**TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO**



**ALUMNO:** AMAYA DELGADO BRYAN

**SEMESTRE:** NOVENO **GRUPO:** 4953 **CICLO ESCOLAR:** 2022-2

**CARRERA:** INGENIERIA EN SISTEMAS COMPUTACIONALES

**ASIGNATURA:** DESARROLLO DE APLICACIONES PARA EL COMERCIO ELECTRÓNICO

**PROFESOR:** IVAN AZAMAR PALMA

**TITULO:** PRACTÍCAS DE LABORATORIO.

**TEMA:** 3

**FECHA DE ENTREGA:** 30 DE NOVIEMBRE DEL 2022

**LIGA DE ACCESO AL REPOSITORIO:**

<https://github.com/Bryan-Amaya-Delgado/Practicas-Unidad-3-4953.git>

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

**Ilustración 1** Aplicaciones Web

**ÍNDICE**

[MARCO TEÓRICO 3](#_Toc120752219)

[Definición de librería. 3](#_Toc120752220)

[Ventajas De Un Librería. 3](#_Toc120752221)

[Desventajas De Las Librerias. 4](#_Toc120752222)

[¿Qué es una librería? 4](#_Toc120752223)

[Ejemplos de Librerias. 5](#_Toc120752224)

[Tipos De Librerías En Programación 5](#_Toc120752225)

[DESARROLLO-PRACTICO 6](#_Toc120752226)

[Practica 2 – Aplicaciones Para El Lado Del Servidor (Api`S Dependencias Y Librerias) 6](#_Toc120752227)

[Practica 3 – Aplicaciones Para El Lado Del Servidor (Conexión BD y Creando un Servidor) 7](#_Toc120752228)

[Practica 4 – Aplicaciones Para El Lado Del Servidor. 8](#_Toc120752229)

[Practica 5 – APLICACIONES PARA EL LADO DEL SERVIDOR 9](#_Toc120752230)

[CONCLUSIÓNES 12](#_Toc120752231)

[REFERENCIAS BIBLIOGRÁFICAS 13](#_Toc120752232)

**Tabla de ilustraciones**

[**Ilustración 1** Aplicaciones Web 1](file:///C:\Users\bryan\Desktop\Amaya_Delgado_Bryan_P3.docx#_Toc120748733)

[**Ilustración 2**. Diferentes Librerias Y Frameworks. 3](file:///C:\Users\bryan\Desktop\Amaya_Delgado_Bryan_P3.docx#_Toc120748734)

# **MARCO TEÓRICO**

Logotipo, nombre de la empresa

Descripción generada automáticamente**Definición de librería.**

**Ilustración 2**. Diferentes Librerias Y Frameworks.

Una Librería es básicamente un componente de software que expone funcionalidades, pero no son ejecutables. En líneas generales, estas funcionalidades son usadas por los programas (ejecutables) que tienen acceso a esa biblioteca.

Pueden ser estáticas o dinámicas, dependiendo de si están compiladas dentro o fuera del ejecutable. En este último caso, las verás como ficheros dentro del file system.

**Ventajas De Un Librería.**

Entre las ventajas de utilizar una librería para el desarrollo de software distinguimos:

