registro, directamente debemos dar clic en “Ir a descarga directa”.
2. Cuando ya se haya completado la descarga del archivo “wampserver.exe” lo ejecutamos y damos permiso como administrador.
3. A continuación, nos irán apareciendo varias configuraciones predeterminadas y solamente debemos dar clic al botón “Siguiente” hasta llegar al botón de “Instalar”. Después, se comenzará a instalar el programa.

C. ¿Cómo usar WAMP?

Cuando ya se haya finalizado la instalación, podemos comenzar a trabajar con él, primeramente, debemos que abrir el programa, el programa que nosotros debemos utilizar es “WAMP control panel”, la interfaz se muestra tal y como aparece en la figura 5.1

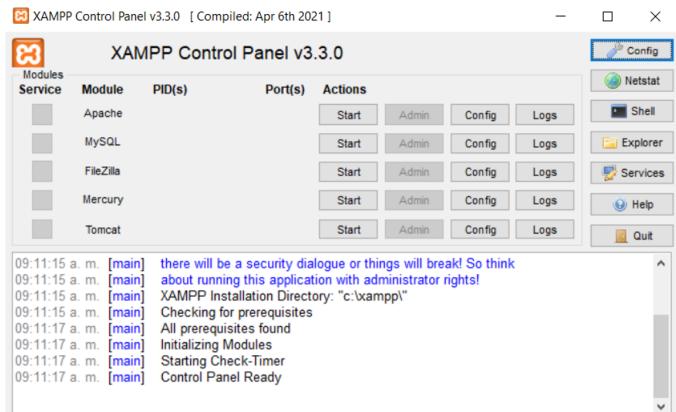


Figura 5.1

Como podemos observar en la figura 5.2 tenemos varios servicios para usar nuestro servidor local, para comenzar a visualizar nuestros proyectos debemos iniciar los servicios de Apache y MySQL.

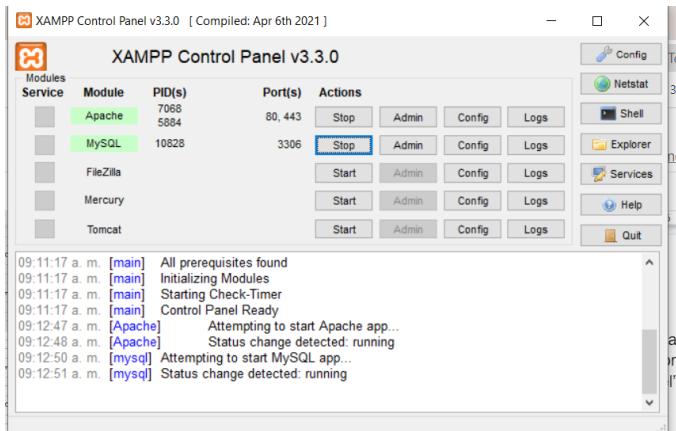


Figura 5.2

Ahora debemos dirigirnos a nuestro navegador y comprobar que, si se está haciendo uso de estos servicios correctamente, en este caso nos dirigiremos a Google Chrome y en la barra de búsqueda escribiremos “localhost” y damos “ENTER”, si todos los pasos fueron perfectos, nos mostrará una pantalla tal cual como en la figura 5.3. En la barra de navegación tenemos varias opciones

pero nosotros solo usaremos phpMyAdmin para gestionar todos nuestros registros.



Figura 5.3

phpMyAdmin nos brinda múltiples opciones para usar, en la figura 5.4 podemos ver opciones para gestionar nuestra base de datos creada u otras opciones que son para conocer más información acerca del programa y recomendaciones de uso.



Figura 5.4

XIV. SQL

A. ¿Qué es SQL?

Structured Query Language, o por su significado en español, lenguaje de consulta estructurado, es un tipo de lenguaje de programación que permite manipular y cargar datos desde una base de datos. Es capaz de realizar álgebra y aritmética avanzada.



B. SQL y MySQL

SQL es un lenguaje para trabajar con bases de datos y MySQL es un sistema de gestión de bases de datos. Es la principal diferencia entre uno y otro. Ambos conceptos están íntimamente relacionados: MySQL es un programa de gestión de bases de datos cuyo lenguaje es SQL. MySQL es un sistema de gestión de bases de datos relativos (RDBMS) de código abierto respaldado por Oracle y basado en el lenguaje de consulta estructurado (SQL).

XV. PHPMYADMIN

A. PHPMYADMIN y MySQL

Cuando se trata de herramientas de administración de bases de datos MySQL, la primera opción que viene a la mente es phpMyAdmin.

PhpMyAdmin es una aplicación web utilizada para administrar bases de datos MySQL en una interfaz simple e intuitiva. Es un programa PHP muy popular. La ventaja de utilizar una aplicación web es que nos permite conectarnos a servidores remotos, a los que no siempre se puede acceder a través de programas GUI.

Para usar phpMyAdmin, simplemente cargue el conjunto de archivos PHP que componen la aplicación en el servidor web, configure el inicio de sesión de MySQL y comience a administrar la base de datos. Con phpMyAdmin, puede realizar cualquier tipo de operación, desde crear y eliminar bases de datos hasta administrar tablas (creación, modificación, eliminación) y, por supuesto, sus datos.

Lo que es muy evidente al utilizarla es el hecho de que todas sus instrucciones o botones se escriben automáticamente en lenguaje de sql cuando se oprimen los botones correspondientes, este lenguaje se refleja en un apartado superior.

B. Implementación

El equipo de desarrollo de BACKEND implementó una base de datos con múltiples tablas donde se guardan datos sacados por otras carreras, y por las experimentaciones, donde en total tenemos 6 tablas con diferentes datos, así como el sitio llamado "experimentación", destinado a ser un CRUD de las múltiples tablas contenidas en las bases de datos, con CURD me refiero a un conjunto de funcionalidades que ofrece la página en relación a la página web, CRUD es un acrónimo que significa Create, Read, Update, Delete; es decir, da la posibilidad o la opción de crear nuevos registros, leer los registros ya contenidos, actualizar o modificar los registros ya contenidos o borrarlos, El CRUD es un concepto básico de funcionalidad que una página web debería poder hacer, por lo tanto algo básico que un programador de páginas web debe poder hacer y un aspecto básico de backend, por lo tanto el usuario debe poder crear, leer, actualizar y borrar registros de ciertas tablas, ya que la base de datos engloba mucha información esto de elegir que se puede modificar es base para la jerarquización y el control de privilegios dentro de una página web, en este caso el usuario promedio que sería cualquier integrante del equipo debe poder acceder a las tablas relacionadas con experimentaciones y nada más, de esa forma se guarda cierta información vital para el funcionamiento, como lo es la tabla de users.

La tabla de users es principal y únicamente utilizada por el login de la página, ya que controla quién tiene acceso a los privilegios de usuario y quien no.

Al ser esta creada con el objetivo que solo un grupo selecto de gente tenga acceso a los registros de la base de datos se quita la posibilidad que haya un signup, es decir, una opción de registrar un nuevo alumno, por lo tanto, los usuarios son suministrados por el administrador que tiene acceso al core de la base de datos, al phpMyadmin, acceso directo a todos los registros, el propietario de los datos, puede ingresar nuevos usuarios por medio del GUI de phpMyadmin y también borrarlos, y ya que el login

soloamente deja ver el acceso a experimentación sólo si el usuario ingresado en login es concordante con alguno dentro de la tabla de users a la que solo puede ingresar el administrador, se hace una exitosa encapsulación de información por medio del código de php.

XVI. Implementación

A. Propósito en el proyecto

Se planteó la creación de una página web en primera instancia para guardar en esta un acceso a una base de datos donde los miembros del equipo pudieran guardar los resultados de sus experimentos, la razón para que esta base de datos estuviera en un ambiente en línea era porque los experimentadores estarían en diferentes laboratorios y se requería una forma de guardar todo en un mismo sitio a pesar de las distancias físicas con las que nos enfrentamos.

