



Universidad Nacional Autónoma de México

Facultad de Estudios Superiores Cuautitlán

**Propuesta de mejora para el Sistema del Congreso de
Matemáticas de la Facultad de Estudios Superiores
Cuautitlán**

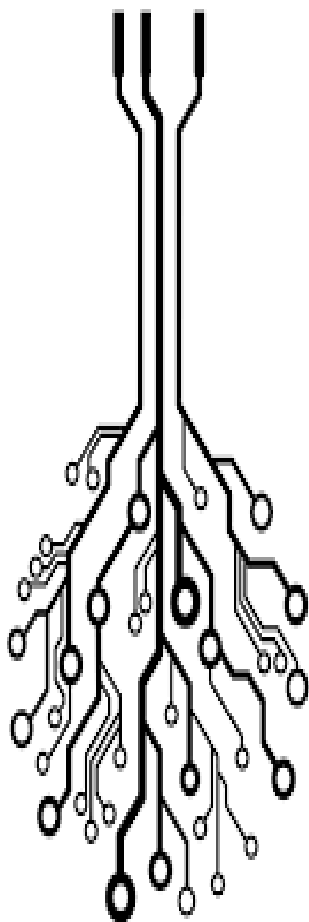
**Tesis
Que para obtener el título de
Licenciado en Informática**

Presenta:

Ernesto Torreblanca González

Eddy Arellanes Bastida

**Asesor:
L. I. Rosalba Nancy Rosas Fonseca
Coasesor:
M. I. Miguel de Nazareth Pineda Becerril**



FESC-INFORMATICA

Cuautitlán Izcalli, Estado de México 2021



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

ÍNDICE

1. MARCO METODOLÓGICO	3
1.1 RESUMEN	3
1.2 INTRODUCCIÓN	4
1.3 PLANTEAMIENTO DEL PROBLEMA	5
1.4 JUSTIFICACIÓN	6
1.5 OBJETIVO GENERAL	7
1.5.1 Objetivos específicos:	7
1.6 HIPÓTESIS	8
1.7 METODOLOGÍA	9
1.8 ANTECEDENTES	11
2. ANÁLISIS, DISEÑO Y PROGRAMACIÓN DE UN SISTEMA WEB.	12
2.1 Técnicas de Investigación	13
2.2 Análisis de Requisitos	19
2.3 Estudios de Factibilidad	23
2.4. Diseño de Interfaces	26
2.5. Diagramas de Caso de Uso	27
2.6. Diagrama de Flujo	29
2.7. Programación de un Sitio Web.	30
2.8. Implementación	44
2.9. Pruebas del Sitio Web	46
3. SEGURIDAD EN LAS PÁGINAS WEB	47
3.1 Ataques	48
3.2 Amenazas	54
3.3 Cifrado	58
4. BASES DE DATOS	61
4.1 Diseño	64
4.2 Diagrama Entidad- Relación	69
4.3 Seguridad en las Bases de Datos	71
5. CASO PRÁCTICO	71
5.1 Arquitectura de Software	72
5.2 Diseño	78
5.3 Base de Datos	87
5.4 Seguridad	93

5.5 Análisis de Resultados	97
6. CONCLUSIONES	103
7. RECOMENDACIONES	104
8. BIBLIOGRAFÍA	105

1. MARCO METODOLÓGICO

1.1 RESUMEN

El presente proyecto pretende contribuir con el desarrollo académico dentro de las Facultad de Estudios Superiores Cuautitlán, en el área de Matemáticas, enfocado al sistema del **Congreso Internacional sobre la Enseñanza y Aplicación de las Matemáticas**, que se realiza de forma anual.

A través de una propuesta de mejora al sistema, que propone implementar la parte de la interfaces de usuario, así como la distribución de contenido, aumentar el tiempo de sesión de los usuarios, mejor navegación por el sitio, implementando seguridad y reducir actividades automatizando todo proceso que se realiza en los congresos anuales para tener un mejor control de todas las actividades a realizar.

1.2 INTRODUCCIÓN

El presente trabajo es para generar una propuesta para un sistema que automatice el **Congreso Internacional sobre la Enseñanza y Aplicación de las Matemáticas**, buscando de esta manera agilizar las tareas y actividades que se llevan dentro de este.

La propuesta del sistema contaría con un control de acceso, el cual será mediante un login (usuarios y contraseñas), lo que se resumirá en un mejor control para los administradores, ya que estos podrán ver quiénes participarán en el congreso, proporcionando mayor seguridad, veracidad e integridad de los datos.

Este contará con una interfaz amigable e intuitiva para el usuario, para que la interacción usuario-interfaces sea agradable.

1.3 PLANTEAMIENTO DEL PROBLEMA

Elaborar una propuesta que mejore el control del sistema actual del **Congreso Internacional sobre la Enseñanza y Aplicación de las Matemáticas**, agregando valor a la Página Web como al control y diseño de las Bases de Datos.

1.4 JUSTIFICACIÓN

Se creará un sistema para apoyo a la logística del **Congreso Internacional sobre la Enseñanza y Aplicación de las Matemáticas**, que nos ayude a reducir el tiempo de la planificación del mismo. Pasando por filtros de calidad administrativa y así poder proporcionar un mejor contenido a los usuarios que asisten a estos eventos.

Los Administrativos tendrán un mejor control, ya que este sistema integrará las siguientes partes para tener un sistema concreto y eficaz:

- El sistema contará con permisos (Login) para poder acceder a la información correspondiente dependiendo el perfil que se tenga (permisos).
- El sistema mostrará una galería de imágenes de los lugares de las ponencias, talleres o carteles para una mejor referencia de los usuarios.
- Los usuarios que deseen participar en el congreso con ponencia, taller o cartel, podrán registrarse en este sistema y subir su archivo según sea el caso, e introducir una pequeña reseña.
- Gracias a este sistema se podrán emitir Diplomas, programas y gafetes
- Se generará un mejor control en cuanto a los horarios y los lugares donde se darán las ponencias, talleres y exposición de carteles, mejorando la logística de los eventos.

1.5 OBJETIVO GENERAL

Desarrollar una propuesta que nos ayude a sistematizar los procesos que se llevan a cabo en la organización del **Congreso Internacional sobre la Enseñanza y Aplicación de las Matemáticas**.

1.5.1 *Objetivos específicos:*

- Proponer el registro de usuarios en el sistema para llevar un mejor control y seguridad del mismo.

- Elaborar un esquema del Sistema web que sea intuitivo para el usuario así mismo amigable.

- Esquematizar las diferentes etapas del sistema web para verificar que sea funcional e integral en las diferentes actividades que se llevan a cabo en el **Congreso Internacional sobre la Enseñanza y Aplicación de las Matemáticas**.

1.6 HIPÓTESIS

Una vez elaborada la propuesta de mejora, se considera de utilidad para implementarse en el sistema actual, optimizando los procesos y reduciendo tareas que se llevan a cabo en el **Congreso Internacional sobre la Enseñanza y Aplicación de las Matemáticas**.

1.7 METODOLOGÍA

La metodología que se emplea en este proyecto será la de **SCRUM**, ya que es un proceso en el que se aplica el trabajo colaborativo en equipo, así de esta manera se obtiene el mejor resultado del proyecto.

Para esta metodología se emplea las entregas parciales y regulares del producto final, de esta manera se puede ir priorizando las tareas a entregar dependiendo el grado de importancia para el proyecto.

Las fases de la metodología que se emplearán en este trabajo serán las siguientes:

- **Elegir un responsable del proyecto:** es fundamental que tenga la visión clara de lo que se necesita, lo que se tendrá que hacer y de los riesgos posibles que se puedan presentar.
- **Elegir un equipo:** se tiene que tomar en cuenta que este equipo es el que desarrollara todo el trabajo, el equipo debe de tener las habilidades necesarias que requiera el proyecto para sacarlo a flote.
- **Elegir un Scrum Master:** Esta persona será el encargado de llevar al equipo de trabajo, ayudando a eliminar cualquier situación que lo frene.
- **Elaborar y priorizar una lista de objetivos:** es una lista donde se definirán todo aquello que se debe hacer para que el proyecto se vea realizado, el cual va asignando u orden de prioridades para poder realizar el trabajo en tiempo y forma.
- **Hacer una estimación afinada de la lista de objetivos pendientes:** se realiza con el fin de que se vea cualquier esfuerzo que llevará cada

integrante, con la intención de que la carga de trabajo se equitativa y sea más fácil hacer el trabajo.

- **Planificación Sprints:** son pequeñas metas que se tiene que fijar el equipo, el Scrum Master y el responsable del Proyecto, para que posteriormente se realicen reuniones para entregar cifras de cada meta propuesta, con el fin de ir mejorando con cada meta impuesta y lograr lo prometido.

- **Hacer que el trabajo sea visible:** lo recomendable es en un pizarrón, dividirlo en tres columnas “PENDIENTES, EN PROCESO Y HECHO”, a manera de que todos los involucrados tengan acceso a estos datos, una vez que se vayan completando los ítems, se deberá pasar los post-it a la siguiente columna.

- **Scrum diario:** se refiere a una reunión diaria donde se reunirá el Scrum master y el equipo para resolver duda y aclarar sugerencias, con el fin de reducir los obstáculos que ayuden al equipo a desarrollar las metas.

- **Revisión de Sprints:** esto se hace para que todos vean el avance de los Sprints ya fijados con anterioridad y así, rectificar los pendientes.

- **Retrospectiva del Sprint:** una vez que un Sprint se cumplió es de vital importancia sentarse para revisar qué es lo que se podría hacer mejor e implementarlo sin que afecte los tiempos de los demás Sprints.

- **Empieza de inmediato el siguiente ciclo de Sprints:** este proceso se lleva a cabo una vez que ya se tienen los obstáculos y las mejoras, se debe a empezar de nuevo los Sprints para que sea viable el proyecto.

1.8 ANTECEDENTES

Actualmente el departamento de Matemáticas cuenta con un sistema que no engloba todo lo que comprende el **Congreso Internacional sobre la Enseñanza y Aplicación de las Matemáticas**, lo que hace que se dificulte para los administrativos poder controlar mejor la parte logística, horarios, lugares y el contenido que se tendrá en el evento.

Se cuenta actualmente con un sistema de registro de eventos, el objetivo de la propuesta es integrar los diferentes procesos que se llevan por separado, para el correcto funcionamiento.

2. ANÁLISIS, DISEÑO Y PROGRAMACIÓN DE UN SISTEMA WEB.

Como sabemos los sistemas se desarrollan para diversos propósitos, según sean las necesidades, para ello se deben realizar un análisis, definir el diseño e identificar el funcionamiento que mejor se adecue.

Según el autor el punto de analizar es la primera fase para poder dividir un tema que es complejo, esto servirá para que se pueda segmentar en partes más pequeñas para que se obtenga mejor comprensión del tema en general, y así poder obtener mejor resultado de cada segmentación.



Figura 1. Análisis de un Sistema Web. Autoría Propia.

2.1 Técnicas de Investigación

La investigación según el autor Sampieri, R. (2010). es un proceso de conjuntos sistemáticos, empíricos y críticos que se aplican al estudio de nómeno.

A lo que podemos entender que la investigación es un conjunto de métodos para entender o estudiar mejor una necesidad, que al unir todos sus elementos se tienen como resultado uno o varias soluciones.

Para realizar un Análisis de un Sistema Web es necesario tener en cuenta las técnicas de investigación y obtener la mayor cantidad de datos posibles que ayuden al entendimiento para la toma de decisiones.



Figura 2. Investigación. www.pixabay.com

Citando al autor Behar, D. (2008), dice que, la investigación no tiene sentido sin la recolección de datos.

Esto quiere decir que es de suma importancia abordar las técnicas de investigación ya que con ellas se recopilan todos los datos necesarios, para que en conjunto nos arrojen información íntegra y segura, para que la toma de decisiones sea la correcta. Las técnicas de investigación más empleadas para desarrollar o implementar un sistema de información son las siguientes:

- **La Entrevista:** se tiene como objetivo recolectar datos e información para indagar. Se hace principalmente mediante un diálogo, a quien más sabe del tema, con el fin de que aporte datos que sean verídicos y de gran utilidad.
 - Se tiene que tomar en cuenta que la entrevista tiene un inconveniente, es que reduce y limita los alcances de la información, ya que a las personas que se entrevistan darán su punto de vista a través de toda su carga sugestiva de interés, prejuicio y estereotipos.

- Para que la entrevista pueda ser exitosa, se necesita de un entrevistador altamente calificado, que conozca las técnicas, el tema de lo que se está investigando para que el entrevistado se sienta con la libertad adecuada de contestar, así como las preguntas adecuadas que le ayuden a garantizar que cada dato recopilado será de suma importancia para la investigación.



Figura 3. Entrevista. Autoría Propia.

- **La Encuesta:** esta se enfoca en una porción de la población de dicho interés, dependiendo el propósito del proyecto, será equivalente al tamaño de la muestra.
 - La información que se obtiene, es mediante información estandarizada, con las mismas preguntas para cada uno de los individuos, la intención de esta prueba es poder medir los parámetros de la información, en qué escala se inclina más cierto sector.

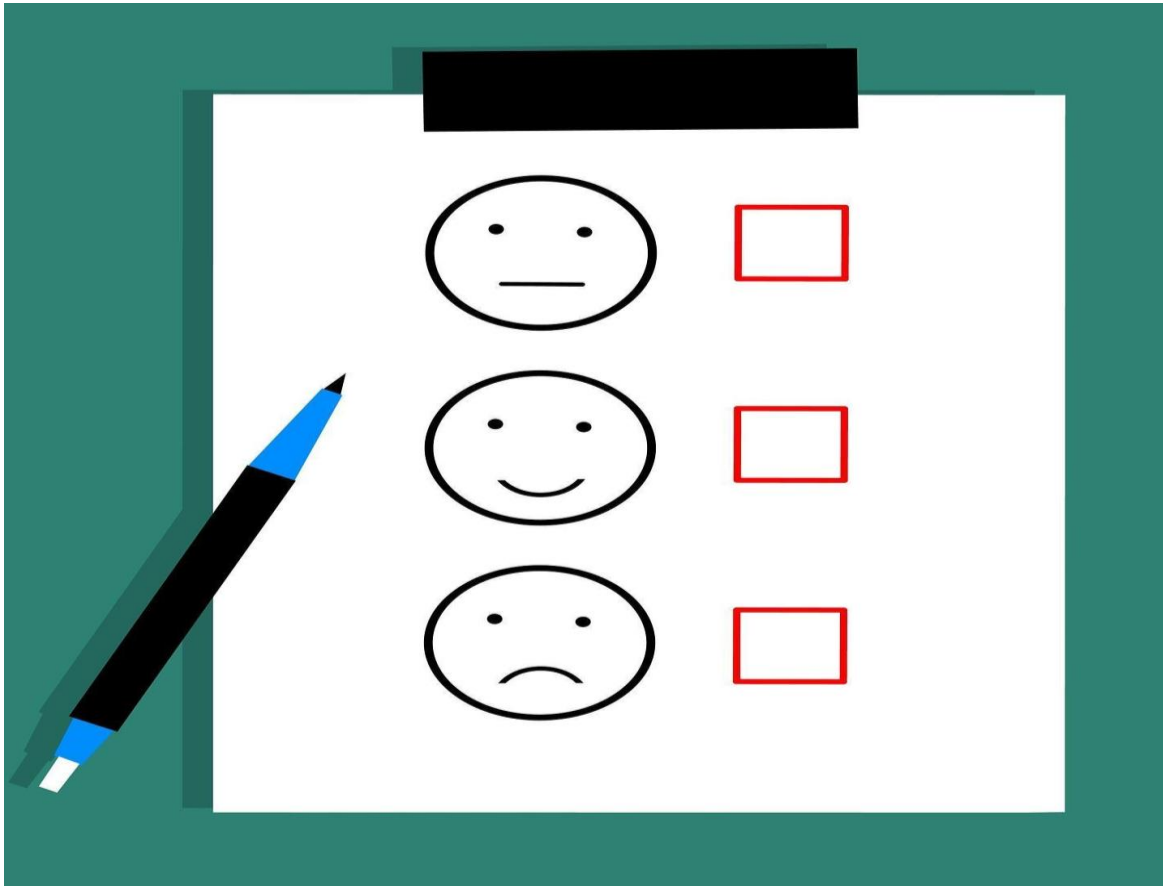


Figura 4. Encuesta. Autoría Propia.