* **Estructura y organización del código predeterminada**. Las Librerias proporcionan tanto un esqueleto como una forma de trabajar. Por lo tanto, evitan tener que realizar un análisis sobre dónde situar los diferentes archivos de la aplicación (recursos, controladores, vistas, modelos, etc.).
* **Reutilización del código. Evitar duplicidad de código.** En el desarrollo de una aplicación existen ciertos apartados que suelen repetirse, como la conexión con la base de datos, validación de formularios, páginas de estilos, etc. Con la utilización de una Libreria ahorraremos tiempo en desarrollar funcionalidades que ya están cubiertas y podremos enfocarnos en el funcionamiento de la aplicación más que en cómo llevarla a cabo.
* **Agilidad y rapidez en el desarrollo.** Precisamente gracias a la reutilización de código mencionada anteriormente, conseguimos mayor rapidez en el desarrollo, ya que no perderemos tiempo en desarrollar funcionalidades nuevas.
* **Menor coste en el desarrollo.** El coste es un parámetro que está directamente relacionado con la rapidez y agilidad. Acabar antes un proyecto implica que la dedicación es menor y por lo tanto el coste del proyecto también disminuye.
* **Al igual que la rapidez en el desarrollo**, esta ventaja beneficia tanto al cliente como al desarrollador.
* **Buenas prácticas de desarrollo con el uso de patrones.** La mayoría de frameworks y Librerias están basados en patrones de diseños, que nos indican pautas sobre cómo solucionar un problema específico que ya ha ocurrido con anterioridad. El patrón de diseño más popular es MVC (Modelo-Vista-Controlador), que nos ayuda a separar la capa de datos de la lógica del negocio de la interfaz con el usuario.
* **Minimizar errores y mayor facilidad para solucionarlos.** Como la Librería ya incorpora código implementado por otros programadores, los posibles errores que este pueda tener siempre serán menores que al desarrollarlo desde cero. Además, en caso de que hubiera un error, lo más probable es que ya haya sido solucionado por la comunidad.
* **Facilidad a la hora de encontrar una librería o código que ya cubra funcionalidades de tu desarrollo.** Resulta más fácil encontrar herramientas (utilidades, librerías) adaptadas al framework que para un desarrollo propio.
* **Facilita la colaboración con otros desarrolladores.** Tanto si son compañeros de tu equipo como de GitHub, leer el código desarrollado por otra persona puede resultar complejo. Sin embargo, si ya sabes qué estructura va a seguir el código y cómo se organiza, resultará más fácil comprenderlo y poder aplicarle nuevos cambios. Lo que nos lleva a la siguiente ventaja.
* **Facilita el mantenimiento.** Si todos los miembros de un equipo trabajan de la misma forma, en el momento que haya que actualizar la aplicación o realizar algún evolutivo, tardaremos menos tiempo y el coste será menor.

**Desventajas De Las Librerias.**

* **Tiempo de aprendizaje.** Antes de empezar a utilizar un framework debemos familiarizarnos con él, con cómo se estructuran sus archivos, con la forma en la que se comunican los componentes, etc.

Por lo tanto, tendremos que invertir tiempo en superar la curva de aprendizaje para poder comenzar un nuevo desarrollo utilizando el framework.

* **Versiones inestables.** El hecho de que los frameworks sean tan populares provoca que estén en constante actualización para cumplir con las nuevas tecnologías y políticas de seguridad.

Por ello, si en el desarrollo surgen incompatibilidades con otras librerías o se detectan errores de seguridad, la elección de una versión muy reciente del framework podría ralentizarnos.

* **Menor rendimiento.** Los frameworks consumen, en general, más recursos que una aplicación creada desde cero y orientada al rendimiento. En aplicaciones muy exigentes, un framework puede resultar poco apropiado.
* **Código sin utilizar.** Si la aplicación es pequeña o no requiere mucha funcionalidad, probablemente estaremos desaprovechando

**¿Qué es una librería?**

Una librería es uno o varios archivos escritos en un lenguaje de programación determinado, que proporcionan diversas funcionalidades.

A diferencia de un framework, una librería no aporta la estructura sobre cómo realizar el desarrollo, sino que proporciona funcionalidades comunes, que ya han sido resueltas previamente por otros programadores y evitan la duplicidad de código. Además, reducen el tiempo de desarrollo y aumentan la calidad de este.

**Ejemplos de Librerias.**

**Estos son algunos de los frameworks más conocidos:**

* **Javascript:** Jquery, Mootols, Moment.Js, Anime.Js, Ramda, D3.Js, Chart.Js, Mathjs, Hammer.Js, Glimmer.Js, Etc.
* **C++:** Iostream, Cmath, Cstring, Ctime, Algorithm, Etc.
* **Python:** Matplotlib, Seaborn, Bokeh, Numpy, Scipy, Pandas, Numba, Gensim, Etc.

**Tipos De Librerías En Programación**

Hay, en términos generales, dos tipos de librerías en programación. Las librerías pueden ser caseras o externas, en función de si están creadas por ti o por otros programadores, pero la principal distinción es entre librerías estáticas y librerías dinámicas.

**1.- Librerías Estáticas**

Estas se graban en un programa como ejecutables. Sirven exclusivamente para esto; después, podemos borrarlas sin problemas, ya que el programa seguirá funcionando con la función necesaria.

**2.- Librerías Dinámicas**

Son distintas a las estáticas en tanto en cuanto no se copian en el programa al compilarlas. Las subrutinas son cargadas en tiempo de ejecución, en vez de enlazarse en tiempo de compilación.