Seguido a esto nos propusimos no solo hacer una página útil para los miembros del equipo si no también una página web que sirviera como presentación de la propuesta que estamos ofreciendo, siendo esta la mejora del asfalto a través del residuo del café.

Fue por esto que dispusimos una página principal abierta a cualquier persona, que explicaba el fundamento principal de nuestro proyecto y una segunda página donde se exhiben los procedimientos que llevamos a cabo

Por último una vez que se le dió un uso a la sección para miembros del equipo tuvimos datos guardados en nuestra base de datos, estos datos en adelante fueron los usados para el análisis de datos.

B. Organización

La organización fue una parte vital de este proyecto en el caso de esta carrera, es famosa la frase de que dos cabezas trabajan mejor que una, y en esta carrera tenemos 23 cabezas, y aunque

tengamos tanto potencial, si no lo dirigimos tendremos serios problemas, por eso mismo se decidió trabajar en equipos con objetivos y percepciones diferentes, se crearon 4 equipos oficialmente, el equipo de frontend, el de backend, el equipo de documentación y alguien aprendiendo manim para implementarlo en el video final, aun así al principio hubo problemas de comunicación, ya que aunque front y back tienen objetivos dentro del trabajo diferentes, también tienen que trabajar con mucha comunicación, ya que están haciendo dos partes de lo mismo, se necesitan el uno del otro, así no bien entradas las vacaciones por fin se tuvo una reunión en la que se relegaron trabajos importantes para iniciar lo que terminó siendo la página web.

Como mencione antes, una de las primeras cosas que se hizo antes de entrar en vacaciones de semana santa y de pascua, fue relegar equipos para aprovechar todas nuestras cabezas de forma eficiente, y que cada equipo pensara en soluciones a sus problemas asignados, algunos de esos problemas en el equipo de backend fue el de decidir que lenguaje gestor de bases de datos elegimos para empezar a ingresar información a un servidor, lo que generó algunos problemas, ya que no muchos de los integrantes sabían o habían trabajado antes con bases de datos o con páginas web, así que al principio la idea de conectar la base de datos con una posible página web parece algo complicado y un poco fuera de nuestros conocimientos, pero después de unas juntas de front donde se acordó que tendrían las plantillas en html listas, los que tenían conocimientos de php tuvieron una mejor idea de como conectar una base de datos con las plantillas enlazadas de HTML.

XVII. Codificación

La mayoría de las siguientes páginas se dividen en controladores y vistas, las vistas es lo que involucra todo lo que se conecta directamente con el usuario (como campos a llenar, botones, listas, etc) mientras que el controlador se trata de lo que hace el

programa por detrás conectando con la base de datos.

Connect.php:

```
<?php
$host = "localhost";
$user = "id18818891_root";
$pass = "#proyectoEquipo2";
$db = "id18818891_caferoad";

$conn = mysqli_connect($host, $user, $pass, $db) or die("Connection Failed" .mysqli_connect_error());
```

El archivo de connect es un archivo que funcionará como cabecera para el resto de los programas, en este indicamos la conexión con la base de datos. La razón de que se encuentre como un archivo aparte es para optimizar el código y en caso de que haya cambios a la base de datos (como fue en el caso de las pruebas) poder hacer un solo cambio en lugar de varios en distintas páginas.

Agregar (vista):

```
<div class="contenedor-agregar-formulario">
<div>
  <h2>Agregar registro</h2>
  <br>
  <form action="agregar_v.php" method="post">
    <p><strong><a href="#">1</a></strong> Selecciona el proceso donde agregar.</a01></p>
    <select name="tables">
      <?php
        include("connect.php");
        $result = $conn->query("SHOW TABLES");
        while($valores = mysqli_fetch_array($result)){
          echo "<option value='".$valores[0]."'>".$valores[0]."</option>";
        }
      >
    </select>
    <input type="submit" name="submit" value="Choose options">
  </form>
</div>
```

Obviando las cabeceras que tienen todas las páginas lo primero que podemos visualizar en la página de agregar es una primera lista desplegable donde se nos pide que seleccionemos el proceso a agregar, esta lista funciona gracias a una instrucción php donde enviamos una query a la base de datos donde le pedimos nos muestre los nombres de todas las tablas (que como hablamos antes en este documento son los nombres de todos los procesos que haya ingresado el usuario).



A donde se envía la selección del usuario una vez selecciona una tabla e indica “enviar” es a la misma

página donde nos encontramos, esto debido a la próxima sección de código:

```
<form action="agregar_c.php" method="POST">
<fieldset>
<?php
include("connect.php");
$qry = "SHOW COLUMNS FROM ".$_REQUEST['tables'];
$result = $conn->query($qry);
echo '<input type="text" name="tabla" value="'.$_REQUEST['tables'].'" readonly="readonly"/> <br><br
>';
while($valores = mysqli_fetch_array($result)){
    if($valores[0]!="ID"){
        echo '<div class="contenedor-campo">';
        echo '<label for="'.$valores[0].'"><strong>'.$valores[0].': </strong></label>';
        echo '<input type="number" step=0.00001 name="rty[]" id="rty[]" autocomplete="off"/>';
        echo '</div>';
    }
}
<br>
<input type="submit" name="submit" id="submit" />
</fieldset>
</form>
```

En la siguiente sección de código podemos encontrar otro form, en este form lo que hacemos es mostrar las columnas de la tabla requerida (la tabla requerida la recibimos al entrar a la página y por eso es importante que en el botón para entrar se envié esta información), las columnas como ya lo habíamos mencionado antes no son más que los parámetros de medición así que le indicaremos al usuario con un texto el campo que estamos requiriendo y con un cuadro de texto junto a estos textos pediremos los datos que se quieren llenar ahí, estos ingresos solo son numéricos cualquier otro intento daría un error. La forma en que este programa muestra las columnas es mediante un for mientras haya existencias por lo que no importa la cantidad de columnas que tenga la tabla esta se mostraran por igual.

Una vez el usuario llene los datos el programa enviará esta información en forma de arreglo hacia la página de agregar_c el cual es el controlador de esta misma página

Agregar (Controlador):

```
<?php
include("connect.php");
$tabla = $_REQUEST['tabla'];
$sql = "INSERT INTO `".$tabla."` VALUES (NULL);"

$rty = $_POST['rty'];
if (is_array($rty))
{
    for ($i=0;$i<sizeof($rty);$i++)
    {
        $sql .= ",".$rty[$i] . "";
    }
}
$sql .= ")";
// print_r($sql);
if($conn->query($sql) === true){
    echo "<h1>SE INSERTO CORRECTAMENTE<h1>";
} else{
    die("Error al insertar: " . $conexion->error);
}
?>
<br><br><br>
<a href="index3.php"><strong>REGRESAR A INICIO</strong></a>
```

El controlador de la página agrega lo que hace es recibir la información anteriormente mencionada y también el nombre de la tabla. Con estos datos lo que hace es crear un texto que en realidad es una instrucción query, esta instrucción se ejecuta gracias a un comando de php, si da error esto se mostrará y si pudo ejecutarse la función igualmente se mostrará junto con un vínculo para regresar al índice de la experimentación .