- **El Cuestionario:** este es una serie de preguntas que se usan para medir una o más variables, las preguntas pueden ser variadas como los valores a medir.
 - Se requiere que se las preguntas sean cerradas para poder delimitar las respuestas y de esta manera poder tener mejor la medición de cada una de las preguntas.
 - Se recomienda que el cuestionario sea “*Cuestionario Restringido o Cerrado*”, ya que este solicita respuestas breves, específicas y delimitadas, que serán más fácil para el cálculo de las variables para las cuales se crea el cuestionario. Y se puede contestar con un *SÍ* o un *NO*, así como con varias alternativas para responder a las preguntas.

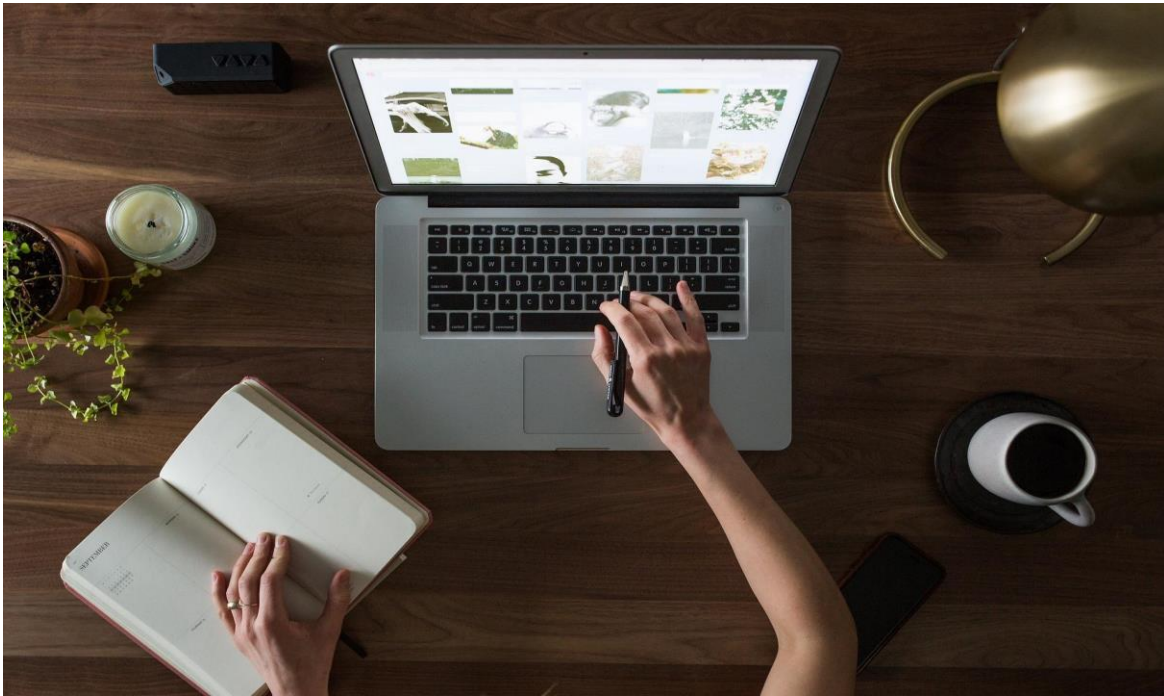


Figura 5. Cuestionario. www.pixabay.com

- **Diagrama de Flujo:** Esta es una representación gráfica de los pasos en proceso para determinar cómo funciona un proceso y se pueda reducir el tiempo para obtener un resultado, al igual que se puede observar cada proceso e identificar los posibles riesgos potenciales.



Figura 6. Diagrama de Flujo. Autoría Propia.

- **Procesamiento de La Información:** Esta es la etapa donde los datos se juntan para crear información que sea útil para la investigación, se toman todas las

muestras que se juntan, y podrá arrojar si está culminado, o le faltó algo y si es necesario reiniciar toda la investigación.

- Para que la Información se procese mejor es necesario cuadros y tablas para obtener matrices de datos con el objetivo de analizarlos e interpretarlos y poder sacar conclusiones.



Figura 7. Procesamiento de Información. Autoría Propia.

- **Conclusiones:** Es la culminación de la investigación, una vez que se ha cubierto cada uno de los aspectos de la investigación, que se analizó a detalles esos resultados e información, se puede llegar a tomar una decisión que obtenga buenos resultados.

Citando al autor Behar, D. (2008). la Conclusión es la parte culminante de todo trabajo, en donde se acentúan los frutos de la investigación. Ya sea que los resultados sean positivos o negativos, aquí se resaltan los resultados para respaldar lo obtenido.



Figura 8. Conclusión. www.pixabay.com.

2.2 Análisis de Requisitos

Es de vital importancia hacer un análisis detallado de los Requisitos que se utilizarán.

El autor Pressman, R. (2005). Menciona que el análisis de los requerimientos da como resultado la especificación de las características operativas del software, indica la interfaz de éste y otros elementos del sistema, y establece las restricciones que limitan al software. El análisis de los requerimientos permite al profesional (sin importar si se llama ingeniero de software, analista o modelista) construir sobre los requerimientos básicos establecidos durante las tareas de concepción, indagación y negociación, que son parte de la ingeniería de los requerimientos.

```
index.html
1 <!DOCTYPE html>
2 <html lang="es">
3
4 <head>
5
6 <meta charset="utf-8">
7 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
8 <meta name="description" content="">
9 <meta name="author" content="">
10
11 <title>Pequeño Negocio</title>
12
13 <!-- Bootstrap core CSS -->
14 <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
15
16 <!-- Custom fonts for this template -->
17 <link href="https://fonts.googleapis.com/css?family=Raleway:100,100i,200,200i,300,300i,400,400i,500,500i,600,600i,700,700i,800,800i,900,900i" rel="stylesheet">
18 <link href="https://fonts.googleapis.com/css?family=Lora:400,400i,700,700i" rel="stylesheet">
19
20 <!-- Custom styles for this template -->
21 <link href="css/business-casual.min.css" rel="stylesheet">
22
23 </head>
24
25 <body>
26
27 <h1 class="site-heading text-center text-white d-none d-lg-block">reco
28 <span class="site-heading-upper text-primary mb-3">E R K A I E</span>
29 <span class="site-heading-lower">pequeño negocio de cafe</span>
30 </h1>
31
32 <!-- Navigation -->
33 <nav class="navbar navbar-expand-lg navbar-dark py-lg-4" id="mainNav" >
34 <div class="container">
35 <a class="navbar-brand text-uppercase text-expanded font-weight-bold d-lg-none" href="#">Start Bootstrap</a>
36 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive"
37 aria-expanded="false" aria-label="Toggle navigation">
38 <span class="navbar-toggler-icon"></span>
39 </button>
40 <div class="collapse navbar-collapse" id="navbarResponsive">
41 <ul class="navbar-nav mx-auto">
42 <li class="nav-item active px-lg-4">
43 <a class="nav-link text-uppercase text-expanded" href="index.html">Home
44 <span class="sr-only">(current)</span>
45 </a>
46 </li>
47 </ul>
48 </div>
49 </div>
50 </nav>
```

Figura 9. Análisis y Requisitos. Autoría Propia.

La acción de modelar los requerimientos da como resultado uno o más de los siguientes tipos

de modelo:

- Modelos basados en el escenario de los requerimientos desde el punto de vista de distintos “actores” del sistema.



Figura 10. Creación de Base de Datos. www.pixabay.com.

- Modelos de datos, que ilustran el dominio de información del problema.

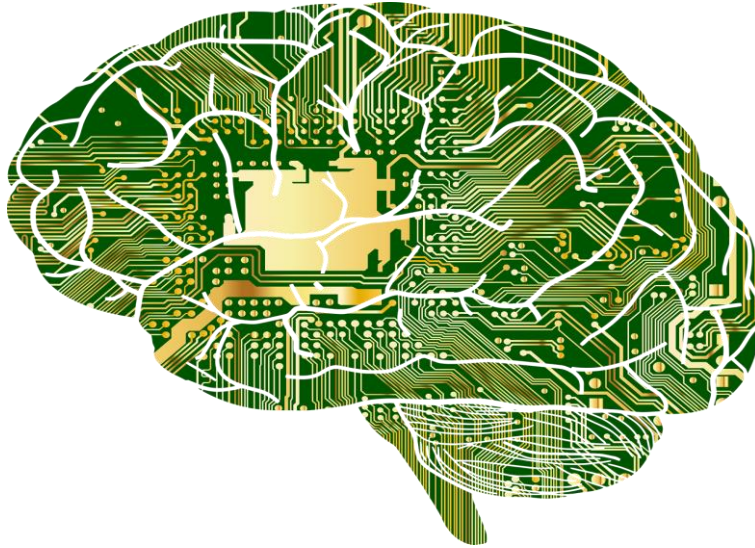


Figura 11. Modelado de Datos. www.pixabay.com

- Modelos orientados a clases, que representan clases orientadas a objetos (atributos y operaciones) y la manera en la que las clases colaboran para cumplir con los requerimientos del sistema.

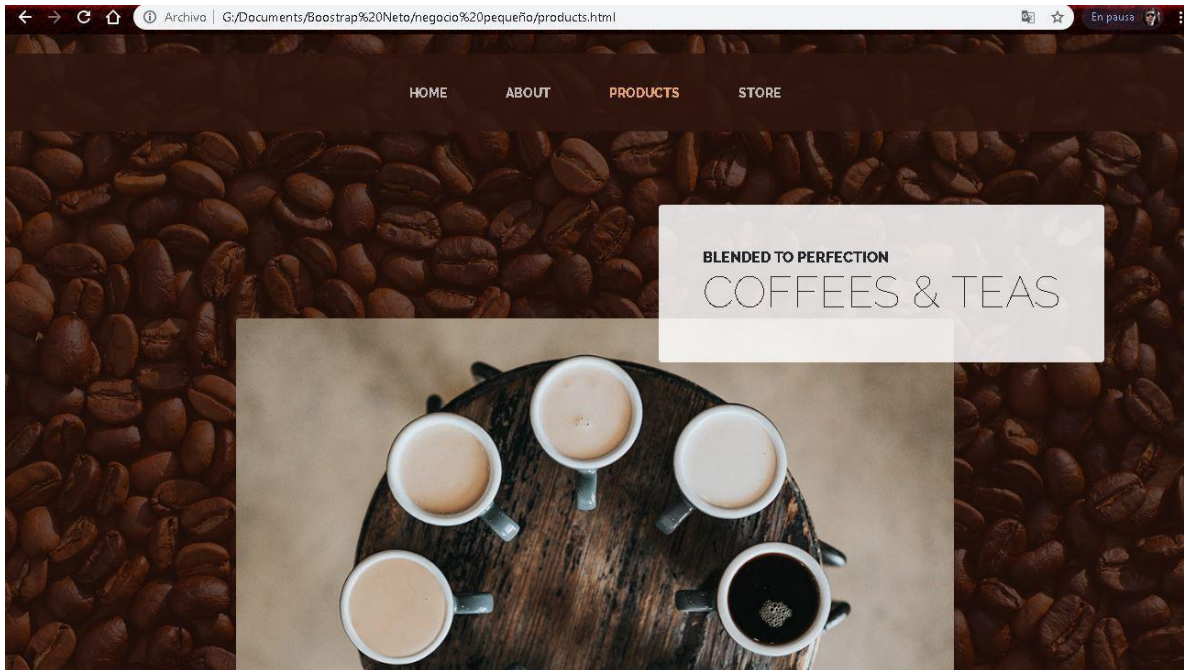


Figura 12. Modelado orientado a Clases. Autoría Propia.

- Modelos orientados al flujo, que representan los elementos funcionales del sistema y la manera cómo transforman los datos a medida que se avanza a través del sistema. (Como se Muestra en la Figura 13).

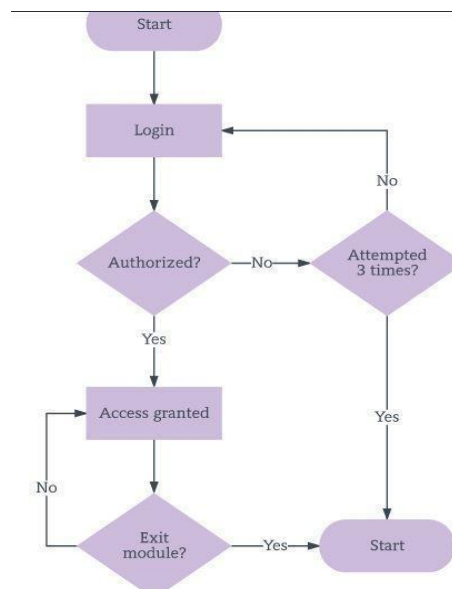


Figura 13. Modelado de Flujo. Autoría Propia.

Modelos de comportamiento, que ilustran el modo en el que se comparte el software como consecuencia de “eventos” externos.

Estos modelos dan al diseñador del software la información que se traduce en diseños de arquitectura, interfaz y componentes. Por último, el modelo de requerimientos (y la especificación de requerimientos de software) brinda al desarrollador y al cliente los medios para evaluar la calidad una vez construido el software.

Se centra en el modelado basado en escenarios, técnica que cada vez es más popular entre la comunidad de la ingeniería de software; el modelado basado en datos, más especializados, apropiado en particular cuando debe crearse una aplicación o bien manipular un espacio complejo de información; y el modelado orientado a clases, representación de las clases orientada a objetos y a las colaboraciones resultantes que permiten que funcione el sistema.

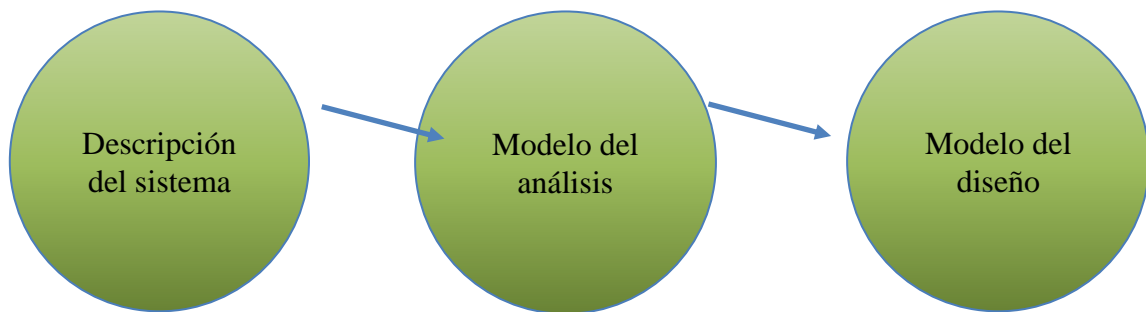


Figura 14. Modelado de Flujo. Autoría Propia.

2.3 Estudios de Factibilidad

El estudio de factibilidad es una herramienta que nos ayuda para la toma de decisiones conforme se va avanzando en las etapas de un proyecto.

El objetivo de este estudio es hacer un análisis de los recursos necesarios para poder implementar el sistema de manera efectiva, así como también evaluar si los usuarios poseen los conocimientos o habilidades necesarias para la utilización de este.