**3.- Librerías Remotas:**

Esta es una biblioteca que utiliza ejecutables separados y por medio de un procedimiento remoto, llevado a cabo por otra computadora. Estas bibliotecas trabajan con un enfoque que maximiza la reutilización del sistema operativo; el código que soporta la biblioteca es el que le da a la aplicación el soporte y seguridad para otro programa.

# **DESARROLLO-PRACTICO**

En esta parte solamente se mostrarán los resultados de la práctica.

# **Practica 2 – Aplicaciones Para El Lado Del Servidor (Api`S Dependencias Y Librerias)**

**RESULTADOS**

1.- Instalación de las librerías.

Texto

Descripción generada automáticamente

2.- Conexión a La Base De Datos Local (MongoDB).

Texto

Descripción generada automáticamente

# **Practica 3 – Aplicaciones Para El Lado Del Servidor (Conexión BD y Creando un Servidor)**

**RESULTADOS**

1.- Impresión de las Respuestas en la consola.

Captura de pantalla de computadora

Descripción generada automáticamente

2.- Impresión de las Respuestas en el Navegador.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

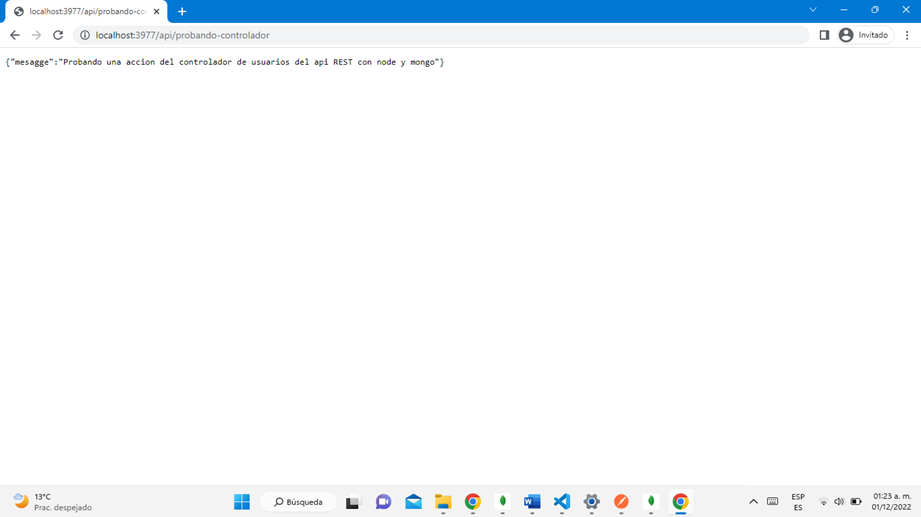
Captura de pantalla de computadora

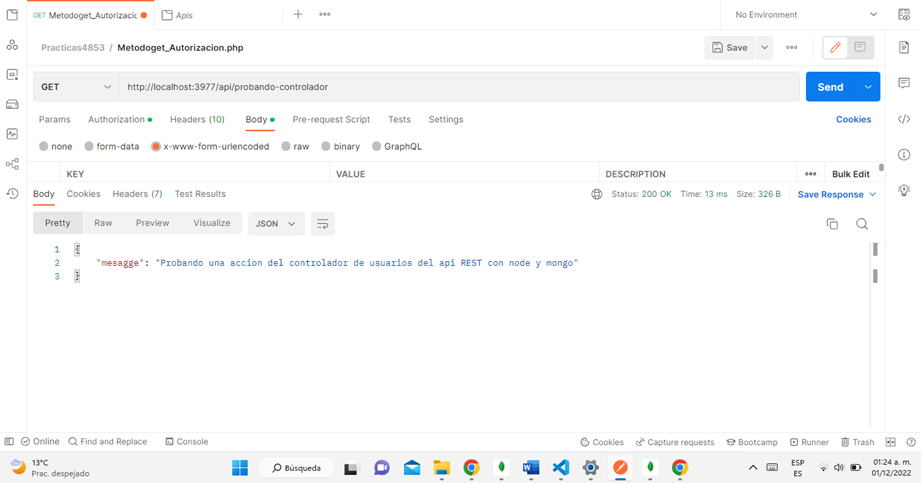
Descripción generada automáticamente

# **Practica 4 – Aplicaciones Para El Lado Del Servidor.**

**RESULTADOS.**

1.- Verificamos que el mensaje Se imprima Exitosamente.