Borrar (vista):

```
<form action="borrar.php" method="POST">
<fieldset>
<div class="contenedor-campo">
<label for="eliminar">SELECCIONA DE QUE PROCESO ELIMINAR:</label>
<select name="tables">
<?php
    include("connect.php");
    $result = $conn->query("SHOW TABLES");
    while($valores = mysqli_fetch_array($result)){
        echo "<option value='".$valores[0]."'>".$valores[0]."</option>";
    }
?
</select>
</div>
<div class="contenedor-campo">
<label for="eliminar">INGRESE EL 'ID' DEL 'EXPERIMENTO' QUE DESEA BORRAR:</label>
<input type="text" name="eliminar" id="eliminar" autocomplete="off" required/>
</div>
<input type="submit" name="submit" id="submit" />
</fieldset>
</form>
```

Lo primero que se encuentra en la página de borrar es, al igual que en el método anterior, una lista desplegable, que funciona gracias a una query donde se muestran las tablas, solo que en esta ocasión no se separa la lista con un botón de “enviar” que abre la misma página, esto no hace

falta ya que solo se necesita para cambiar los datos que se están mostrando, ya que no importa de cual tabla se trata en cualquier caso se pide el ID entonces no es necesario este botón. Habiendo mencionado esto justo abajo se muestra lo que justo se mencionó, un texto donde se le indica al usuario que debe escribir el ID y un cuadro de texto donde se indicará el número del ID. La información de la tabla y el ID a eliminar se envía a la página de borrar.php el cual es el controlador de la página borrar.

Nota: La forma en que el usuario pueda conocer cuál es el ID está relacionada justo con el punto donde se encuentra el vínculo a esta página, en el mostrar, ya que justo ahí podría observar cual es el ID que desea eliminar.

Borrar (controlador):

```
<?php
include("connect.php");
if($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['submit'])){
    $conn = mysqli_connect('localhost', 'root', '', 'caferoad') or die("Connection Failed" .mysqli_connect_error());
    if(isset($_POST['eliminar'])){
        $eliminar = $_POST['eliminar'];
        //mysql_query($conn, "DELETE FROM `experimento` WHERE ID = '$eliminar'");
        //query = mysqli_query($conn, $sql);
        $qry = "DELETE FROM `$_REQUEST['tables']` WHERE ID = '$eliminar'";
        if(mysqli_query($conn, $qry)){
            echo "<h1>DATO BORRADO CORRECTAMENTE</h1>";
        }
    } else{
        echo "<h1>EL 'ID' INGRESADO NO EXISTE :(</h1>";
    }
}
?>
<br><br>
```

El controlador de la página borrar lo que hace es recibir la información de la tabla y del ID a eliminar con esta información se hace una query de Delete que se ejecuta con el comando de php en la base de datos, si la función tiene algún error muestra el error de que el ID no existe (ya que es el único error al que se puede llegar por como está escrito el programa) y de igual forma si la función se ejecutó correctamente lo único que hace es mostrarlo en pantalla y después mostrar un vínculo para regresar al índice de la sección de experimentación del programa.

Index3.php:

```
<div>
    <button class="Agregar_Pro">
        <a class="agregar-boton" href="procesos_vista.php"><a1>Agregar Procesos</a1></a>
    </button>
</div>
<br><br>
<div>
    <form action="agregar_v.php" method="post">
        <button type="submit" class="tables" name="tables" value="Angulo_Fase_experimento"><a2>Agregar Registros</a2></button>
    </form>
</div>
<br><br>
<div>
    <form action="mostrar.php" method="post">
        <button type="submit" class="tables" name="tables" value="Angulo_Fase_experimento"><a3>Mostrar Registros</a3></button>
    </form>
</div>
```

El index3 funciona como menú de las operaciones que pueden usar los miembros del equipo, lo que se hace en este menú es mostrar tres vínculos a las funciones más importantes del programa que son primero que nada agregar procesos, pantalla en la cual, como dice su nombre, se agrega un nuevo proceso, este es el único vínculo a una página que no necesita recibir nada así que el botón solo se trata de eso, un simple botón que contiene un vínculo hacia la página dentro de él. El siguiente botón es hacia la página de agregar registros, ya que esta página necesita recibir la información de una tabla desde el inicio lo que se hace es usar un botón del tipo submit dentro de un form, de esta forma el botón enviará la información bajo el nombre de "tables" y el contenido que enviará es "Angulo_Fase_experimento" que es una tabla antes agregada durante el proceso del experimento (cabe aclarar que para unas primeras pruebas se enviaba de contenido una tabla prueba que usamos para programar la base de datos).

El último botón y en varios casos el más usado es el botón que envía a la pagina de mostrar registros, esta es otra pagina que requiere de recibir información al momento de abrirla y es por esto que, de igual forma que en el caso pasado, se envía la información del nombre de la tabla elegido.

mostrar.php.

La pagina de mostrar es una especial que se decidió no sería dividida en controlador y vista, la razón de esto fue que, como dice el mismo nombre de la página, se trata de una página donde se expone lo que se tiene guardado en la base de datos, siendo una página donde el usuario está tan cercano a esta

base de datos era más sencillo dejar el código como uno mismo.

```
<div class="contenido">
  <div class="contenedor-lista">
    <div class="">
      <h2><a24>Lista de productos</a24></h2>
      <form action="agregar_v.php" method="post">
        <button class="agregar-botón" name="tables" value="Angulo_Fase_experimento">Agregar</button>
      </form>
      <form action="borrar_v.php" method="post">
        <button class="agregar-botón" name="tables" value="Angulo_Fase_experimento">Borrar</button>
      </form>
      <!-- <a class="agregar-botón" href="agregar.html">Agregar</a> -->
    </div>
```

Una vez explicado esto pasamos al código, lo primero que podemos encontrar en el código son los dos botones que aparecen en la parte superior derecha de la página, estos botones no son más que vínculos trabajados como se explicó anteriormente en el index3, siendo estos unos botones del tipo submit que envían la información de la tabla escogida para ser la primera en mostrar.

```
<form action="mostrar.php" method="post">
  <a24><strong>Selecciona el proceso que mostrar.</strong></a24>
  <select name="tables">
    <?php
      include("connect.php");
      $result = $conn->query("SHOW TABLES");
      while($valores = mysqli_fetch_array($result)){
        echo "<option value='".$valores[0]."'>".$valores[0]."</option>";
      }
    <?>
    </select>
    <input type="submit" name="submit" value="Choose options">
  </form>
```

Lo siguiente se trata de un trozo de código que ya hemos visto a lo largo del documento, este código se trata de una lista desplegable que muestra todas las tablas en la base de datos (que recordemos son los procesos guardados) junto con un botón para enviar la información cuando queramos cambiar la tabla sobre la cual estamos desplegando la información.

```
<?php
echo "<h3> TABLA QUE SE MUESTRA: <strong>".$_REQUEST['tables'].</strong></h3><br></a31>";
include("connect.php");
$qry = "SELECT * FROM ".$_REQUEST['tables'];
$qry2 = "SHOW COLUMNS FROM ".$_REQUEST['tables'];
$result = $conn->query($qry);
$result2 = $conn->query($qry2);
$result = $result->fetch_all();
echo '<table id="lista-de-productos">';
echo "<tr>";
while($valores = mysqli_fetch_array($result2)){
  echo "<td>".$valores[0]."</td>";
}
echo "</tr>";
for($i=0;$i<sizeof($all);$i++)
{
```

```
  echo "<tr>";
  for($j=0;$j<sizeof($all[$i]);$j++)
  {
    echo "<td>".$all[$i][$j]."</td>";
  }
  echo "<td>";
  echo '<div class="contenedor-opciones-enlaces">';
  echo '<a href="#">.</a>';
  echo '</div>';
  echo '<div class="contenedor-opciones-enlaces">';
  echo '<a href="#">.</a>';
  echo '</div>';
  echo '<div class="contenedor-opciones-enlaces">';
  echo '<a href="#">.</a>';
  echo '</div>';
  echo "</td>";
}
echo "</tr>";
}
echo "</table>";
?>
```

Por último y siendo esta sección el cuerpo del programa en el programa se hace una tabla, dentro de esta tabla se presentan todos los datos de la tabla seleccionada, para saber que tabla es esta esa información se recibe con un request, lo primero que se hace es pedir el nombre de las columnas, para que estos sirvan como encabezados en la tabla, y con esta información de la tabla se envía también una query pidiendo regresar toda la información guardada en la tabla indicada, esta información que regresa se divide y se le da un formato para imprimirlo en la misma tabla final, dando como resultado una tabla del proceso con encabezados de los datos y con la información guardada antes por el usuario.