Al realizar este estudio debemos iniciar haciendo un inventario del software y hardware con el que se cuenta actualmente, así como registrar los procesos que se llevarán a cabo por medio de un sistema computacional. Evaluaremos tres puntos principales: factibilidad técnica, operativa y económica.

En este estudio se planea evaluar la cantidad de los usuarios del sistema, así como las herramientas disponibles para la implementación del mismo.



Figura 15. Estudio de Factibilidad. www.pixabay.com

Podemos concluir que el estudio nos ayudará a conocer, las herramientas disponibles, los activos, y los conocimientos necesarios para la utilización del sistema, evaluando si se requiere la compra de más recursos, si es óptima la inversión, si requiere de capacitación, etc.

A continuación, se mostrarán unos cuantos ejemplos del estudio de factibilidad:

Operativa

Pregunta	Si	No
¿Tiene establecidos los procesos que se van a utilizar?		•
¿Requerirá el uso de esta aplicación, la capacitación del usuario?	•	
¿Se requerirá mantenimiento?		•
¿Se beneficiará económicamente el cliente con el uso de esta aplicación?	•	

Técnica

Pregunta	Si	No
¿Se cuenta con los dispositivos adecuados?	•	
¿Tiene acceso a internet?	•	
¿Se puede desarrollar el sistema?	•	
¿Se beneficiará económicamente el cliente con el uso de esta aplicación?	•	

Económica

Pregunta	Si	No
¿Se requerirá de un capital excesivo para el sistema?		•
¿Supone pérdidas para el negocio el no implementar el sistema?		•
¿Se usará hosting privado?		•

2.4. Diseño de Interfaces

Un diseño de interfaces de usuario es una tarea que ha adquirido relevancia en el desarrollo de un sistema. La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso.

Los autores Kendall, E. & Kendall, J. (2005), en muchas ocasiones, menciona que la palabra interfaz se refiere en realidad a un concepto mucho más específico; la interfaz gráfica de usuario (GUI del inglés graphical user interface). La GUI es el entorno visual en el que se desarrolla la interacción entre la persona y el dispositivo, y puede ser el propio del sistema operativo o bien el particular de la aplicación que se está utilizando.

Es importante a la hora de desarrollar un sitio web, identificar qué interfaz es la que ofrece las mejores y más completas herramientas que puedan satisfacer las necesidades del sitio web, así como para quienes va enfocada la interface, para poder brindar una mayor comodidad a la hora de usarla.

Ejemplo:

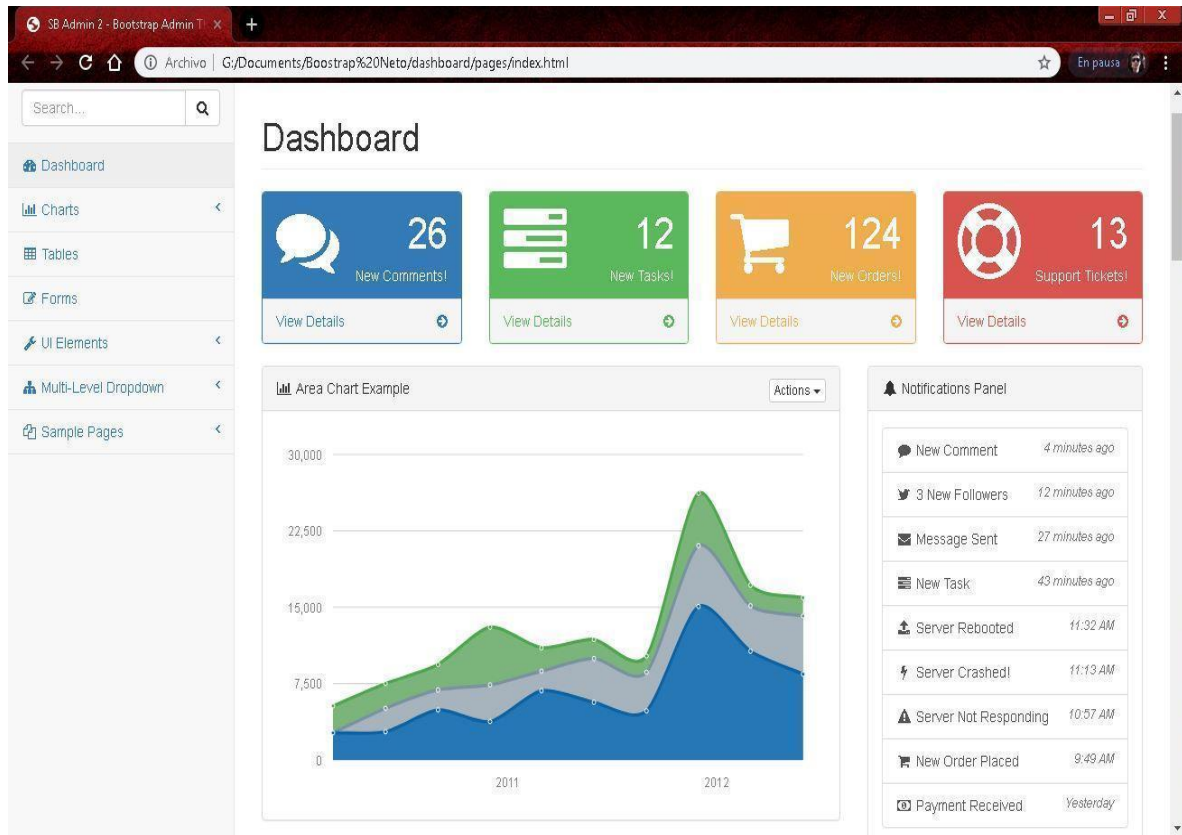


Figura 16. Interfaz de Negocios. Autoría Propia.

2.5. Diagramas de Caso de Uso

Lenguaje Unificado de Modelado “UML”, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Se lo puede definir como un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

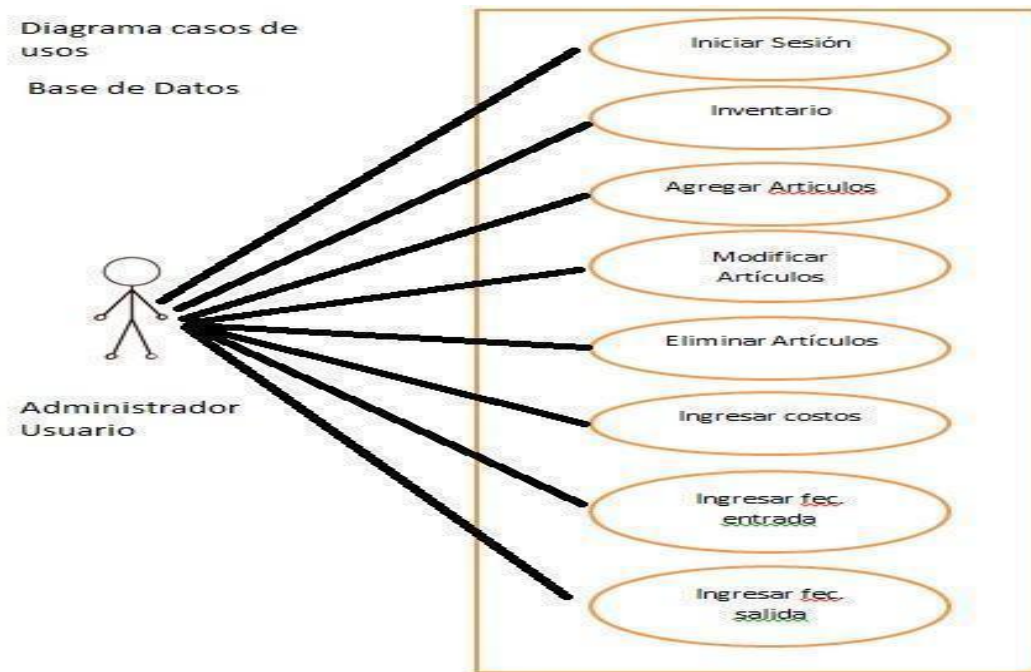
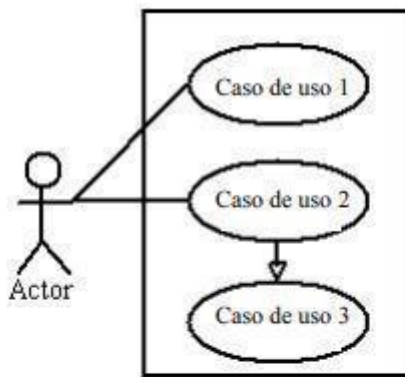


Figura 17. Diagrama UML. Autoría Propia.

UML ofrece un estándar para describir un “plano” del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

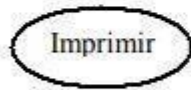
Pressman, R. (2005). Define que Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario.

Conocidos también como UML (Lenguaje de Modelo Unificado). Estas herramientas ayudan a definir una notación gráfica de quien usará el sistema o aplicación e identificar qué puede hacer en una interface.



Sistema

El rectángulo representa los límites del sistema que contiene los *casos de uso*. Los *actores* se ubican fuera de los límites del sistema.



Casos de Uso

Se representan con óvalos. La etiqueta en el óvalo indica la función del sistema.



Actores

Los *actores* son los usuarios de un sistema.

Figura 18. Caso de Uso. Autoría Propia.

2.6. Diagrama de Flujo

Es una representación gráfica de los procesos del proceso o algoritmo sobre el funcionamiento de un sistema, se representa mediante figuras geométricas, las cuales tiene un significado dependiendo del proceso a realizar.

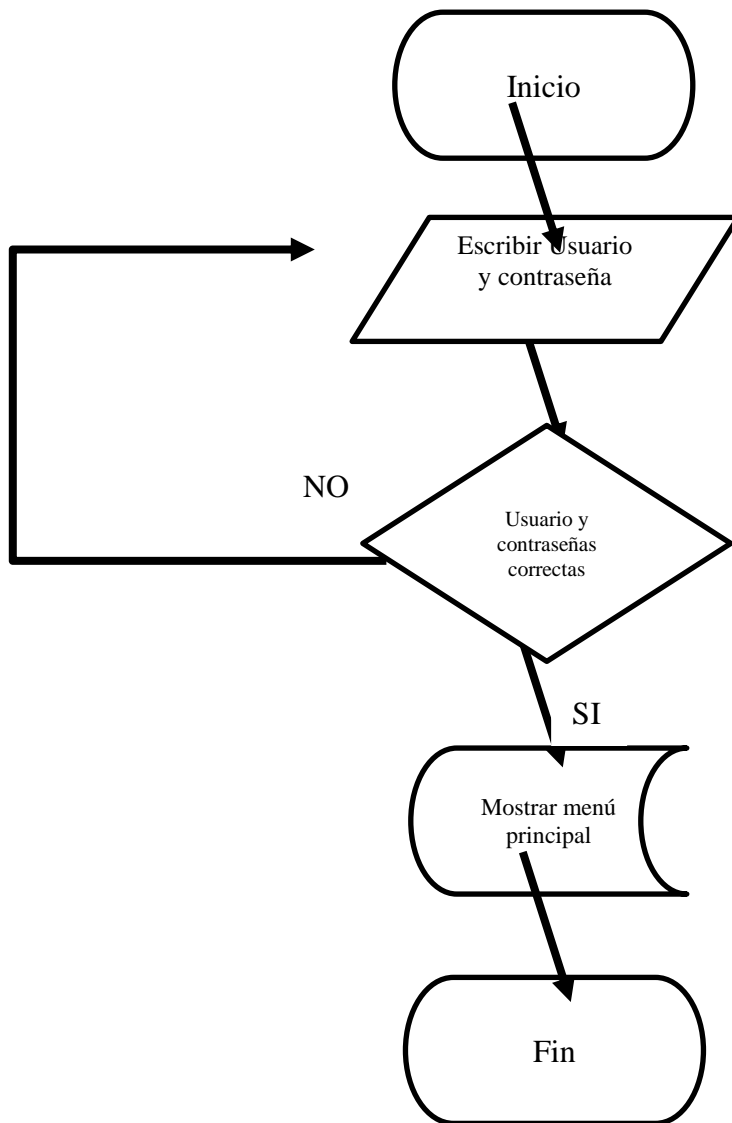


Figura 18. Diagrama de Flujo. Autoría Propia.

2.7. Programación de un Sitio Web.

El diseño de un sistema también se conoce como DSI, es definido por distintos autores como del desarrollo de una arquitectura de hardware y software para un entorno tecnológico que satisfaga las necesidades, ya sea de empresas u organizaciones.

El diseño del sistema tiene una parte muy importante en el procesamiento de datos, ya que por medio de este se puede hacer la acumulación y manipulación de los datos, y de esta manera poder generar información significativa que ayude a la toma de decisiones.

Es importante hacer un análisis del sistema ya que este tendrá relación con información, la cual es muy valiosa, y para que esta pueda ser de confianza a la hora de utilizar el sistema se debe dividir en tres partes según el autor Borges E. (2015).

- **Unidades:** el tipo de dato que se empleaba para la información, es necesario identificarlo ya que dependiendo el dato nos ayudará a poder reflejar mejor la información, la cual ayude a tomar una decisión.



Figura 19. Unidad de imagen. www.pixabay.com.

- **Procesos:** definir las instrucciones adecuadas para que puedan realizar las tareas que se les pide dentro del sistema, como lo son consultas, actualizaciones, reportes, eliminar o modificar información, entre otras tareas.



Figura 20. Proceso. Autoría propia.

- **Estructuras:** este es un punto muy crucial, ya que hablamos de las Bases de Datos, es importante definir bien los parámetros de ésta puesto que toda la información recae en ellas, se debe de tener una gran capacidad para manejar grandes capacidades de datos, así como múltiples tareas.

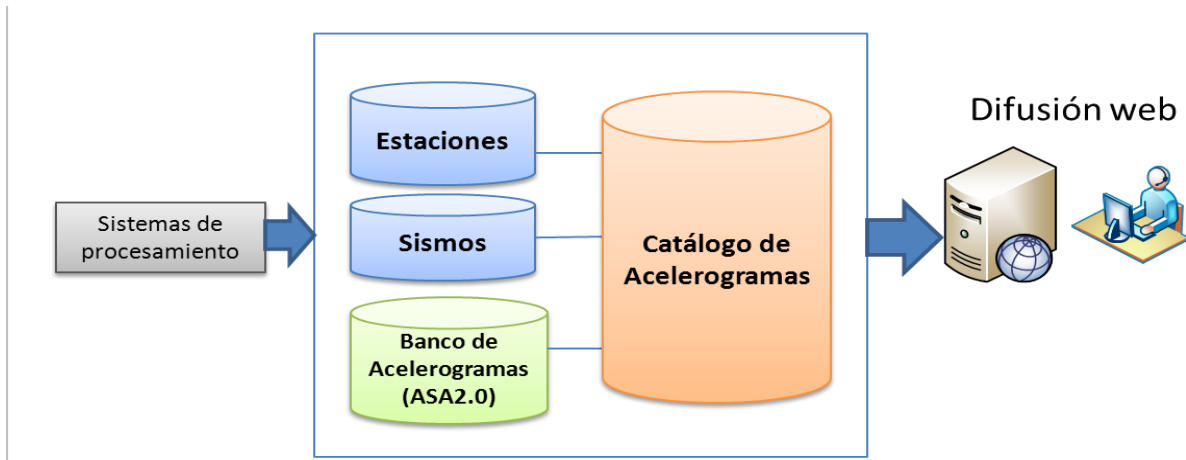


Figura 21. Estructura. Autoría propia.

Una parte fundamental del Diseño del Sistema es la programación, para lo cual debemos definir algunos conceptos que se usan a menudo en los diseños;

Programación: es el proceso en el cual se puede diseñar, codificar y depurar código fuente, para crear aplicaciones o sistemas informáticos que ayuden a cumplir una función.