****

2.- Envió de Datos.

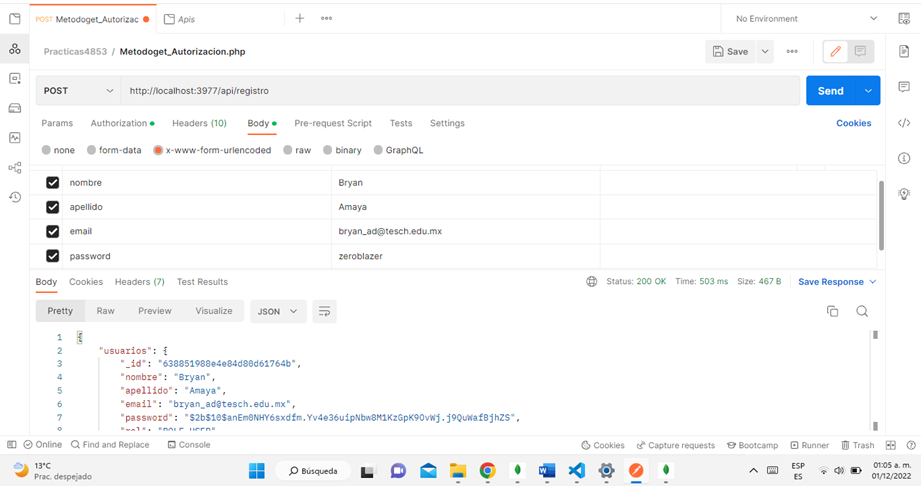
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

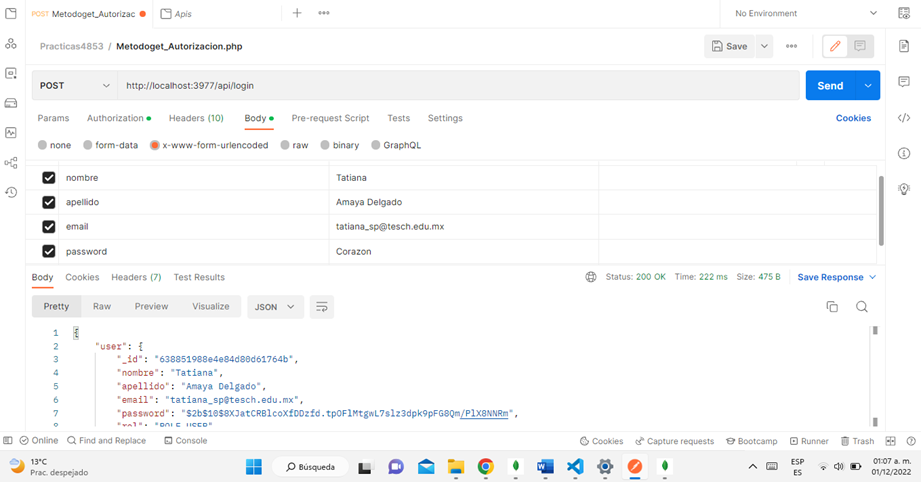
Descripción generada automáticamente

# **Practica 5 – APLICACIONES PARA EL LADO DEL SERVIDOR**

**RESULTADOS.**

1.- Enviamos los datos por el método post.





2.- Ejecución del Método Login.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

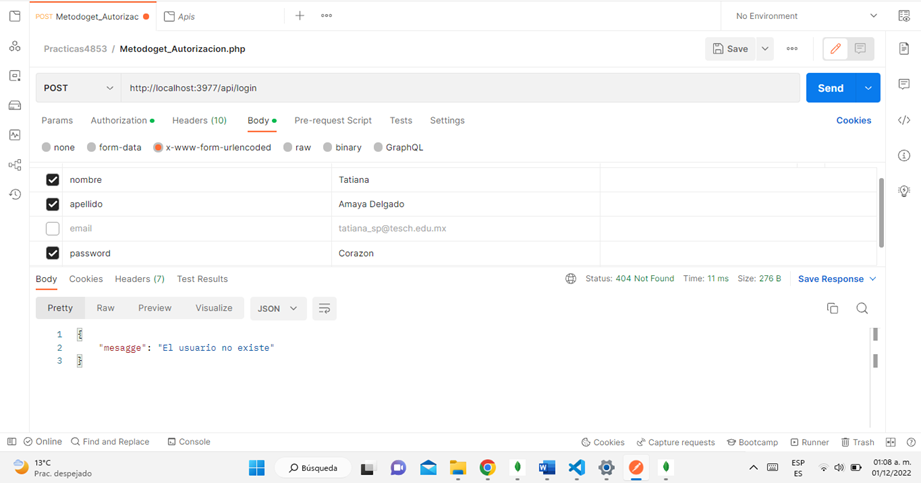
Descripción generada automáticamente

3.- Autenticación a través de consola.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

4.- Fallo de autenticación.