TABLA QUE SE MUESTRA: Modulo_reologico_Kpa_medicion				
ID	porcentaje lignina	porcentaje lignina	porcentaje lignina	porcentaje lignina
1:38	1.85	1.94	2.05	1

Agregar Procesos (vista):

```
<div class="contenedor-agregar-formulario">
  <div>
    <h2>Agregar procedimiento</h2>
  </div>
  <form action="procesos_controlador.php" method="post">
    <fieldset>
      <div class="contenedor-campo">
        <a8>Título del procedimiento:</a8><input type="text" name="nombre"><br>
      </div>
      <a9>Parametros del experimento</a9> <br>
      <div class="contenedor-campo">
        1: <input type="text" name="qty[]"/>
      </div>
      <div class="contenedor-campo">
        2: <input type="text" name="qty[]"/>
      </div>
      <div class="contenedor-campo">
        3: <input type="text" name="qty[]"/>
      </div>
      <div class="contenedor-campo">
        4: <input type="text" name="qty[]"/>
      </div>
    </fieldset>
  </form>
```

```
Parametros de la medición <br>
<div class="contenedor-campo">
1: <input type="text" name="qry[]">
</div>
<div class="contenedor-campo">
2: <input type="text" name="qry[]">
</div>
<div class="contenedor-campo">
3: <input type="text" name="qry[]">
</div>
<div class="contenedor-campo">
4: <input type="text" name="qry[]">
</div>
<input type="submit" value="Confirmar">
</fieldset>
</form>
```

La parte gráfica de la página de agregar procesos no es más que un formulario con 8 campos de texto, dos de ellos para parámetros del experimento (mediciones de los elementos antes de la ejecución del experimento) y los otros dos parámetros de la medición (mediciones del resultado después de la ejecución del experimento), cada uno de estos datos se guardan bajo el nombre de qry[], siendo estos corchetes una instrucción que indica que en realidad todo fue guardado como arreglo para futuramente ser enviado al controlador de la misma página.

Agregar procesos (controlador):

```
<?php
include("connect.php");

$nombre = $_REQUEST['nombre'];
$sql = "CREATE TABLE " . $nombre . " _experimento(ID INT(10) AUTO_INCREMENT PRIMARY KEY";
$qty = $_POST['qty'];
if (is_array($qty))
{
    for ($i=0;$i<sizeof($qty);$i++)
    {
        $sql = $sql . "," . $qty[$i] . " FLOAT(11)";
    }
}
$sql .= ")";

if($conn->query($sql) === true){
    echo "<h1>EL PROCESO HA SIDO CREADO</h1>";
} else{
    die("Error al crear la tabla: " . $conn->error);
}
```

Con la información recibida de parte de la vista de la página lo que hace el programa es crear una tabla con el nombre que indico el usuario seguido por el texto “experimentación” y en esta tabla se crean distintas columnas con el nombre que indico el usuario para los parámetros de experimentación, la query se ejecuta y si funcionó no se hace nada, si hay error este se escribe.

```
$sql = "CREATE TABLE " . $nombre . " _medicion(ID INT(10) AUTO_INCREMENT PRIMARY KEY";
$qry = $_POST['qry'];
if (is_array($qry))
{
    for ($i=0;$i<sizeof($qry);$i++)
    {
        $sql = $sql . $qry[$i] . " FLOAT(11)";
    }
}
$sql .= "Experimento INT(10)";

if($conn->query($sql) === true){
    // echo "La tabla se creó correctamente...";
} else{
    die("Error al crear la tabla: " . $conn->error);
}
```

De forma casi idéntica a la instrucción pasada se crea una query destinada a crear una segunda tabla, en esta ocasión una con terminación “_medicion” y que en esta ocasión guardará como columnas los parámetros de medición indicados para el proceso. De igual forma que en el caso anterior si la función se ejecutó correctamente no sucede nada pero si sucede algún error este se muestra en pantalla con un texto.

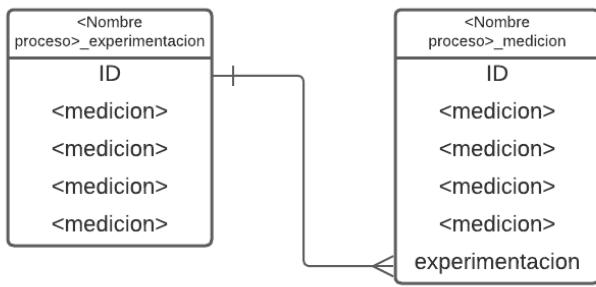
```
$sql = "ALTER TABLE " . $nombre . " _medicion add foreign key (Experimento) references " . $nombre . " _experimento
(ID)";
if($conn->query($sql) === true){
    echo "<h1>EL PROCESO HA SIDO CREADO</h1>";
} else{
    die("Error al crear la tabla: " . $conn->error);
}
```

Por último se altera la tabla de terminación “medición” para agregarle una llave foránea (esta llave ayudará a relacionar esta tabla con la tabla del mismo nombre con terminación “experimento” si funciona ahora si finalmente se enviará un mensaje de éxito, pero por otro lado si se encuentra algún error se mostrará un mensaje de error.

XVIII. Diseño base de datos

Al principio se presentó una idea de lo que se quería que hiciera la página web, y en específico la base de datos, se decidió que se quería un ambiente donde poder crear nuevos procesos, estos procesos siendo los que se ejecutaban en la experimentación, se nos explicó en la base de datos se deberían crear dos “tablas” diferentes donde se guardarán diferentes parámetros de medición que se llevaban a cabo durante la experimentación, estos siendo las mediciones de los elementos que se usaban para hacer el proceso y por otro lado las mediciones de los elementos que resultaban al terminar el proceso.

Desde ese momento entre los miembros del equipo se decidió con qué lenguajes se trabajaría, entre las diferentes técnicas y programas que podíamos usar se llegó a un consenso de hacer uso de MySQL, fue por esta razón que al diseñar el cómo se usaría la base de datos se tuvo que tener en cuenta la estructura de tablas relacionales propia de MySQL. Con esto en mente se formó la idea de que dentro de nuestra base de datos cada proceso fueran dos tablas, divididas en los datos de medición antes de la experimentación y los datos de medición después de la experimentación. Fue así que dentro de la estructura de nuestra base de datos cada vez que el usuario ingresa un procedimiento internamente se están creando las dos tablas, una con la terminación _experimentacion y otra con la terminación _medicion, ambas relacionadas mediante una llave foránea, estas tablas tienen de campos los ingresados por el mismo usuario al momento de crear el proceso, y con eso ya se tienen las tablas donde más adelante se podrá ingresar información cuando lo requiera el usuario.



XIX. Documentación

A. Inicio de proyecto

Primeramente, se definió al equipo que trabajaría en el Back-end y Front-end del proyecto.

Para la realización del proyecto se utilizó el entorno de Visual Studio Code que es un programa para codificación de proyectos creado por Microsoft, queríamos sistema para uso de un servidor local el cual fue WAMPserver, el lenguaje de programación utilizado fue PHP mientras hacía consultas e

instrucciones con base a MySQL para la gestión de nuestra base de datos, para lo visual de nuestro proyecto usamos el lenguaje de marcado de hipertexto (HTML) y para su diseño utilizamos hojas de estilo CSS pero con frameworks de bootstrap para diseño de páginas web, y por último, al tener listo nuestro proyecto funcional de manera local, necesitamos subirlo a un servidor web para que cualquier persona pudiera tener acceso a él, y no tan solo de la computadora donde lo realizamos, el servidor gratuito que usamos fue 000webhost el cual nos permite alojar nuestro sitio web de repartición broadcast y además nos permite tener uso de gestor de bases de datos con una phpMyAdmin. En general, estos fueron los elementos requeridos para llevar a cabo el proyecto.