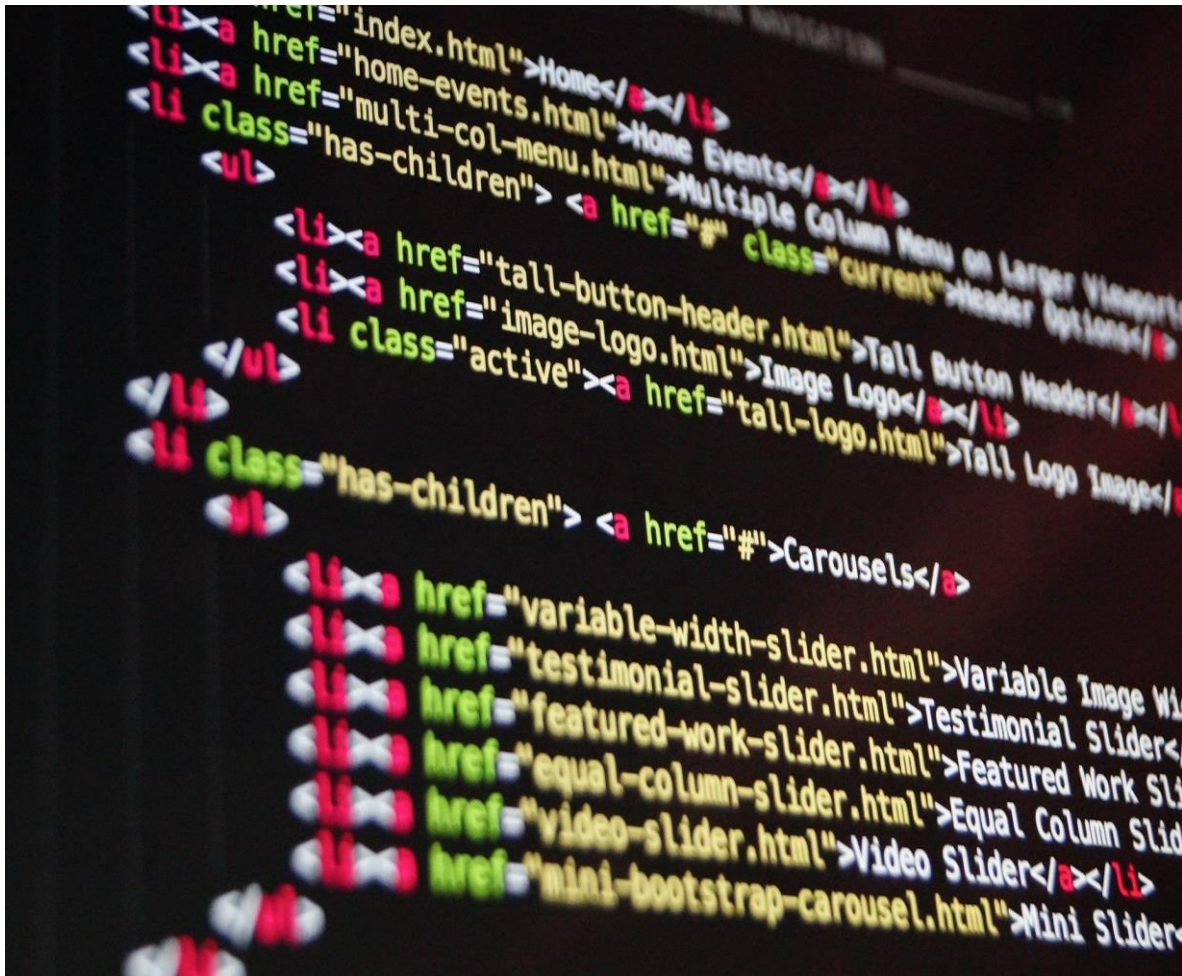


Figura 22. Programación. Autoría propia.

Programación web: es un tipo de programación que se dedica a crear sitios web, donde se puede meter información, imágenes, videos y audio, así como archivos con diversas extensiones.


```
<div class="container">
  <div class="row">
    <div class="col-md-6 col-lg-8"> <!-- BEGIN NAVIGATION
      <nav id="nav" role="navigation">
        <ul>
          <li><a href="index.html">Home</a></li>
          <li><a href="home-events.html">Home Events</a></li>
          <li><a href="multi-col-menu.html">Multiple Column Men
          <li class="has-children"> <a href="#" class="current">
            <ul>
              <li><a href="tall-button-header.html">Tall But
              <li><a href="image-logo.html">Image Logo</a></li>
              <li class="active"><a href="tall-logo.html">Ta
            </ul>
          </li>
          <li class="has-children"> <a href="#">Carousels</a>
            <ul>
              <li><a href="variable-width-slider.html">Variab
              <li><a href="tall-button-header.html">Testimoni
```

Figura 23. Programación Web. Autoría propia.

HTML: es un lenguaje de marcas de hipertexto, el cual sirve para la creación de páginas web. Este lenguaje crea la estructura básica de la página mediante código para la distribución del contenido (texto, imágenes, videos, juegos, etc.) en la página.

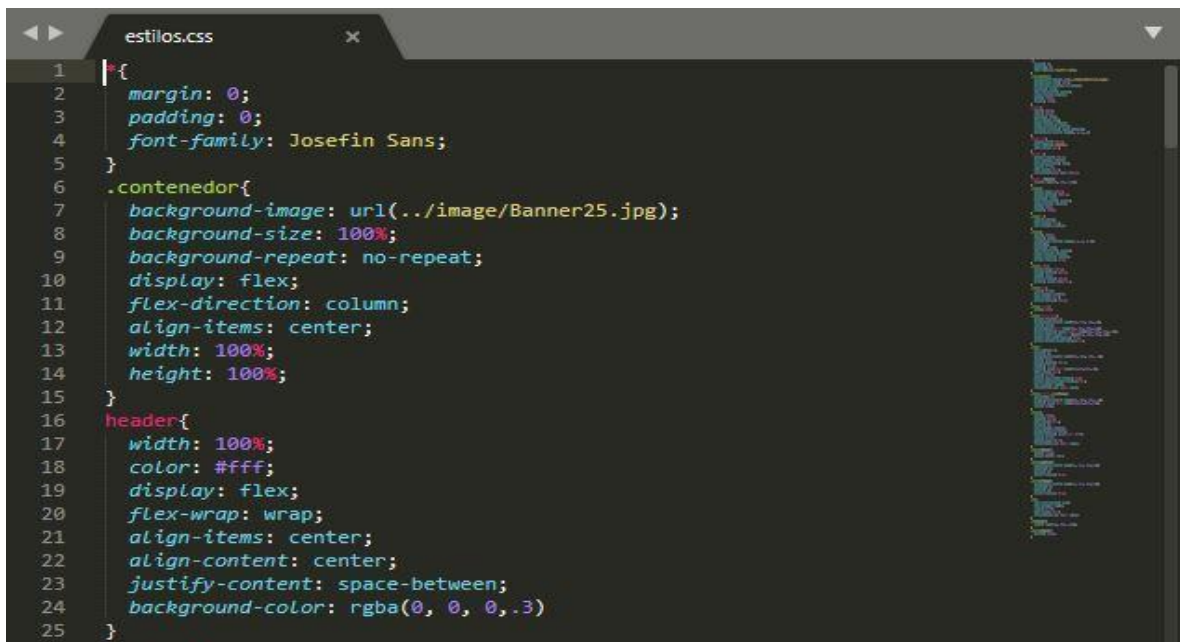
En la Figura.24 se muestra la estructura básica de un HTML es:

```
Documento html5

<!DOCTYPE html>
<html lang="es">
<head>
  <title>Documento HTML5</title>
  <meta-charset="UTF-8">
</head>
<body>
  <p>Espacio para el contenido</p>
</body>
</html>
```

Figura 24. Estructura HTML. Autoría propia.

CSS: las hojas de estilo en cascada o Cascading Style Sheets, nos permiten mejorar la visualización de nuestra página web, dando estilos amigables y coloridos para una mejor atracción del contenido.



```
estilos.css
1 {
2   margin: 0;
3   padding: 0;
4   font-family: Josefin Sans;
5 }
6 .contenedor{
7   background-image: url(../image/Banner25.jpg);
8   background-size: 100%;
9   background-repeat: no-repeat;
10  display: flex;
11  flex-direction: column;
12  align-items: center;
13  width: 100%;
14  height: 100%;
15 }
16 header{
17   width: 100%;
18   color: #fff;
19   display: flex;
20   flex-wrap: wrap;
21   align-items: center;
22   align-content: center;
23   justify-content: space-between;
24   background-color: rgba(0, 0, 0,.3)
25 }
```

Figura 25. CSS. Autoría propia.

Framework: Son herramientas que nos permiten escribir código más fácilmente, nos permite organizar todo de tal manera que sepamos en qué archivo está cada función que juega una parte vital dentro de la página web.



Figura 26. Concepto Framework. www.freepik.es

A continuación, se enlistan algunos tipos de Framework que son más utilizados en la actualidad:

Material Design:

CodeIgniter: Es un Framework muy completo que se encarga de tener un marco de desarrollo enfocado para el desarrollo web, así como aplicaciones, utilizando PHP. Permite a los usuarios poder generar el código de manera más rápida, ya que proporciona unas bibliotecas donde ya tiene funciones que te ahorran estar escribiendo el código desde cero, debido a que este conjunto de bibliotecas tiene ya las tareas comúnmente necesarias.

Su interfaz es totalmente simple, la cual ofrece una estructura lógica que accede a las bibliotecas, minimizando el código, y permite enfocarse de una forma más creativa al proyecto que se realice, ahorrando tiempo.



Figura 27. Framework CodeIgniter. www.codeigniter.com

Laravel: Este es un Framework Reciente de código totalmente libre para código PHP, tiene las mismas características que los demás Framework, pero el objetivo de este framework es facilitar al usuario que la sintaxis sea más expresiva y de una forma refinada a manera que el código sea de forma sencilla, permitiendo multitudes de funcionalidades, por lo que hace que sea un framework confiable



Figura 28. Framework Laravel. www.laravel.com

Procesador de Código: son aquellas herramientas que nos ayudan a crear, modificar y compilar cualquier lenguaje de programación, con la finalidad de poder modificar la tipografía, color tamaño, letra, objetos, dar efectos, como otras características de cada uno de ellos, a continuación, se muestran algunos tipos de procesadores de texto:

Sublime Text: este es un editor de texto y código fuente que sirve para poder modificar o crear programas para Páginas Web, Bases de Datos, API's, entre otras funciones, es uno de los procesadores de código más famosos entre la comunidad de programadores, por su fácil uso, tiene la ventaja de que se anticipa al reconocimiento de sintaxis, además de que es demasiado amigable la navegación para poder trabajar en su entorno.

Unas de las características de este programa que atrae mucho es que su fondo es totalmente negro lo cual es perfecto para poder resaltar el código, además de que el código se muestra en varias tonalidades dependiendo el tipo de texto que se escriba, esto ayuda notoriamente para poder identificar librerías, variables, etiquetas, párrafos, entre otras cosas.



Figura 29. Sublime Text. www.sublimetext.com

CodeBlocks: es también una buena opción para desarrollar código libre, ya que cuenta con un entorno integrado por desarrollo multiplataforma, que puede soportar diversos compiladores.

Al igual que varios procesadores de códigos este se puede mejorar agregando un paquete de plugins, de esta manera el sistema se vuelve más rápido para la construcción personalizado.

Una de las características por la que los usuarios ocupan este programa es porque tiene la herramienta “code folding”, que es la agrupación de código, facilitando mejor la sintaxis y la identificación del código en general.

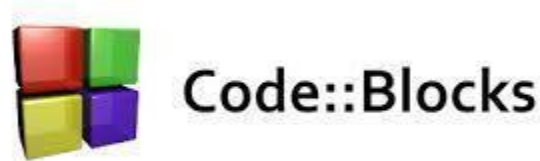


Figura 30. Code Blocks. www.codeblocks.org

NetBeans: este programa nos sirve como un entorno de desarrollo integrado, también conocido como “IDE”, el cual nos permite programar en diversos lenguajes. Los principales lenguajes con los que cuenta este procesador de código son los siguientes; Java, PHP, HTML, C++, C# y Ruby.

Este programa es casi en su totalidad de forma gratuita, lo que facilita tener acceso a esta herramienta que es muy famosa en el mundo de la programación para poder generar y editar código. Una de las de las características más relevantes de este programa por la cual también se usa mucho es que se puede trabajar con JAVA y todas las derivadas de esta, así mismo como PHP que es ideal para la creación de Páginas Web.



Figura 31. NetBeans. www.netbeans.org

Brackets: este es un editor de código fuente abierto principalmente enfocado en el desarrollo de páginas web, el cual es muy bueno para desarrollar código basado en JavaScript, HTML y CSS.

Este Software cuenta con una herramienta enfocada a la parte visual que hace más amigable el entorno para el usuario, así como un soporte de procesadores que facilita el diseño para la navegación.

Un punto clave de este Software, es su herramienta de soporte de procesadores, el cual ayuda a que sea más ligero a la hora de usar, así mismo se acompaña de su parte visual, para que se pueda brindar la máxima ayuda cuando se desee y sin interponerse a la parte creativa.



Figura 32. Brackets. www.brackets.io

Aplicaciones web: se define como una herramienta informática que, que puede acceder desde cualquier dispositivo y navegador, el cual se puede dar por medio de una internet o una red local, estas herramientas son muy común en día, las podemos encontrar en cualquier dispositivo móvil, pc o computadora, y su uso es muy fácil.

Una característica muy notoria es una aplicación web es que son multidispositivo y multiplataforma, esto quiere decir que nos olvidamos totalmente de instalar un software, en cada equipo, se puede acceder desde cualquier dispositivo, además

de que su funcionalidad no depende del equipo en el que se use, si no del servidor donde se aloje, esto quiere decir que no importa si no tienes un equipo tan potente, la aplicación funcionará totalmente.



Figura 33. Aplicación Web. www.netsoft.es

Librería: es un conjunto de archivo o archivos que en conjunto facilitan la programación, ya que en estos archivos se escriben varias funciones o tareas que ejecutan alguna función en específico en el código principal, la cual se llama mediante el nombre con el cual se llama el archivo, en el código principal.

Existen dos tipos de librerías, “las Caseras” y “las Externas”, las caseras se refieren a las librerías que creamos para un código en específico, es decir son librerías con autoría del programador.

Las librerías Externas es un archivo o archivos que son creados por expertos programadores, con funciones más compleja, las cuales se ocupan no para una parte en concreto, si no para la parte general, por lo general estas librerías ya viene incluidas en los Software de Código.

Una librería muy famosa entre la comunidad de programadores web es “Jquery”, el cual es una librería de JavaScript, la cual nos ayuda a facilitar e implementar mucho las partes que trabajan con JavaScript, de manera que trabaja con objetos Jquery para acceder de manera más rápida a los elementos de las páginas.

```
jquery.js x
1  /*! jQuery v3.1.0 | (c) jQuery Foundation | jquery.org/license */
2  !function(a,b){"use strict";"object"==typeof module&&"object"==typeof module.exports?module.exports=a.document?b(a,!0):function(a){if(!a.docume
3  r.isFunction(d)||g=!0),j&&(g?(b.call(a,d),b=null):(j=b,b=function(a,b,c){return j.call(r(a),c)}),b)}for(;h<i;h++)b(a[h],c,g?d.call(a[h],h,b
4  if(e&&1===a.nodeType)while(c=e[d++])a.removeAttribute(c)}),hb={set:function(a,b,c){return b===!1?r.removeAttr(a,c):a.setAttribute(c,c)},r.e
5
```

Figura 34. Librería. Autoría propia.

Modelo Vista Controlador: también conocido como “MVC”, es un modelo de la programación web enfocada a los objetos, esta propone un diseño de software para implementar un sistema donde se requiera el uso de una interfaces de usuario.