# **CONCLUSIÓNES**

En la siguiente actividad perteneciente a la Asignatura de Desarrollo de Aplicaciones para el Comercio Electrónico, fue un trabajo muy interesante debido al gran análisis que se realizó sobre sobre las actividades propuestas Del Lado del Servidor. En la cual se habla sobre cómo ha avanzado el procesamiento de datos a través de la programación de software y su implementación en los servicios de almacenamiento y generación de aplicaciones a través de la implantación de código puro enfocado al área de programación web.

A su vez cabe resaltar La web no es un ente individual como nos gusta pensar, se trata un conjunto de archivos variados entrelazados entre sí a los cuales podemos acceder por medio de una interfaz virtual que conecta dichos archivos con los usuarios.

Por otro punto debemos de saber que, desde la perspectiva del usuario, utilizando el navegador, la persona encuentra una bonita página web de uso intuitivo que le muestra la información ordenada y fácil de leer que más tarde podrá comentar o compartir en sus redes.

Por otra parte, desde la perspectiva del programador el esquema es totalmente distinto y sumamente complejo. Son ellos los que deben organizar a través de lenguajes de programación tales como HTML o CSS un conjunto de reglas de comportamiento que incluyen los más mínimos detalles para que el usuario pueda navegar con total comodidad sin conocer nada de lo que se esconde detrás.

Como pudimos notar en el trabajo que realizamos, nos da una detallada información sobre el uso de frameworks, los cuales nos permitirán darle más funcionalidades a los sistemas web que estemos desarrollando en un determinado momento.

Ya que gracias a esto nos da la pauta para que empecemos a crear y diseñar nuestras propias aplicaciones, los pasos que requerimos para hacerlo y darle una solución correcta al problema que queremos trasladar a la nube y después enlazarlo ala BD, y para hacerlo requerimos de las formas normales, las cuales hacen que nuestra base de datos quede con una buena lógica, ya que, sin eso, sería un caos y no resolvería nada.

Otro punto importante fue el saber que JS nos permite darle “movimiento” y dinamismo a la web. Es un lenguaje de programación del lado del cliente, es decir, se ejecuta en el navegador, no el servidor, lo que nos permite hacer acciones más rápidas.

Gracias a esto es importante decir que es vital para cualquier negocio, pues nos ayuda a encontrar información potencial para usarla para mejorar tu empresa. es un instrumento muy interesante para predecir el ritmo de consumo de los clientes. Si, con nuestro análisis, logramos establecer qué recurrencia tiene el cliente para comprar de nuevo nuestro producto o servicio, podremos ajustar la producción.

# **REFERENCIAS BIBLIOGRÁFICAS**

1. Alondra Lara. (22 de Noviembre de 2013). *Programador Clic*. Obtenido de https://programmerclick.com/article/29141531099/
2. Aritnetrics. (s.f.). *Developers Aritmetricks*. Obtenido de https://www.arimetrics.com/glosa
3. Desconocido . (25 de Mayo de 2014). *Thithink*. Obtenido de https://www.tithink.com/es/2018/08/29/framework-o-librerias-ventajas-y-desventajas/
4. Desconocido . (28 de Diciembre de 2020). *Girhub*. Obtenido de https://gist.github.com/BCasal/026e4c7f5c71418485c1
5. Desconocido. (08 de Marzo de 2014). *Courters - Que es un Yarg*. Obtenido de https://www.geeksforgeeks.org/node-js-yargs-module/
6. Desconocido. (03 de Septiembre de 2016). *Mind Web*. Obtenido de https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Statements/async\_function#:~:text=Una%20función%20async%20puede%20contener,y%20devuelve%20el%20valor%20resuelto.
7. Desconocido. (s.f.). *Persistencia Web* . Obtenido de https://agpele92.es.tl/2-.-4\_-PERSISTENCIA.htm