B. Administración de archivos

Todos los archivos que utilicemos para el proyecto de preferencia deben estar guardados en una sola carpeta, dentro de los archivos que conforman al programa wamp existe una carpeta llamada “htdocs”, es muy importante que nuestro proyecto se encuentre dentro de esa carpeta ya que solo así podremos visualizarlo de manera local.

C. Creación de base de datos (creardb.php)

Antes de todo debemos crear un archivo .php para crear nuestra base de datos, en este debemos un bloque de código en el cual debemos definir 3 variables, estas son:

- \$servidor = “localhost”;
- \$nombreusuario = “root”;
- \$password = “”;

Estas 3 variables son el nombre del servidor al que nos vamos a conectar, el nombre del usuario que va a trabajar y su contraseña, si nos damos cuenta, root es el usuario preestablecido que generamos cuando hicimos la instalación de nuestro wampserver, y no optamos por ponerle contraseña ya que no corremos riesgo si alguien hace uso de este registro.

Ya tenemos definidos nuestras variables, es hora de hacer una consulta y comunicar nuestro código PHP con MySQL, para esto haremos una consulta la cual la llamamos con la función “mysql(\$var1, \$var2, \$var3)”, en cada parámetro que se necesitamos vamos a poner nuestras variables creadas, el resultado de esta función lo guardamos en otra variable llamada \$conexión y es aquí donde nos aseguramos que tenemos una conexión correcta hacia phpMyAdmin y podemos realizar cambios que si se vean reflejados. Posteriormente, comprobamos que la conexión sea exitosa y si es así, ahora hacemos una nueva consulta con una instrucción MySQL la cual es “CREATE DATABASE Caferoad”, con esto creamos nuestra base de datos para poder trabajar con ella y si la creación de ella fue exitosa, se mostrará un mensaje de que fue creada exitosamente. Ahora sí, podemos trabajar con ella.

D. Conectarnos a la base de datos (connect.php)

Solamente necesitamos crear un archivo el cual nos permite conectarnos en cualquier momento con la base de datos, este archivo tenemos 4 variables las cuales son:

- \$host = “localhost”;
- \$user = “root”;
- \$pass = “”;
- \$db = “Caferoad”;

Estas variables son los parámetros para la función “mysqli_connect()” en donde vamos a pasar nuestras variables separadas por comas y tenemos una alternativa, que si en caso de no realizar una conexión correcta tenemos una función llamada “die()” la cual nos muestra un mensaje de error y obviamente debemos revisar algo más.

E. Página principal (index.php)

Esta página siempre va a ser la primera en leerse por nuestro servidor, cuando queramos abrir

el proyecto index.php es la primera página que se mostrará, esta página tiene una maquetación en HTML, pero es un archivo con extensión .php, esto se debe a que también usaremos comandos php que nos ayudarán a hacer la página web interactiva, ya que sin este código, la página se vuelve meramente informativa.

Para nuestra página principal (index) colocamos imágenes, párrafos de textos, elementos con eventos por el cursor (hover), y nuestra información hasta el pie de página. También decidimos colocar una barra de navegación hasta arriba de la página que nos traspase hasta el apartado deseado y un inicio de sesión que funciona con php, y nos direcciona al inicio de sesión.

F. Inicio de sesión con php

En nuestro archivo index al inicio del código tenemos el bloque de código con el que indicamos que durante la navegación de la página podemos tener un usuario habilitado hasta que este se desabilite con un cierre de sesión. Primeramente, decimos que podemos iniciar sesión con la función “session_start();” con esta necesitamos un archivo que nos brinde una conexión con nuestra base de datos, este archivo lo llamamos “DataBase.php” y cumple casi el mismo objetivo de “connect.php”, solo que en este ya usamos un evento que se debe cumplir, lo que es cierto sobre estos dos archivos, el “DataBase.php” y “connect.php”, es que los dos crean una conexión con la base de datos, pero de diferentes formas, ya que el método implementado en “DataBase.php” devuelve un objeto de sql que es compatible con las sentencias implementadas en el “index.php” y el “index2.php”, y el método utilizado en el archivo de nombre “connect.php” arroja otro objeto incompatible con las sentencias condicionales y de búsqueda en el documento “index2.php”, por lo tanto arroja un error inmediato, de sintaxis y compatibilidad de métodos, este error de compatibilidad de métodos se originó en el espacio de tiempo entre la creación del espacio de

“experimentación”, y para no volver a hacer otro archivo de php que fuera compatible con el archivo de nombre “connect.php”, se tomó la decisión que era más factible crear el archivo “DataBase.php” para que funcionara en conjunto con el “index2.php” y el “index.php” en la validación de usuarios para que diera paso a experimentación .

Como se menciona anteriormente, el propósito de este código se hace con el objetivo de limitar la accesibilidad de las personas que entren a la página, para que solo un conjunto de usuarios determinado fuera capaz de acceder a la página y modificar la base de datos, por lo tanto la página no cuenta con la implementación de un forma de registrar nuevos usuarios, sino que esa parte se guarda para los administradores de la página

Lo siguiente que se hace es usar una condicional “if()” donde se pregunte el valor del arreglo “\$_session”, ya que este arreglo contiene el valor del valor que se consultó, en este caso fue el “id_user” que tenemos registrado en la tabla de “users” almacenada en nuestra base de datos, después, si el “id_user” es válido, debemos obtener todos los datos del usuario utilizando una variable y esta variable guardará todos los datos del “id_user” ingresado, aquí nosotros ya tendremos sesión iniciada y nos permitirá acceder a nuestra base de datos para poder trabajar con ella.

G. Importación de base de datos a servidor

Como ya se sabe, se estuvo trabajando primero el proyecto de forma local, se hicieron pruebas, se modificaron y mejoraron detalles, y cuando ya toda la conexión estaba realizada tuvimos que exportar todos nuestros, para esto desde phpMyAdmin podemos exportarla en formato .sql, ya teniendo el archivo listo, nos dirigimos ahora a phpMyAdmin del servidor (000webhost, este nos permite administrar igual que de forma local nuestro proyecto) pero tuvimos que hacer unos cambios, nuestra base de datos no iba a llamarse

igual que de forma local, se iba a implementar un usuario y contraseña, entonces al generar estos se tuvo que hacer un cambio en los archivos de conexión para evitar algún fallo, ya que al no hacerlo, nuestro proyecto no iba a funcionar subido en la web. Después de hacer todos estos cambios nos dirigimos a phpMyAdmin e importamos nuestra base de datos, esperamos a que se cargaran y ahora si podemos continuar con el trabajo en ella.

H. Experimentación (Base de datos)

Dentro de la barra de navegación contamos con nuestro enlace a “Experimentación” donde tenemos 3 botones establecidos en los cuales podemos dirigirnos a cada uno de las actividades que podemos realizar, los botones son:

- Agregar Procesos: Podemos agregar procesos que durante la experimentación y pruebas que se obtuvieron en cada proceso donde se llevaba a cabo el mismo objetivo, de la creación de asfalto con lignina. Aquí cualquier integrante del equipo puede añadir un nuevo proceso y colocar datos y valores obtenidos con cada prueba realizada y así seguir utilizando esa tabla.
- Agregar Registros: Aquí se elige en qué proceso se hizo una prueba y se agregan los valores que se obtuvieron nuevamente con otro experimento. Esto nos permite no estar creando tablas cada vez que se necesita registrar algo, ya que nos ahorra recursos y podemos reutilizar tanto código como almacenamiento. Además, nos permite que cada integrante pueda ver anteriores pruebas y comparar resultados.
- Mostrar Registros: Un complemento de ambos apartados anteriores, en Agregar Proceso/Registro ingresamos valores, pues ahora estos valores pueden ser mostrados obviamente al usuario para que tenga el conocimiento de lo que registro y los anteriores registros. Además, en este

apartado tenemos también las opciones de agregar y eliminar registro, los cuales son funciones muy útiles porque cualquier registro que no se desea o no aporta, puede ser removido de la base de datos.