Lo que propone este modelo es que el código se divida en tres capas totalmente diferentes, pero que se relacionan una con otra, ordenarlas por jerarquía según su responsabilidad: “MVC” “Modelo, Views & Controllers”.

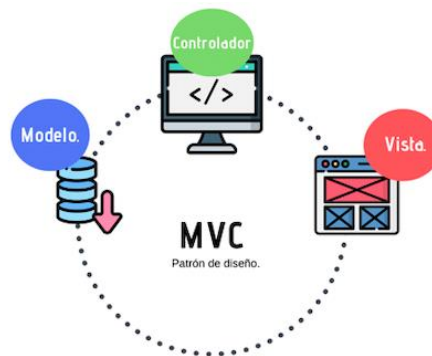


Figura 35. Modelo MVC. www.desarrolloweb.com.

Modelo: esta capa es donde se trabajará con los datos, en la cual se usan funciones para poder acceder a la información y de esta manera manipular sus estados, por lo que se usan bases usando palabras como select, updates, inserts, delete, entre otros.

Vista: en esta capa se tendrá todo el código relacionado con la visualización de nuestra aplicación de nuestra interfaces del usuario, por lo que en esta etapa se trabajará con código HTML y PHP, ya que con esta parte solo se mostraran salidas.

Controladores: se tendrá el código con el cual se podrá responder a las acciones que el usuario solicite en la aplicación, es decir, si presiona un botón con cierto efecto, que este lo ejecute para que se muestre en la visualización, o si se realiza una compra, que se ejecute adecuadamente el proceso para que el usuario pueda empezar, y finalizar, así como visualizar un objeto dentro de la aplicación, realizar una búsqueda, es decir esta capa es la que se encarga de realizar la conexión entre la capa de Modelo y Vista.

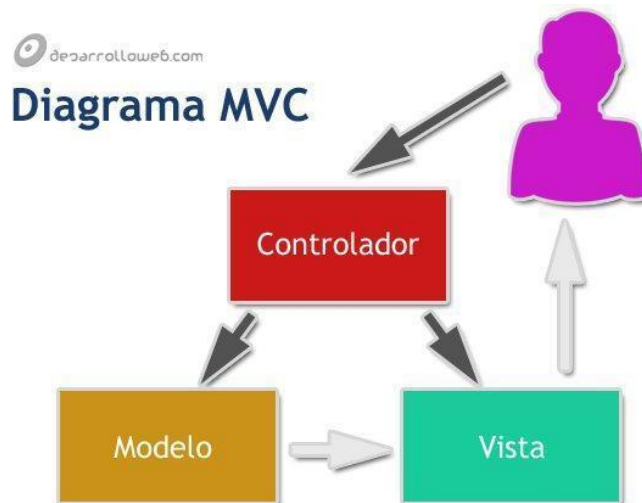


Figura 36. Ejemplo MVC. www.desarrolloweb.com

2.8. Implementación

Esta fase es conocida como ejecución, en la cual podemos empezar a validar subtareas y actividades, donde se reunirá y organizará toda la información previa que se consolidó en la etapa de planeación como en la etapa de diseño.

todas las etapas de un proyecto son importantes, pero quizá esta etapa sea la más valiosa, ya que en esta como su nombre lo dice, se implementará (ejecutará), el proyecto en el cual se ha trabajado y se ha reunido bastante información.

Lo que busca esta etapa es la comprobación de las hipótesis, poniendo en práctica todo lo recabado en las etapas anteriores, para poder dictaminar si el curso de la investigación va por buen camino o si es necesario cambiar algo, añadir o quitar, para que se llegue al resultado esperado de la investigación.

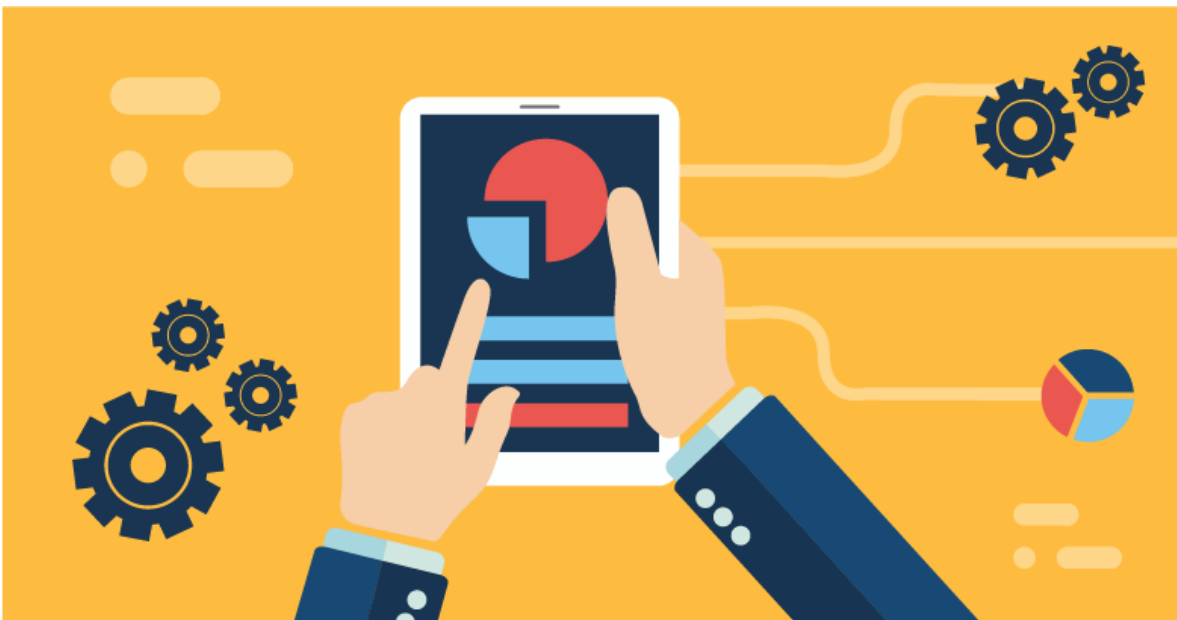


Figura 37. Etapa Implementación. Autoría Propia.

Es necesario comprender las siguientes etapas para poder saber que se requiere para la implementación:

- **Organizar:** todos los medios con los que se cuentan, humanos como materiales, del proyecto para que se pueda definir de una forma eficaz la asignación adecuada de cada tarea.
- **Controlar:** es necesario realizar una adecuada ejecución para poder tener en cuenta los posibles riesgos, y poder controlarlos o minimizar para que no se vea afectada la investigación.
- **Concluir:** una vez que termine la implementación es necesario comprender que dados los resultados obtenidos se tiene que dar entrega de los mismos para saber si se obtuvo lo esperado.



Figura 38. Requerimientos de la Implementación. Autoría Propia.

2.9. Pruebas del Sitio Web

La etapa de Pruebas del sitio web se refiere a realizar pruebas con cada una de las partes que conforman un sitio web, es decir, se tiene que checar la parte de seguridad, errores que puedan ocasionar que no funcione adecuadamente y por último pero no menos importante las pruebas con el usuario, quien será el usuario final al que se enfocara el sitio, ya que este nos ayudará a saber si el sitio es totalmente funcional para cubrir las necesidades por la cual se creó, así como si se es necesario realizar algunos cambios.

Es necesario comprender que existen millones de estructuras potenciales para un sitio web, que el diseño, la posición del contenido, e incluso el contenido de anuncios del sitio afecta en cómo navega el usuario en el sitio, por lo que es necesario equilibrar estos puntos para aumentar la experiencia de los usuarios y a su vez las visualizaciones del sitio web suban y el tiempo de sesión de cada usuario sean más prolongadas.



Figura 39. Pruebas de Página Web. Autoría Propia.

3. SEGURIDAD EN LAS PÁGINAS WEB

Hoy en día es común usar una página web casi para cualquier cosa, pero un punto crítico en estos sitios es la seguridad. Si se tiene un sistema que guarde información es de vital importancia que se cuente con seguridad dentro del sitio, para resguardo e integridad de la información, y evitar el mal uso de la misma.

El propósito primordial de la seguridad en un sistema es que se inviolable, denegando cualquier acceso no permitido, fuga de información o la extracción de esta. Para lo cual se tiene que proteger desde el punto Lógico(Software), código que pueda encriptar la información, así como algunas medidas preventivas que no dejen entrar código malicioso, otra manera de apoyar a la seguridad lógica es mediante la parte física(Hardware), la cual puede ser desde un lector de huella, hasta un lector de retina o ADN, esto con el fin de tener un sistema más seguro y complejo, dependiendo del sistema y para que se use



Figura 40. Seguridad informática. www.pixeles.com

3.1 Ataques

El Ataque a un Sistema informático a el acto de acceder sin permiso, generando un fallo en el sistema, pérdida de información o robo de datos.

los ataques más comunes son los siguientes:

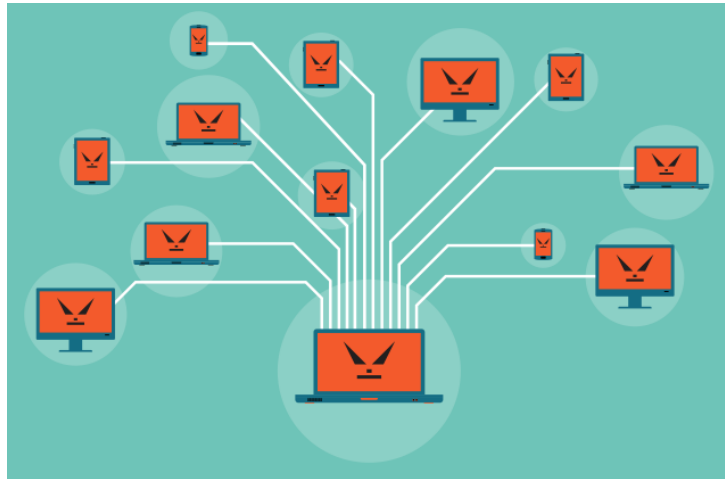


Figura 41. Ataques. www.kaspersky.es

- **DDoS**: este ataque está específicamente enfocado a los servidores, el cual manda varias solicitudes al servidor web haciendo que no se procese dicha solicitud, arrojando la leyenda de “Denegación de Servicio”, de esta manera no se permite la entrada al sitio solicitado.

Existen diversos tipos de ataques Dos:



Figura 42. Ataque DDos. www.kaspersky.es

- **Tear Drop Attack:** este ataque manda una serie de paquetes a la computadora destino, estos paquetes van con una serie de súper valores para que hacen que se posicione en primer lugar y bloquee la computadora y hacer que no funcione o se reinicie.



Figura 42. Ataque Drop Attack. www.kaspersky.es

- **Smurf Attack:** este ataca a las redes enviando una gran cantidad de solicitudes de eco ICMP a la dirección IP Broadcast, haciendo que todas las computadoras envíen la solicitud a la red, haciendo que este empiece a trabajar lento y a fallar, ya que tiene muchas peticiones que cubrir.

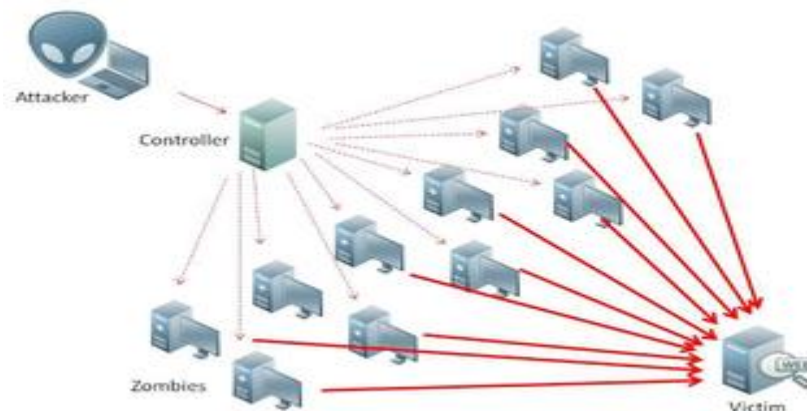


Figura 42. Ataque Smurf Attack. www.kaspersky.es

- **SYN Flood:** se enfoca a atacar al servidor por medio de las peticiones de conexión, este envía solicitudes de conexión con remitente falso, haciendo que

se sature el número de conexión del servidor, denegando las solicitudes de las conexiones reales.



Figura 45. Ataque SYN Flood. www.kaspersky.es

- **Fraggle DDoS Attack:** es similar al “ Smurf Attack”, solo que esta manda infinidad de paquetes UDP, los cuales envía a través de la IP falsificada a el servidor víctima, entrega la dirección IP a todos los hosts para que a su vez ellos respondan esos paquetes, que recibirá la red atacada, y se saturara, haciendo que sea incapaz de funcionar.



Figura 46. Ataque Fraggle DDoS Attack. www.kaspersky.es

- **Ping Flood:** este tipo de ataque funciona mandando una gran cantidad de paquetes ping al sistema, red o servidor, con el fin de saturar el número de peticiones para que el sistema se alente y deje de funcionar hasta dejar de

contestar las peticiones, este tipo de ataque afecta más a usuarios que utilizan el sistema operativo UNIX, en Windows no tiene tanta repercusión.



Figura 47. Ataque Ping Flood. www.kaspersky.es

- **Ping de la muerte:** este ataque se caracteriza principalmente por enviar un paquete de datos de 65.536 bytes, lo que sucede es que el sistema operativo no sabe qué hacer con un paquete grande se congela. Actualmente en los sistemas operativos ya no tienen este problema ya que descarta este tipo de paquetes, y esto hace que el ataque “Ping de la Muerte” ya no tenga efecto como en su momento lo hizo.



Figura 48. Ataque Ping de la Muerte. www.kaspersky.es

- **DoS:** este tipo de ataque se caracteriza por apoderarse de otras computadoras, y su principal tarea es para enviar miles de datos a sitios web o correos no deseados a cuentas de correos en específico, con el objetivo de denegar el servicio de un sitio web. Actualmente este ataque se realiza a través de una red de robots informáticos que automáticamente hacen esta labor del ataque, la cual puede controlar todas las computadoras y/o servidores que estén bajo el efecto del ataque.



Figura 49. Ataque DoS. www.kaspersky.es

Escaneo de Puertos: este ataque es muy popular hoy en día, ya que se usa para detectar los servicios expuestos a un posible ataque, la forma de operar de este

ataque es mandando un mensaje a cada puerto para ver en cual se encuentra la exposición para el ataque.

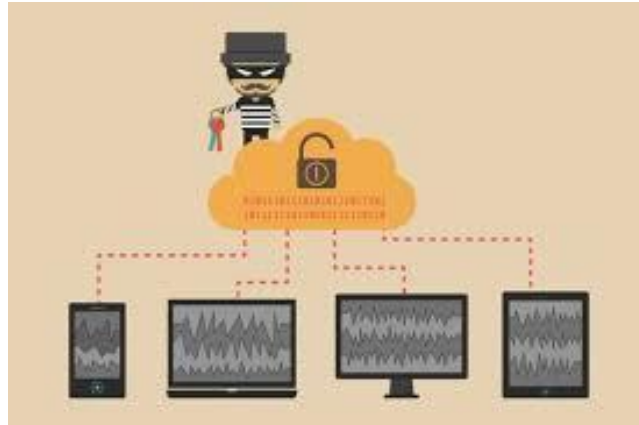


Figura 50. Escaneo de Puertos. www.kaspersky.es

- **ARP Spoofing:** este ataque trabaja con las redes ARP y las inalámbricas, la forma de atacar es clonando las direcciones MAC, para atacar switches, manda una serie de paquetes, todos con dirección MAC diferente, esto con el fin de consumir la memoria que tiene el mismo y hacer que se congele y sea imposible responder todas las peticiones.