En todos ellos se puso una etiqueta de “<select>” en la cual añadimos todas las tablas que tenemos, para mostrar cada uno de los verdaderos nombres de estas se usa una combinación entre código de HTML y php, en donde el contenido es traído por medio de php hacia elementos HTML que nos permiten interactuar de mejor manera con todos nuestros datos, entonces al elegir que tabla necesitamos solamente debemos desplegar la lista, seleccionar la tabla y enviar la consulta, automáticamente nuestro sistema hará la tarea correcta.

XX. Optimización

A. El sitio web.

Para trabajar en la optimización del sitio web hay que tener en cuenta estas variables:

- La elección del dominio. Aquí lo mejor es escoger un .com, aunque todos los dominios se posicionan de forma similar.
- Pensada para satisfacer las necesidades del usuario. En este punto es centrar bien la información, no abusar en exceso de los textos y nunca implementar elementos flash o música automática, ya que son prácticas que pueden ser penalizadas por los buscadores.
- Evitar las páginas duplicadas. Un claro ejemplo es cuando para acceder a una web se puede hacer desde diversas URL(por ejemplo, con o sin el www).

B. La estructura de una web.

Hacerlo de forma piramidal, de tal manera que de la home salgan páginas enlazadas, y estas, a su vez, enlacen a otras páginas de categorías de manera que

la autoridad de la home pueda fluir hacia el resto de la página.

Otro punto a tener en cuenta para mejorar la navegación de los usuarios por la web es añadir un mapa del sitio web en el que se puedan ver de un vistazo todas las páginas que se incluyen dentro de la página. Todo esto para ayudar a mejorar la naveabilidad de los usuarios y a que los buscadores puedan indexar mejor las páginas de un site.

C. Las páginas de contenido

Tener contenido actualizado que trate sobre los servicios y/o productos. Todos ellos tienen que estar redactados y acompañados, siempre que sea posible, de material gráfico que irá debidamente etiquetado con la keyword que se quiere posicionar.

Además, en la medida de lo posible, se debe de incluir en el contenido enlaces hacia otras páginas de la misma web.

XXI. Mejoras a futuro

La página web fue diseñada para que en un futuro pudiera ser mejorada, dándole un mejor funcionamiento y siendo capaz de almacenar más información de importancia, una vez que el proyecto esté concluido.

Entre las posibles mejoras a futuro se encuentra un claro ejemplo de el aumento de paginas principales disponibles a todo el publico, paginas donde una vez se encontrará alguna línea de tendencia más fiable podría presentarse por ejemplo.

Otra posible mejora a posible es la capacidad de escoger distintos datos a ingresar al momento de ingresar un procedimiento, ya que en el momento el número se encuentra fijo, resultando esto en un problema si es que se desea usar más o menos datos, sin embargo la forma en que está programada la parte lógica del programa hace muy sencillo hacer el cambio a aceptar más o menos datos, el

único inconveniente sería lograr hacer que se presentaran más o menos campos a llenar en la página que se le presenta al usuario

Por último y siendo esto lo más importante sería lo ideal lograr en un futuro que la misma página con la base de datos fuera capaz de calcular mediante algún proceso matemático la línea de tendencia que se tiene entre los datos, para así a más pruebas poder conocer un aproximado del número de elementos necesarios para llevar el experimento a un éxito.

XXII. Manim

A. ¿Qué es manim?

Manim es un motor explicativo de animación de video matemático. Utiliza principalmente Python para convertir muchas fórmulas y teoremas matemáticos en efectos visuales.

B. Aplicaciones de manim al proyecto:

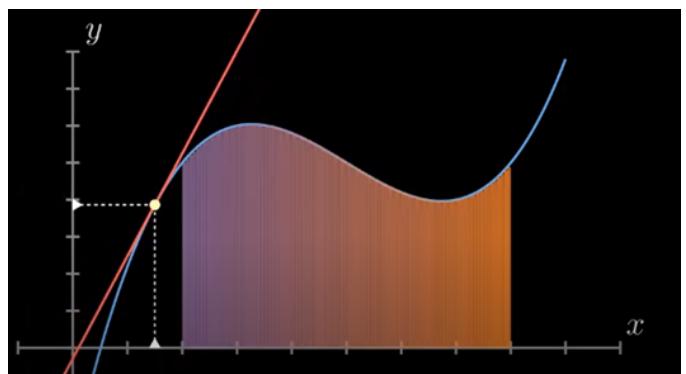
Esta herramienta tenía como propósito el crear animaciones para el video de los métodos matemáticos usados, hubo varias ideas de las opciones por las que se le podría haber dado uso, una de ellas fue el hacer animaciones de los caracteres matemáticos que serán mencionados en el video:

$$x^2 + \frac{b}{a}x + \left(\frac{b}{2a}\right)^2 = -\frac{c}{a} + \left(\frac{b}{2a}\right)^2$$

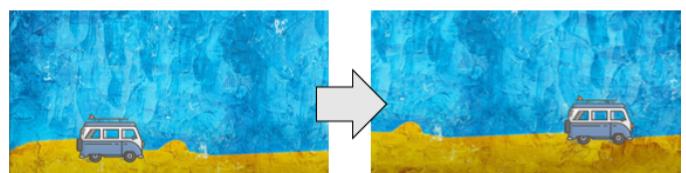
Cabe mencionar que manim no es un programa muy complicado de usar, pues basta con tener el mínimo conocimiento para manejar el lenguaje Python, ya que es con este lenguaje de programación con el que se hace uso de manim.

Otro uso era el de hacer las demostraciones gráficas de los resultados, o las conclusiones sacadas día tras día, incluso aprender esto podría servir para hacer

cualquier tipo de animación si se tiene el tiempo y la imaginación:



Estas son algunas de las cosas que se implementarían para explicar datos numéricos, pero como mencionado anteriormente manim es una herramienta de animación, por lo que a su vez se podrían crear animaciones en base a coordenadas y otras funciones de código nada difíciles de usar, esto a través de la importación de imágenes, puedes poner fondos y movimiento a objetos, solo hace falta tener imaginación:



Mejorando un poco la visualización del video y facilitando la explicación de los métodos, manim pudo ver sido una buena opción para este proyecto.

XXIII. Salida económica de la base de datos

El hosting es mucho más importante de lo que parece, ya que es el sitio en donde se alojarán todo el dominio y contenido de la página web.

Además, este almacenamiento afecta directamente a diversos factores que evitarán tener una pérdida de usuarios significativa.

Y, es clave tener en cuenta que su importancia radica en que:

- Un hosting debe tener calidad para garantizar los recursos, el rendimiento web y una velocidad de carga decente. Como recomendación, si es posible, se esté alojado en el mismo país que el sitio web.
- El almacenamiento web evita códigos de error, de esta forma se mejora el posicionamiento en los buscadores y, por ende, la estrategia SEO.
- El hosting permite obtener, de manera rápida, soluciones para resolver incidencias que surjan en algún momento dado.
- Un buen hosting contiene las medidas de seguridad adecuadas, evitando posibles ataques informáticos.

Entre los tipos de hosting más usados se encuentran:

A. Hosting Compartido

En este hosting (shared hosting), se alojan varios sitios web en un mismo servidor. Así, todos los clientes del proveedor comparten los recursos de dicho servidor, como el CPU, procesador, memoria RAM, ancho de banda, dirección IP y/o transferencia mensual de datos.

Entre las ventajas de este hosting suelen ser los más económicos en el mercado y es fácil de manejar e instalar.

Por otro lado, las desventajas de este tipo de hosting es que si hay un problema en alguno de los proyectos alojados, esto puede afectar al resto; existe una menor flexibilidad y al compartir los recursos, la velocidad y disponibilidad del sitio web puede verse perjudicados.

B. Hosting Privado Virtual (VPS)

Un VPS, es un servicio de almacenamiento web caracterizado por ser un servidor físico fraccionado virtualmente de forma independiente, es decir, cada proyecto trabaja con su propio sistema operativo y no comparte recursos con otros.

Este tipo de hosting es un servicio superior a los hosting compartidos, ya que en este caso el rendimiento de las páginas web no se ve afectado.

La razón es que los recursos como la memoria RAM, el procesador, el ancho de banda o la capacidad de transferencia de datos no son compartidos entre los múltiples usuarios. Sin embargo, no llega a ser un hosting dedicado.

Entre sus ventajas, posee mayor flexibilidad y acceso que el hosting compartido; si alguno de los proyectos tiene un problema, no afectará al resto; es una alternativa más económica que el hosting dedicado.

Pero, sólo dispone de una parte de los recursos totales de un servidor; es más costoso que el hosting compartido; y normalmente son más difíciles de configurar y usar que el hosting compartido.

Es útil en proyectos que tengan requerimientos y configuraciones específicas, y que necesitan mayores recursos del servidor, pero no pueden permitirse contratar un servidor dedicado.

C. Servidor Dedicado.

Es aquel cuyo uso exclusivo es de un solo cliente, por lo que el servidor no se comparte con otros proyectos.

Además, ofrece acceso completo al hardware y al software de dicho servidor, por lo que permite adaptar sus capacidades a las necesidades específicas de cada usuario.

Entre sus ventajas el rendimiento no se verá afectado por el tráfico de datos; permite que el sitio web pueda funcionar de manera óptima; garantiza mayor flexibilidad, velocidad y acceso; tiene mayores recursos y posee una mayor seguridad.

Por otro lado, puede resultar bastante caro y es complicado de configurar y mantener.

Este es el más recomendable para proyectos de gran envergadura que necesiten disponer de un servidor entero para alojar su proyecto.

D. Hosting en la nube (Cloud Hosting).

Este tipo de hosting distribuye los proyectos en varios servidores interconectados en una nube por lo que no estará alojado en un solo servidor.

Al tener esta característica, si uno de ellos fallara, el resto compensa esa pérdida de recursos.

Además, el hosting en la nube puede ser a su vez:

- Hosting Compartido
- Hosting Virtual (VPS)
- o, incluso, hosting dedicado.

Entre sus ventajas es más eficiente que el hosting dedicado; este servicio se adapta a las necesidades del proyecto en tiempo real; posee la mejor disponibilidad; tiene gran capacidad de adaptación y al contar con el respaldo de varios servidores, ofrece una mayor seguridad.

Pero, para su gestión requiere conocimientos avanzados; su precio de entrada es elevado; debido a que posee recursos variables, los precios finales también lo son y transferir datos personales entre países puede tener implicaciones legales.

XXIV. Conclusión

En conclusión, la totalidad de recursos que necesitamos para construir nuestro aporte al proyecto es un conjunto de tecnologías desarrolladas y establecidas sobre conceptos técnicos muy complejos, tan solo los servidores en los que se aloja nuestra página web es lo suficientemente avanzado para interpretar distintos lenguajes de programación y el cómo hace accesible esta información para cualquier persona con acceso a internet alrededor del mundo ya es algo impresionante y tiene su grado de dificultad, por lo

que pudimos ver de una forma viva el cómo se crean estos puntos de interconexión mundial y el gran potencial que ofrecen y ponen en manos de los conocedores de su lenguaje y funcionamiento, ademas de ver lo importante que es la comunicación constante de personas en el funcionamiento eficiente de cualquier trabajo que se quiera llevar a cabo; al crear este centro de encuentro de nuestro equipo nos dimos cuenta de la importancia que estos sitios tienen y han tenido para las personas que trabajan y navegan por esta enorme red, el mundo cambió mucho desde que personas de todo el mundo fueron capaces de comunicarse en segundos y nuestro sitio está pensado para ser esto mismo, un lugar donde podamos dar a conocer nuestras ideas y su progreso y al mismo timepo un sitio donde todods los miembros de nuestro equipo puedan encontrar nuestros datos.

Así que después de trabajar con herramientas de creación de páginas web, de estilos de páginas web y de bases de datos podemos decir lo extendido que está este campo y lo importante que son los espacios que se crean con estas tecnologías, así como los datos que se alojan en ellos.

En cuanto al proyecto del asfalto consideramos que aun con la infinidad de maneras en las que la ingeniería en Computación se pudo acoplar no logramos cumplir con los mínimos dados, esto por una mala organización en todos los niveles de equipo, aun así considerando que el proyecto se puede considerar como fallido creo que una parte de los integrantes logró adaptarse y sacar adelante nuestro aporte aprendiendo cosas nuevas más avanzadas que nuestro nivel actual en el proceso.

XXV. Glosarios

- [1] Sistemática: Sistema o método con que se clasifica algo.
- [2] Relacional: Perteneciente o relativo a la relación o correspondencia entre cosas.
- [3] Registros: Es un tipo de conjunto de datos almacenados en un sistema.

- [4] Administrador: Es un componente vital del entorno de TI para cualquier organización que dependa de uno o más sistemas de administración de base de datos.
- [5] Intranets: Red informática interna de una empresa u organismo, basada en los estándares de internet, en la que las computadoras están conectadas a uno o varios servidores web.
- [6] Internet: Es una red que conecta a otras redes y dispositivos para compartir información.
- [7] Interfaz: Conjunto de elementos de la pantalla que permiten al usuario realizar acciones sobre el sitio web que está visitando.
- [8] Servidor. Programa informático que proporciona servicios a otros programas informáticos (y sus usuarios).
- [9] Fronted: Es la interfaz que los visitantes pueden ver y usar.
- [10] Backend: Es la parte trasera de cualquier página web.
- [11] PHP: Es un lenguaje de programación de propósito general que se ejecuta en el lado del servidor.
- [12] Servidor: Es aquel que almacena y sirve ficheros a equipos de una red.
- [13] Página web: Es un medio digital para promover productos, servicios o información de interés, se consultan a través de cualquier dispositivo.
- [14] HTML: Es un lenguaje de marcado que nos permite indicar la estructura de nuestro documento mediante etiquetas.
- [15] CSS: Es un lenguaje que se utiliza para estilizar los elementos de un lenguaje de marcado como HTML.
- [16] Vínculos: Son elementos de las páginas que permite que el navegador
- [17] Links: Son enlaces que te pueden guiar a una página interna de un blog o a una externa.
- [18] Datos experimentales: Son datos producidos por una medición, método de ensayo, diseño experimental o diseño cuasi-experimental.
- [19] Dispositivo: Es un dispositivo de computación portátil que generalmente incluye una pantalla y un método de entrada.
- [20] Navegador: Programa que permite navegar por internet u otra red informática de comunicaciones.
- [21] Software: Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.
- [22] Hardware: Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.
- [23] FETCH: Es el nombre de una nueva API para javascript con la cual podemos realizar peticiones HTTP asíncronas utilizando promesas y de forma que el código sea un poco más sencillo.
- [24] Query: Concepto que un usuario escribe en un buscador.
- [25] Javascript: Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript.
- [26] login: Es un nombre dado al momento de autenticación al ingresar a un servicio o sistema.
- [27] CRUD: Está estrechamente vinculado a la gestión de datos digitales.
- [28] HOST: Es un ordenador que funciona como el punto de inicio y final de las transferencias de datos.
- [29] Red: Hace referencia a un conjunto de sistemas informáticos independientes conectados entre sí.
- [30] Algoritmo: Método que describe cómo se resuelve un problema en término de las acciones que se ejecutan y especifica el orden en que se ejecutan estas acciones.
- [31] Análisis: Proceso de identificación, modelado y descripción de lo que hace un sistema y de cómo trabaja.
- [32] Array: Objeto que almacena una secuencia indexada de los mismos tipos de datos.
- [33] Compilador: Programa de software que realiza un proceso de compilación (traducción del lenguaje fuente a lenguaje máquina) de un programa escrito en un lenguaje de programación de alto nivel.
- [34] Contenedor: Clase que implementa una estructura de datos que contiene una colección de objetos.
- [35] Declaración: Define las variables, métodos y clases en un programa.
- [36] Depuración: Proceso de encontrar, fijar y eliminar errores en un programa.
- [37] Función: Construcción matemática a la que se pueden aplicar valores y que devuelve un resultado.