Figura 51. Ataque ARP Spoofing. www.kaspersky.es

3.2 Amenazas

Cualquier Software (programa), que ponga en riesgo la integridad de nuestro o nuestros equipos, así como datos o información, los cuales se engloban en una sola palabra:



Figura 52. Amenazas. www.kaspersky.es

Malware: Se cataloga así a todo Software malicioso que tiene como principal objetivo infiltrarse al sistema para poder ocasionar daños, como rendimiento bajo, mal funcionamiento o pérdida de información, estos Software se clasifican en los siguientes:



Figura 53. Malware. www.kaspersky.es

- **Virus:** estos son programas sencillos cuyo fin es infectar a otros archivos mediante inyección de código para tomar el control de estos archivos y convertirlos en una nueva fuente de infección, regularmente los virus afectan a programas ejecutables y la forma de poder eliminarlos son con los llamados “Antivirus”.



Figura 54. Virus. www.kaspersky.es

- **Gusanos:** este tipo de malware ocupa los recursos de la red para poder desplazarse de un equipo a otro, ya que no necesita un archivo anfitrión para poder prevalecer, una vez que entra a un equipo, trata de acceder a la memoria RAM para poder generar sus copias y mandarlas ya sea por correo o por la misma red a la que esté conectado el equipo, lo que tratan de hacer es crear equipos zombis para posteriormente se pueda usar el equipo infectado para generar ataques como el DoS y sus derivados.



Figura 55. Virus Gusano. www.kaspersky.es

- **Trojanos:** son similares a los virus, pero a diferencia de ellos no se propagan, lo que hace este tipo de malware, es abrir una puerta para que de forma ilegal

entren programas maliciosos, su principal tarea es recolectar información para posteriormente mandarlo al creador de dicho troyano, para tener acceso a cuentas bancarias, cuentas de correo etc. Por lo general estos programas vienen escondidos en programas ejecutables que parecen inofensivos.



Figura 56. Virus troyanos. www.kaspersky.es

- **Spyware:** son programas diseñados para la recopilación de información de una persona u organización sin su consentimiento, se utiliza generalmente en todo lo que consulta en su navegador, o Apps, también se puede emplear para controlar los dispositivos de forma remota.



Figura 57. Virus Spyware. www.kaspersky.es

- **Phishing:** es un programa dedicado a la recolección de contraseñas y claves, preferentemente de cuentas bancarias o cualquiera que tenga que ver con algo económico, estos se infiltran por medio de los correos, envían un enlace de un sitio falso, donde les pedirán ingrese su usuario y contraseña, el usuario

proporciona estos datos pensando que el sitio es seguro, y en ese momento el sitio ya cuenta con el usuario y contraseña para acceder a la o las cuentas.

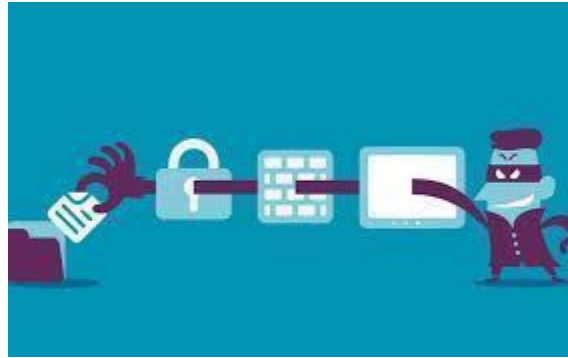


Figura 58. Virus Phishing. www.kaspersky.es

- **Adware:** la principal función de este programa es de “Mostrar Publicidad”, una vez que el usuario da clic en una publicidad, se recopila la información de dicho usuario, esto con el fin de mandar esta información como estudio de las preferencias y orientar el tipo de publicidad.



Figura 59. Virus Adware. www.kaspersky.es

- **Ransomware:** también conocido como secuestrador, se encarga de encriptar información, discos duros, e inclusive equipos completos, para que el usuario se ponga en contacto con el criminal y se pague para obtener una clave, en algunos casos se programa la encriptación con un tiempo por si el usuario no paga el rescate o a tiempo la información sea borrada o se deje inservible el equipo.



Figura 60. Virus Ransomware. www.kaspersky.es

3.3 Cifrado

Este es un Proceso en el cual la información se transcribe de una forma que no se pueda leer, únicamente por la persona, sistema o programa que lo cifra, dependiendo de éste, la información aparecerá con símbolos, letras, caracteres o número. Estos son algunos tipos de Cifrados:

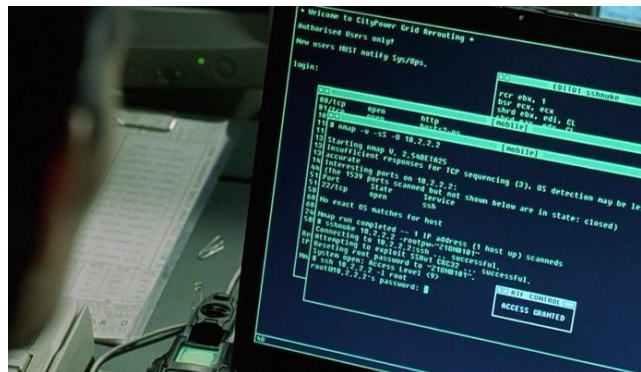


Figura 61. Cifrado. www.solvetic.com

- **Encriptación Manual:** para generar este tipo es necesario usar un software, para eso el usuario debe de elegir qué archivos o información desea cifrar.



Figura 62. Encriptado Manual. www.solvetic.com

- **Cifrado Transparente:** se ocupa mediante un Software, lo que hace es cifrar toda la información que se encuentre en una computadora o dispositivo.



Figura 63. Encriptado Transparente. www.solvetic.com

- **Cifrado Simétrico:** este tipo de cifrado no es necesario ocupar un programa para hacerlo, se puede cifrar la información por uno mismo, puedes cambiar una letra por un número, o un número por una letra, por símbolos o caracteres, según sea el caso con que se acomode un, por ejemplo.

- o e=1; r=2; n=3; e=1; s=5; t=6; o=7;
- o 1=e; 2=d; 2=d; 5=y;
- o !=U; “=N; #=A; \$=M;

Figura 64. Cifrado Simétrico. Autoría Propia.

➤ **Cifrado Asimétrico:** se realiza con un programa, en el cual se pondrá la llave pública, la cual se debe proporcionar a quien reciba la información cifrada para que pueda tener acceso a ella, ya que pedirá esa llave (clave).

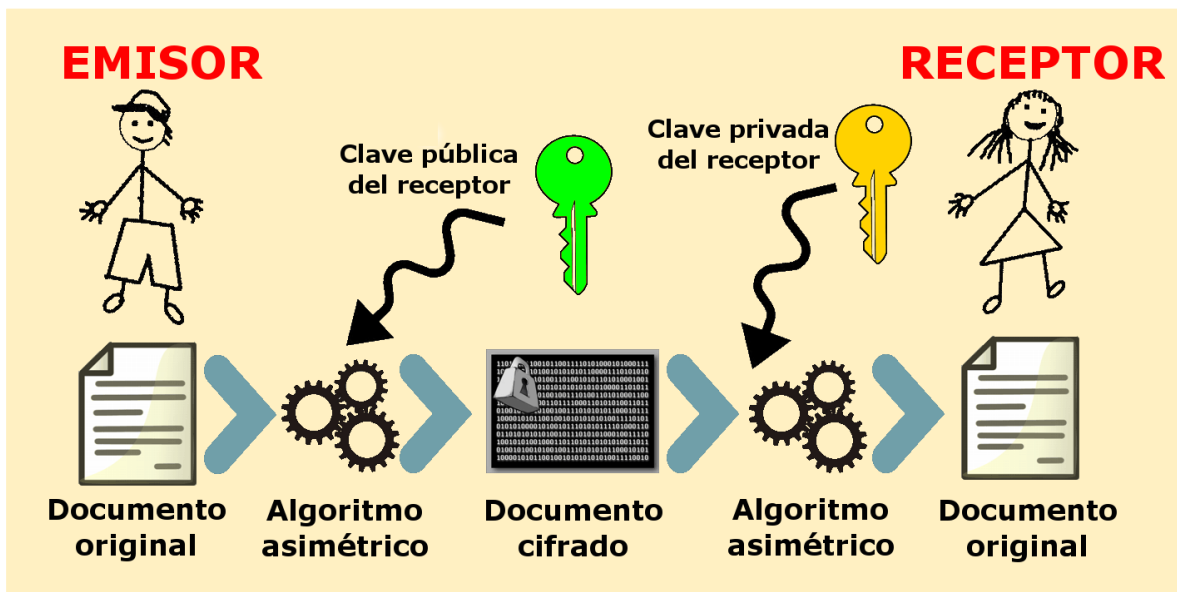


Figura 65. Cifrado Asimétrico. www.solvetic.com

4. BASES DE DATOS

Se considera base de datos a un sistema que gestiona la administración de los datos para que en conjunto la convierta e información útil para una persona, empresa u organización.

La cual puede aceptar todo tipo de caracteres alfanuméricos, o numéricos, que se caracteriza por estar altamente estructurada y organizada para el almacenamiento de la información, para que posteriormente se ofrezca la posibilidad de manipular esa información mediante la edición, inserción, eliminación, actualización , creación de tablas, campos y registros de datos, todo esto mediante un lenguaje de programación orientada a las Bases de Datos, esto ofrece que se pueda consultar hasta recuperar información, asignando parámetros ya sea establecidos o a elegir.



Figura 66. Base de Datos. www.maestrosdelweb.com

La interfaz mostrada en una Base de Datos tiene como fin poder declarar el objetivo, así como la operación que se desea realizar, así como el tipo de información que

se empleará, la seguridad que se usará, y quizá una de las principales funciones; declarar las Llaves para la conexión entre diversas tablas para obtener la información totalmente completa a la hora de consultar.

Existen varios tipos de bases de datos, cada una de estas se acoplan a las necesidades del sistema que las adecua para el correcto funcionamiento, organización e implementación de la información que resguardaran.



Figura 67. Componentes de Base de Datos. www.maestrosdelweb.com

- **Jerárquico:** se asemeja a la estructura de un árbol, tal como Microsoft Windows organiza las carpetas y archivos. En un modelo jerárquico de bases de datos, cada enlace es anidado con el fin de conservar los datos organizados en un orden particular en un mismo nivel de lista.



Figura 68. Jerarquía. www.maestrosdelweb.com

- **Plano:** hay dos dimensiones (estructura plana) de conjunto de datos. Hay una columna de información y dentro de esta columna, se supone que cada dato tendrá que ver con la columna.



Figura 69. Plano Base de Datos. www.maestrosdelweb.com

- **Red:** la característica definitoria es que se almacena un registro con un enlace a otros registros en efecto, una red. Estas redes (o, a veces, a que se refiere como punteros) puede ser una variedad de diferentes tipos de información como números de nodo de un disco o incluso la dirección.



Figura 70. Red en Base de Datos. www.maestrosdelweb.com

- **Relacional:** es el más popular tipo de base de datos y una herramienta extremadamente potente, no sólo para almacenar información, también para acceder a ella. Las bases de datos relacionales son organizadas en forma de tablas. La belleza de estos cuadros es que la información se puede acceder o añadir sin reorganizar las tablas.



Figura 71. Base de Datos Relacional. www.maestrosdelweb.com

4.1 Diseño

Una base de datos correctamente diseñada le permite obtener acceso a información actualizada y precisa. Como es esencial tener un diseño correcto para lograr sus objetivos de trabajar con una base de datos, tiene sentido invertir el tiempo necesario para obtener información sobre los principios de un buen diseño. Al final, es mucho más probable que acabe con una base de datos que se ajusta a sus necesidades y que puede adaptarse fácilmente al cambio.



Figura 72. Diseño de Base de Datos. www.maestrosdelweb.com

El proceso de diseño se basa de la siguiente forma:

- **Determinar el propósito de la base de datos:** Esto le ayudará a prepararse para los pasos restantes. Ya que es la razón del para qué crear la Base de Datos y el funcionamiento de la misma.



Figura 73. Propósito de una Base de Datos. www.maestrosdelweb.com

- **Buscar y organizar la información necesaria:** Recopile todos los tipos de información que podría querer registrar en la base de datos, como los nombres

de producto y los números de pedido. se tiene que contemplar cada dato que se usará para definir de forma correcta los datos.



Figura 74. Información a usar. www.maestrosdelweb.com

- **Dividir la información en tablas:** Divida los elementos de información en entidades principales o temas, como Productos o Clientes. Después, cada tema se convierte en una tabla.

	idusuario	usuario	password	email	nivel	idzona	idconcesionario
<input type="checkbox"/>	1	john01	asdf1234	jig@intemetya.com.co	1	0	0
<input type="checkbox"/>	10	Sonia Rodriguez	Srodriguez2	srodriguez@sofasa.com.co	2	2	-1
<input type="checkbox"/>	3	Jorge Nino	Jnino1	jnino@sofasa.com.co	2	1	-1
<input type="checkbox"/>	4	Nestor Moreno	Nmoreno15	nmoreno@sofasa.com.co	3	1	.3
<input type="checkbox"/>	5	Lina Berrio	Metis123	lberrio@metisla.com	1	-1	-1
<input type="checkbox"/>	6	Tatiana Hurtado	Thurtado17	thurtado@metisla.com	1	-1	-1
<input type="checkbox"/>	7	Diana Vargas	Dvargas18	dvargas@metisla.com	1	-1	-1
<input type="checkbox"/>	8	Luis Henao	Lhenao16	lhenao@sofasa.com.co	2	-1	-1
<input type="checkbox"/>	11	Tatiana Orrego	Torrego3	torrego@sofasa.com.co	2	3	-1
<input type="checkbox"/>	12	Herman Dario Molina	Hmolina4	hmolina@sofasa.com.co	2	4	-1
<input type="checkbox"/>	13	Fabio Barrera	Fbarrera5	fbarrera@sofasa.com.co	2	5	-1
<input type="checkbox"/>	14	Moises Rodriguez	Mrodriguez6	mrodriguez@sofasa.com.co	2	6	-1
<input type="checkbox"/>	15	Larry Gomez	Lgomez7	lgomez@sofasa.com.co	2	7	-1
<input type="checkbox"/>	16	Juan Camilo Escobar	Jescobar8	jescobar@sofasa.com.co	2	8	-1
<input type="checkbox"/>	17	Jimena Lemaitre	Jlemaitre9	jlemaitre@sofasa.com.co	2	9	-1

Figura 75. Tablas. Autoría Propia.

- **Convertir los elementos de información en columnas:** Decida qué información quiere almacenar en cada tabla. Cada elemento se convierte en un campo y se muestra como una columna en la tabla. Por ejemplo, una tabla de empleados podría incluir campos como Apellidos y Fecha de contratación.

Tablas	CALIFICACIONES	Lista de Alumnos de Musica	NIVELES DE APRENDIZAJE	TURNO				
CALIFICACIONES	RFC	Primer Ap	Segundo Ap	Nombre	Direccion	Telefono	Fecha de	Lugar de
Lista de Alumnos de Musica	CAPM910430	CARRILLO	PEÑA	MANUEL	C. 21 # 37 X 45	9923344	30/04/1991	KANASIN
NIVELES DE APRENDIZAJE	CAPY890308	CARRILLO	PEÑA	YUSELMY SARA	C.83 # 12 X 7 Y	9734737777	08/03/1989	MERIDA
TURNO	GAGO690826	GALVAN	GONZALEZ	OSCAR	C. 43 # 345 X 24	9991939838	26/08/1969	MERIDA
	PEMA781125	PECH	MERIDA	ANGELINA ISAI	C. 12 # 233 X 6	9232343	25/11/1978	MERIDA
	PERN790704	PEÑA	RODRIGUEZ	NEFTALI	C. 34 # 98 MISA	9993767383	04/07/1979	MERIDA
	PEMA880109	PEÑA	PECH	MATIAS	C. 32 # 23 X 89	9233736	15/09/1990	MERIDA
	PEMA900119	PEÑA	PECH	LUIS	C. 34 # 993 X 12	9993373727	28/09/1946	MERIDA
	PERY701208	PEÑA	RODRIGUEZ	YUSELMY	C. 24 # 94 X 17	9224030	08/12/1970	MERIDA
	ROPM501008	RODRIGUEZ	PEREZ	MARIA EUGEN	C.22 # 382 X 12	9992993384	08/10/1950	MERIDA
	TRCJ770120	TRINIDAD	CASTILLO	JOSE	C. 22 # 323 SAN	9937645	20/01/1977	MERIDA
	VAVD700702	VAZQUEZ	VENTURA	DULCE	C. 65 # 553 CEN	1838848	02/07/1970	MERIDA

Figura 76. Información Columnas. Autoría Propia.

- **Especificar las claves principales:** Elija la clave principal de cada tabla. La clave principal es una columna que se usa para identificar cada fila. Un ejemplo podría ser Id. de producto o Id. de pedido.

	idusuario	usuario	password	email	nivel	idzona	idconcesionario
<input type="checkbox"/>	1	john01	asdf1234	jig@internetya.com.co	1	0	0
<input type="checkbox"/>	10	Sonia Rodriguez	Srodriguez2	srodriguez@sofasa.com.co	2	2	-1
<input type="checkbox"/>	3	Jorge Nino	Jnino1	jnino@sofasa.com.co	2	1	-1
<input type="checkbox"/>	4	Nestor Moreno	Nmoreno15	nmoreno@sofasa.com.co	3	1	3
<input type="checkbox"/>	5	Lina Berrio	Metis123	lberrio@metisla.com	1	-1	-1
<input type="checkbox"/>	6	Tatiana Hurtado	Thurtado17	thurtado@metisla.com	1	-1	-1
<input type="checkbox"/>	7	Diana Vargas	Dvargas18	dvargas@metisla.com	1	-1	-1
<input type="checkbox"/>	8	Luis Henao	Lhenao16	lhenao@sofasa.com.co	2	-1	-1
<input type="checkbox"/>	11	Tatiana Orrego	Torrego3	torrego@sofasa.com.co	2	3	-1
<input type="checkbox"/>	12	Hernan Dario Molina	Hmolina4	hmolina@sofasa.com.co	2	4	-1
<input type="checkbox"/>	13	Fabio Barrera	Fbarrera5	fbarrera@sofasa.com.co	2	5	-1
<input type="checkbox"/>	14	Moises Rodriguez	Mrodriguez6	mrodriguez@sofasa.com.co	2	6	-1
<input type="checkbox"/>	15	Larry Gomez	Lgomez7	lgomez@sofasa.com.co	2	7	-1
<input type="checkbox"/>	16	Juan Camilo Escobar	Jescobar8	jescobar@sofasa.com.co	2	8	-1
<input type="checkbox"/>	17	Jimena Lemaitre	Jlemaitre9	jlemaitre@sofasa.com.co	2	9	-1

Figura 77. Asignar Id. Autoría Propia.

- **Establecer las relaciones de tablas:** Busque en cada tabla y decida cómo se relacionan los datos en una tabla con los datos de otras tablas. Agregue campos a las tablas o cree tablas para aclarar las relaciones, según sea necesario.

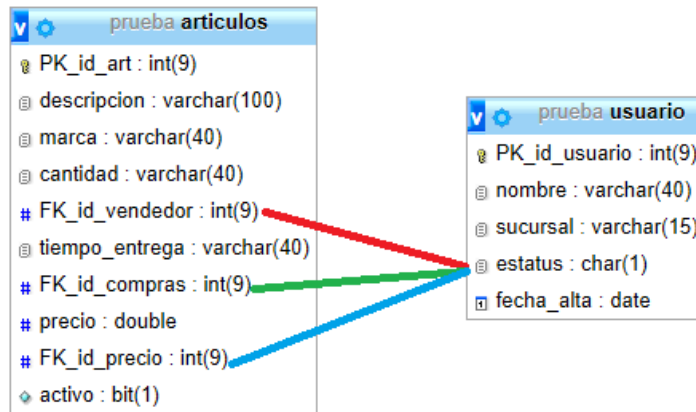


Figura 78. Relación. Autoría Propia.

- **Perfeccionar el diseño:** Analice el diseño en busca de errores. Cree las tablas y agregue unos cuantos registros de datos de ejemplo. Compruebe si puede obtener los resultados que quiere de las tablas. Haga algunos ajustes en el diseño, si es necesario.

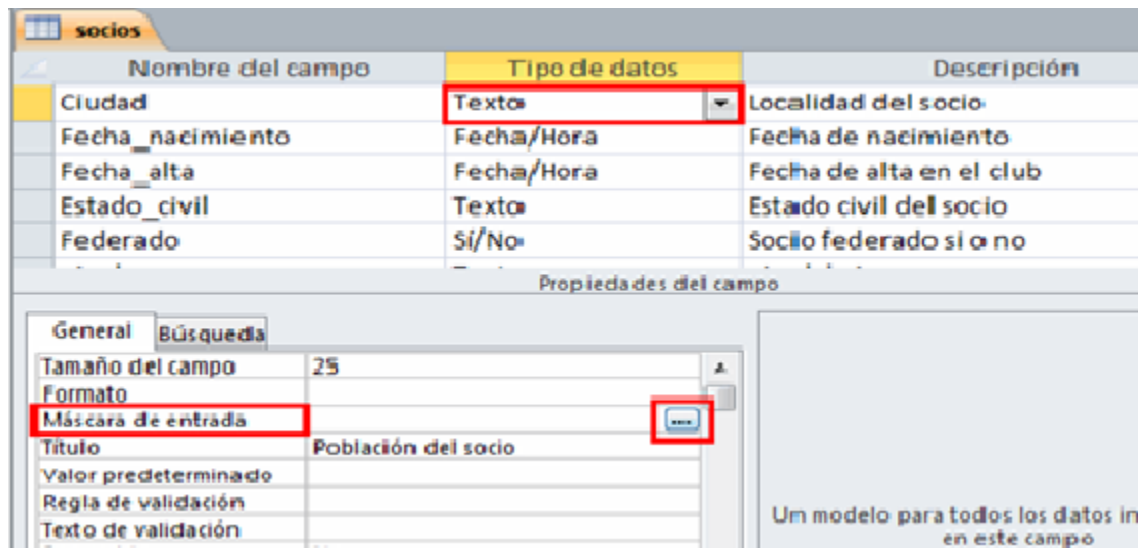


Figura 79. Perfección de Diseño. Autoría Propia.

- **Aplicar las reglas de normalización:** Aplique las reglas de normalización de datos para ver si las tablas están estructuradas correctamente. Haga algunos ajustes en las tablas, si es necesario.

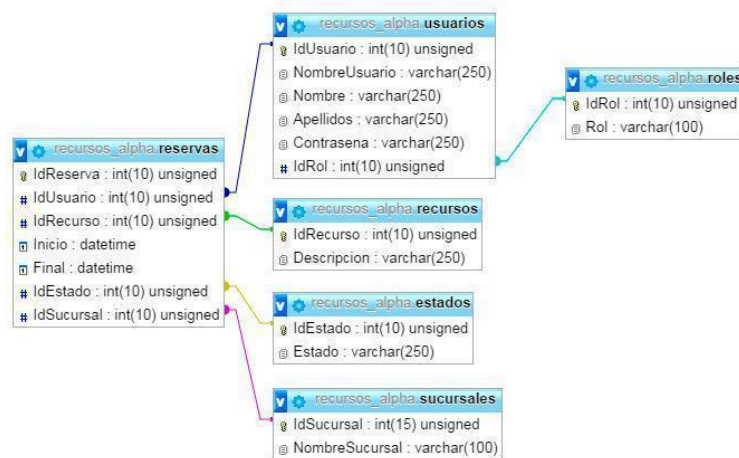


Figura 80. Reglas de Normalización. Autoría Propia.

4.2 Diagrama Entidad- Relación

Las bases de datos son un gran pilar de la programación actual, ya que nos permiten almacenar y usar de forma rápida y eficiente cantidades ingentes de datos con cierta facilidad. En la actualidad se usa de forma mayoritaria las bases de datos relacionales (dominadas por distintos gestores a través del lenguaje SQL, en gran medida).

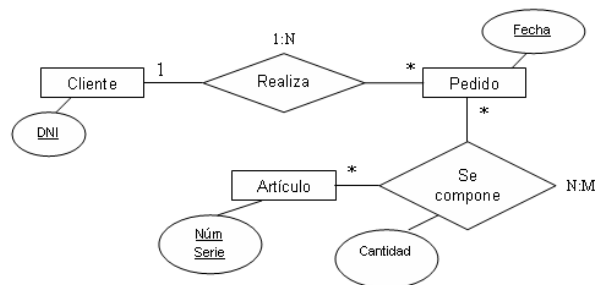


Figura 81. Diagrama Entidad Relación. Autoría Propia.

Los elementos son los siguientes:

Entidad:

Las entidades representan *cosas* u *objetos* (ya sean reales o abstractos), que se diferencian claramente entre sí.

Ejemplo de entidades

- Proveedor
- Sucursal
- Pedido

Atributos:

- Los atributos definen o identifican las características de entidad (es el contenido de esta entidad). Cada entidad contiene distintos atributos, que dan información sobre esta entidad. Estos atributos pueden ser de distintos tipos (numéricos, texto, fecha...).
- Siguiendo el ejemplo de antes podemos analizar los atributos de la entidad "Proveedor", que nos darán información sobre los proveedores de la empresa.

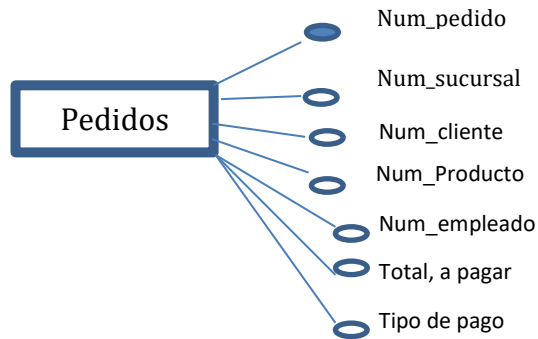


Figura Núm. Ejemplo diagrama entidad relación. Autoría propia.

4.3 Seguridad en las Bases de Datos

Esto se refiere a las medidas de protección digital que se aplica para evitar el o los accesos no autorizados a los datos, sin importar porque tanto sea el nivel de importancia de los datos de la información que se almacenan en las bases de datos, se debe de tomar en cuenta el factor si no el más importante uno de los que recaen peso para evitar una manipulación indebida en la información.



Figura 82. Seguridad en Base de Datos. www.maestrosdelweb.com

5. CASO PRÁCTICO

Para realizar el desarrollo de la plataforma, se utilizaron una serie de herramientas, metodologías y buenas prácticas que sirven para que un Proyecto de Software sea mantenible y escalable con el tiempo.

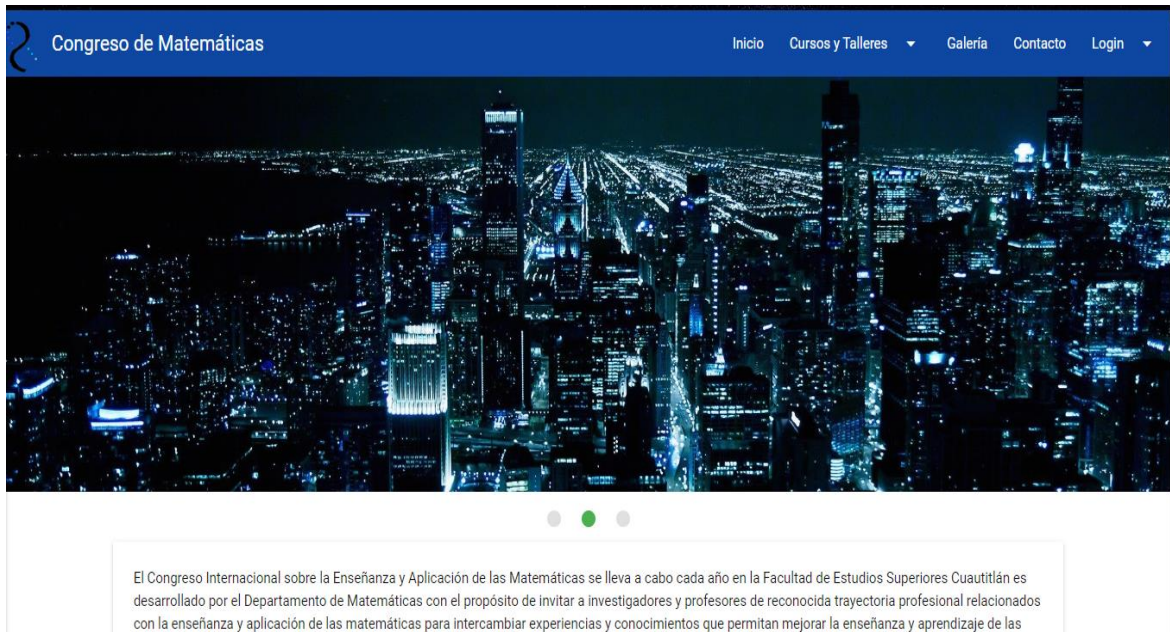


Figura 83. Propuesta de Sitio Web. Autoría Propia.

5.1 Arquitectura de Software

Para la automatización e implementación de los procesos de mejora en el sitio web Congreso de Matemáticas, se ha seguido un camino que suele llevarse en diversos entornos o empresas de Ingeniería de Software.

Comenzamos haciendo un análisis, que nos permitió conocer los procesos, los formatos, las áreas de control de acceso y las necesidades que tiene este sistema, se realizaron entrevistas, cuestionarios, y se estructuró la arquitectura de software donde nos enfocamos en diseñar, a nivel lógico las soluciones que plantea ésta tesis, por ello se usan diversas técnicas y se definen ciertas reglas que hemos de seguir.

Así mismo, de acuerdo a la **Arquitectura de Software** se deben definir los Atributos de calidad, con el análisis detectamos la importancia de estos en específico:

- **Mantenibilidad**

- Seguridad
- Eficiencia de ejecución

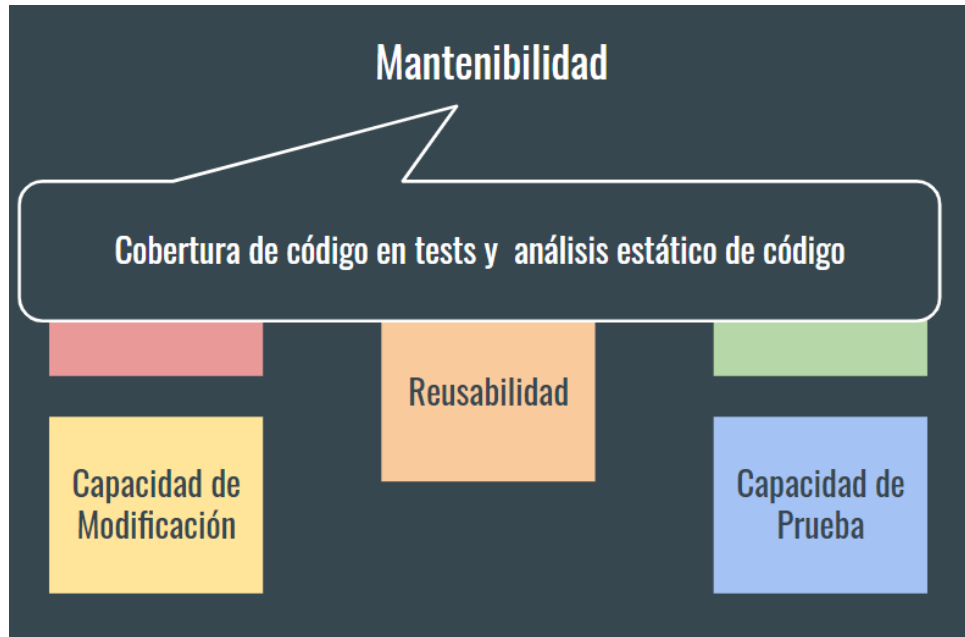


Figura 84. Mantenibilidad. Autoría Propia.