- [38] Parámetro: Se especifican en una declaración de un método en una llamada a un método.
- [39] Anular: Donde un método de una subclase redefine y especializa un método del mismo tipo heredado de una superclase.
- [40] Programa: Conjunto de instrucciones (o sentencias) que describen alguna aplicación o actividad ejecutada en una computadora.
- [41] Lenguaje de programación: notación utilizada por los programadores para escribir programas.
- [42] WAMP: Son las siglas de Windows-Apache-MySQL-PHP, este es un software utilizado para páginas web, es parecido a Wordpress.
- [43] Wordpress: WordPress es un sistema de gestión de contenidos web y que sirve para publicar contenido en la web de forma sencilla.
- [44] localhost: Es un nombre reservado que tienen todas las computadoras, routers o dispositivos independientemente de que dispongan o no de una tarjeta de red ethernet.
- [45] Ethernet: Es un estándar de redes de área local para computadores, por sus siglas en español Acceso Múltiple con Escucha de Portadora y Detección de Colisiones (CSMA/CD).
- [46] Plugins: Es un complemento de software que ayuda a que un programa haga algo que normalmente no haría por sí solo.
- [47] Apache: Es un programa especialmente diseñado para transferir datos de hipertexto, es decir, páginas web con todos sus elementos (textos, widgets, banners, etc).estos servidores web utilizan el protocolo http.
- [48] Web banner: Es un formato publicitario en internet. Es un tipo de anuncio que generalmente es empleado en la publicidad llamada display, piezas publicitarias de contenido gráfico que se encuentran localizadas en diferentes páginas web.
- [49] Widgets: Son herramientas que utilizamos diariamente y que nos ofrece una mayor facilidad para interactuar con todas las aplicaciones que descargamos o tenemos integradas en nuestro teléfono móvil.
- [50] HTTPS: Es la versión segura del protocolo HTTP, utiliza un cifrado basado en Secure Socket Layers (SSL) para crear un canal seguro entre servidor y cliente.
- [51] Caché: Es una capa de almacenamiento de datos de alta velocidad que almacena un subconjunto de datos, normalmente transitorios, de modo que las solicitudes futuras de dichos datos se atienden con mayor rapidez que si se debe acceder a los datos desde la ubicación de almacenamiento principal.
- [52] URL: Es el mecanismo usado por los navegadores para obtener cualquier recurso publicando en la web.
- [53] archivo .htaccess: Este archivo funciona para el control de acceso al sitio web, redirige subdominios a subcarpetas, redirecciona a una conexión segura HTTPS.
- [54] Subdominio: Su función principal es separar las diferentes páginas de un sitio web para organizar los contenidos de forma más efectiva.
- [55] Subcarpeta: Son carpetas de archivo pero por su tamaño o por su función que cumplen se colocan dentro de otro tipo de carpeta o archivador.
- [56] IP (Protocolo de internet): Son los identificadores que permiten el envío de información entre dispositivos en una red.
- [57] Google Chrome: Es un navegador web, por lo que como su propio nombre indica sirve para poder visitar cualquier sitio web que se encuentre en internet.
- [58] frameworks: Es una estructura real o conceptual destinada a servir de soporte o guía para la construcción de algo que expande la estructura en algo útil.
- [59] Fireworks: Es un software de edición de mapas de bits y de gráficos vectoriales que forma parte de Adobe Creative Suite, el cual está integrado por diferentes aplicaciones y herramientas focalizadas principalmente en la publicación web, impresión y postproducción de video.
- [60] Visual Studio Code (VS Code): Es un entorno de desarrollo integrado para Windows y macOS.
- [61] Windows: Es un sistema operativo desarrollado por la compañía de software Microsoft Corporation, que cuenta con una interfaz gráfica de usuario basada en el prototipo de Windows.

- [62] Hover: Es el estilo de un enlace que tiene el ratón encima, pero sin estar seleccionado.
- [63] Link CSS: Estilo de un enlace que no ha sido explorado por el usuario.
- [64] Visited CSS: Estilo de un enlace que ha sido visitado:
- [65] Active CSS: Estilo de un enlace seleccionado, mientras está siendo seleccionado.
- [66] 000webhost: Es un espacio en un servidor, que está conectado a internet y permite almacenamiento y despacho de información (archivos.html, javascript, css, imágenes, videos, etc...) a cualquier visitante.
- [67] File: Es un comando de la familia de los sistemas operativos Unix, que permite detectar el tipo y formato de un archivo.
- [68] Export: Forzar a una aplicación a crear un documento que la misma aplicación no podrá editar luego, o solo podrá editar solo a través de la importación.
- [69] Importar: Se utiliza para incorporar a una aplicación objetos, documentos u otro tipo de archivos que fueron creados en otra aplicación, o que pertenecen a versiones antiguas de la misma aplicación.
- [70] Password: Es una forma de autenticación que utiliza información secreta para controlar el acceso hacia algún recurso.
- [71] Llave foránea: En bases de datos se le llama una llave foránea a un campo en una tabla que hace referencia a un campo principal en otra tabla diferente
- [72] SEO (Search Engine Optimization): Se trata del conjunto de estrategias y técnicas de optimización que se hacen en una página web para que aparezca originalmente en los buscadores de Internet como Google, Yahoo o Youtube.
- [3] Rouse, M. (s. f.). ¿Qué es servidor? TechTarget. Recuperado 11 de mayo de 2022, de <https://www.computerweekly.com/es/definicion/Servidor#:~:text=Un%20servidor%20es%20un%20programa,con%20frecuencia%20como%20un%20servidor>.
- [4] HostGator México. (2021). ¿Qué es Wamp? Aprende cómo utilizarlo. 26 Mayo, de HostGator Sitio web: <https://www.hostgator.mx/blog/que-es-wamp/>
- [5] A, M. (2022, 4 marzo). SQL - todo lo que necesitas saber sobre el lenguaje de programación de bases de datos. Formación en ciencia de datos | DataScientest.com. Recuperado 23 de mayo de 2022, de <https://datascientest.com/es/sql-todo-lo-que-necesitas-saber-sobre-el-lenguaje-de-programacion-de-bases-de-datos>
- [6] de DocuSign, C. (2021, 19 noviembre). 10 tipos de servidores que debes conocer. DocuSign. Recuperado 23 de mayo de 2022, de <https://www.docusign.mx/blog/tipos-de-servidor-es>
- [7] de Souza, I. (2021, 12 febrero). Aprende sobre los tipos de hosting más importantes de la actualidad y sus funciones. Rock Content - ES. Recuperado 23 de mayo de 2022, de <https://rockcontent.com/es/blog/tipos-de-hosting/>

XXVI. Fuentes

- [1] Valdés, D. P., & Valdés, D. P. (2015, 2 marzo). ¿Qué son las bases de datos? Maestros del Web. Recuperado 11 de mayo de 2022, de <https://www.maestrosdelweb.com/que-son-las-bases-de-datos/>
- [2] Editorial Etecé. (2021). Página web. 11 de Mayo de 2022, de Enciclopedia Concepto Sitio web: <https://concepto.de/pagina-web/>