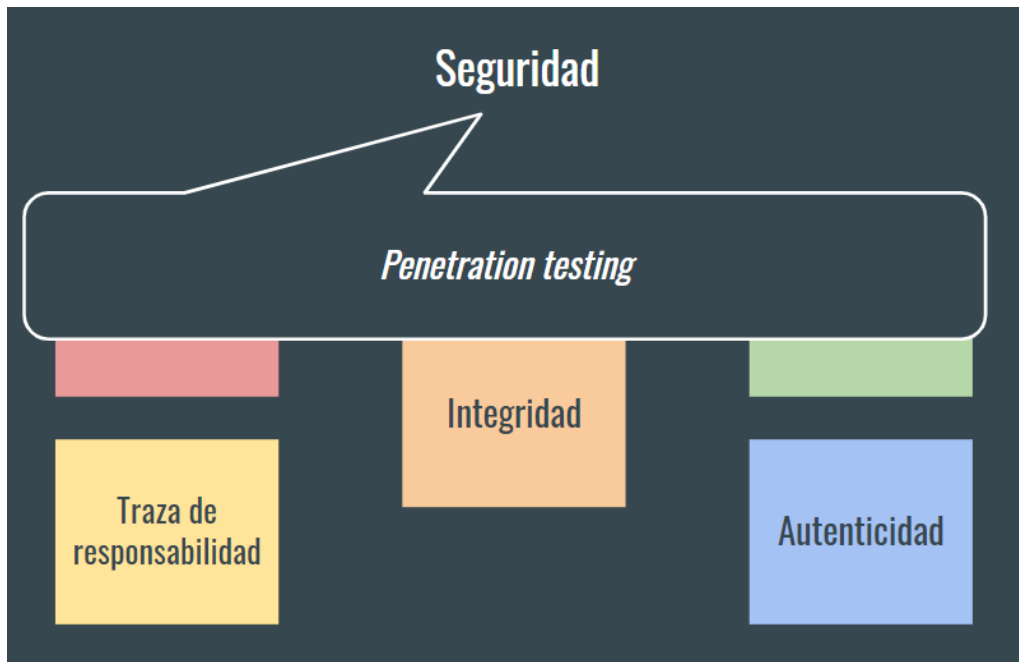


Figura 85. Seguridad. Autoría Propia.

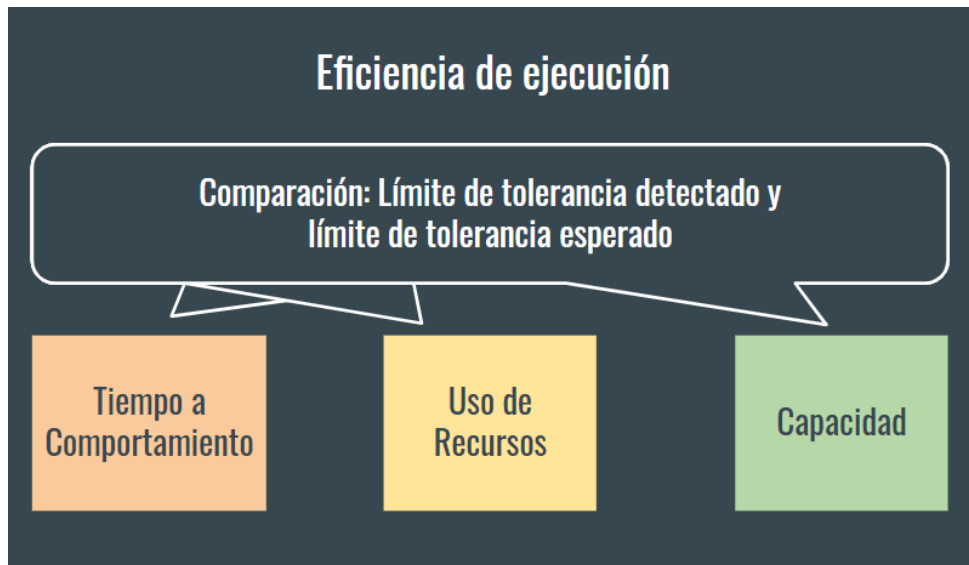


Figura 86. Eficiencia de ejecución. Autoría Propia.

Básicamente buscamos en la mantenibilidad que el código se pueda modificar, probar y reutilizar.

Que la seguridad debe ser una premisa importante en cuanto a integridad de los datos pues hablamos de una escuela, y la información de quiénes y qué la conforma es sensible.

Y la Eficiencia de ejecución, en donde buscamos que tenga la capacidad para soportar de forma óptima la carga de trabajo que llegue en los momentos picos del Congreso.



Figura 86. Calidad de un Sitio web. mx.depositphotos.com

En cuanto a la Arquitectura, existen tantas como MVC, Capas, de Eventos, Micro servicios, Comparte-nada, etc. Con el análisis previo determinamos que utilizaremos una Arquitectura

Un patrón combinado utilizando

- Modelo-vista-controlador
- Comparte-nada

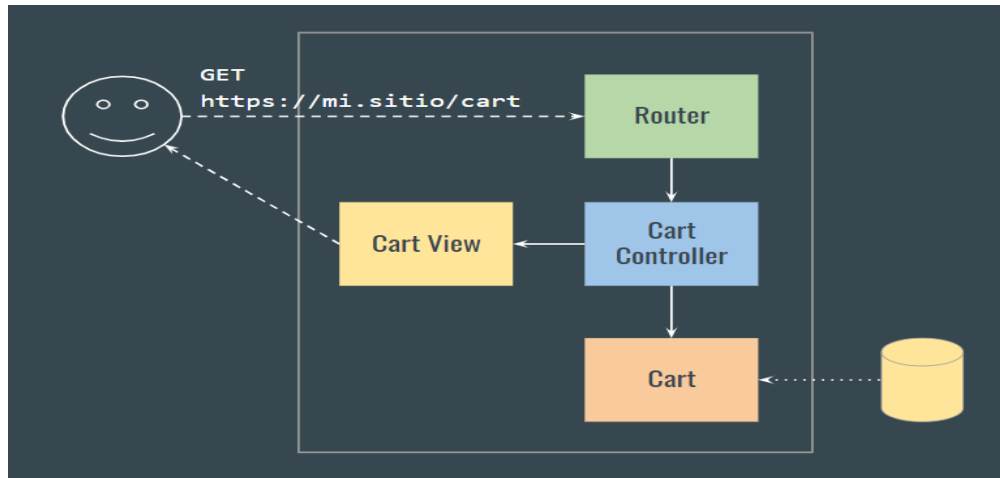


Figura 88. Eficiencia de ejecución 1. Autoría Propia.

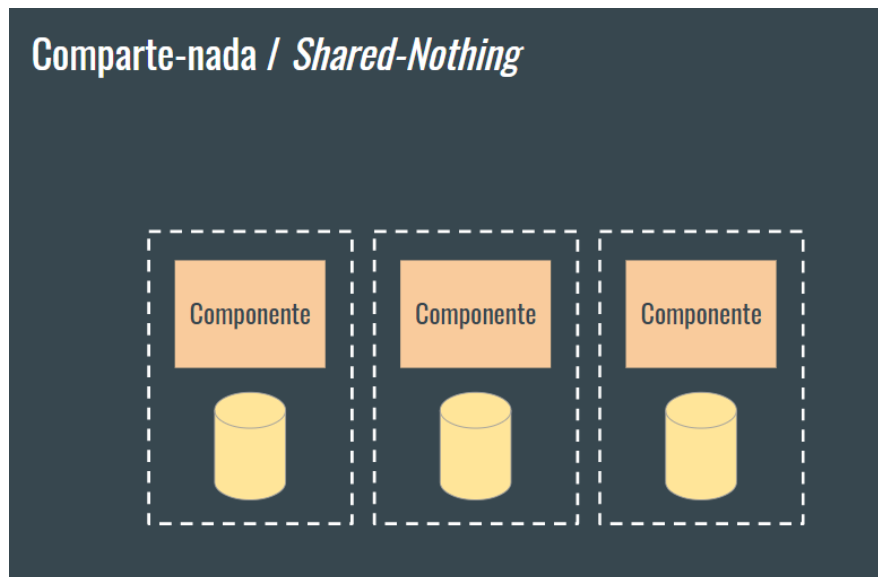


Figura 89. Eficiencia de ejecución 2. Autoría Propia.

Ahora sólo falta definir en el diseño de la arquitectura, las herramientas para su creación.

La primera pregunta es ¿Implementar un sistema ya realizado, adaptarlo a nuestras necesidades, o desarrollar desde cero uno nuevo?

Ésta es una de las preguntas que se hacen la mayor parte del tiempo, y es que para poder definir esto, se puede contestar siguiendo la siguiente pregunta.

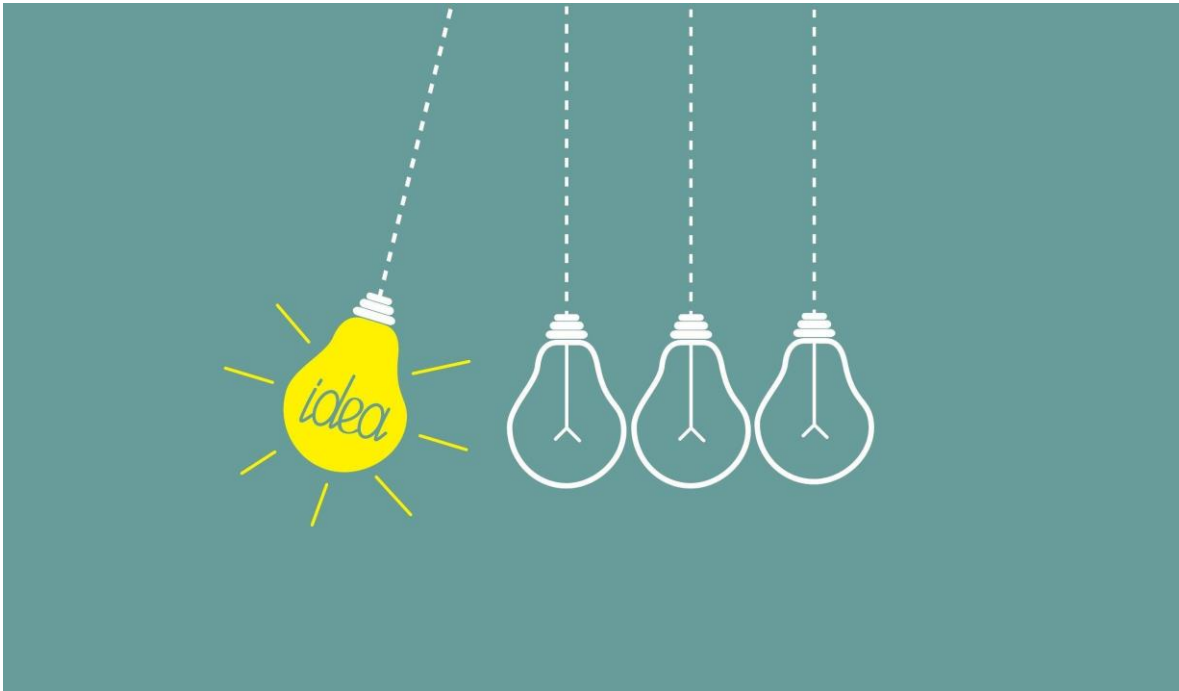


Figura 89. Definir ideas. Autoría Propia.

¿El proyecto es urgente?

Si es el caso, entonces deberemos por practicidad tomar un sistema que ya esté realizado, pongamos un ejemplo: **Wordpress, OctoberCMS, Prestashop**. Cualesquiera de estas soluciones son factibles, pero si la respuesta fue que no es urgente, y además de esto, la solución que se necesita requiere cierto nivel de personalización ya sea en la arquitectura de la base de datos o de la lógica de negocio, entonces una solución desarrollada desde cero es la opción más válida.



Figura 91. Desarrollo desde Cero. Autoría Propia.

Después del análisis se determinó que el sistema se desarrollaría desde cero.

5.2 Diseño

Para la interacción entre el sistema y los usuarios que lo usarán, decidimos enfocarnos en dos cosas.

La primera fue el diseño del producto digital, la User Interface.

Para ello decidimos utilizar una tendencia actual propuesta por Google, **Material Design**.

¿Qué es Material Design?

Material Design nace como una solución al problema constante de desarrollo interfaces de usuario para los dispositivos con sistema operativo Android, pues hasta antes de su versión 5, Se proporcionaba un libre diseño para el developer de las apps, esto daba como resultado muchas apps poco intuitivas y difíciles de usar para usuario. Debido a esto, Google tomó la decisión de crear guidelines sobre los cuáles los developers pudieran basarse y así mitigar el problema, tal como Apple cuenta con su Human UI.

Así Google creó el concepto de Material Design, el cual es un guideline enfocado en el desarrollo de interfaces, las cuales tenían como característica ser una abstracción de la realidad, es decir estos tenían propiedades que recuerdan cómo un usuario interactúa con elementos de la vida real, toma mucho en cuenta cosas como superficies, sombras, etc.



Figura 92. Material Design. www.material.io.

Una vez terminado el UI, se inició el UX.

User Experience

El User Experience es la facilidad con la que una persona puede interactuar con un sistema, UX por sí sólo es un mundo completo, pero nosotros sólo nos enfocamos en el UX del producto digital, es decir la UI.

Nos enfocamos en 3 cosas principalmente.

- Empty State: Cuando no haya información disponible, el usuario lo reconozca al instante y sepa que puede hacer una acción como crear información, o que la información que está buscando aún no ha sido llenada.
- Onboarding: Que el usuario de forma natural entienda cómo hacer las actividades que tenga disponibles en la interface, y si alguno no es lo suficientemente clara por sí misma, el sistema mismo sea el guía. Y **si** funcionalidad no está disponible, también sepa porqué.
- On Success: Cuando el usuario realice una acción en el sistema, sepa que ha ocurrido algo, y cuál fue el resultado de dicha acción.

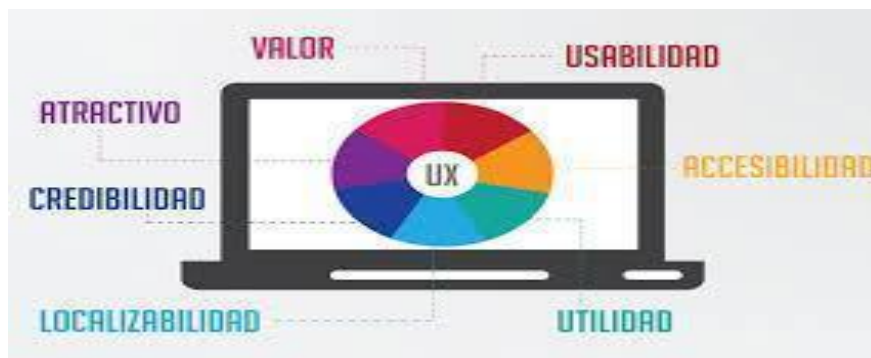


Figura 93. User Experience. www.mdirector.com

Frontend

El frontend en el mundo del desarrollo de aplicaciones web, es el encargado de crear todas las interfaces que se encargan de conectar la información que quiere transmitirse, al usuario, y en el proceso hacerla atractiva y fácil de manejar para el público objetivo utilizará dicha aplicación.

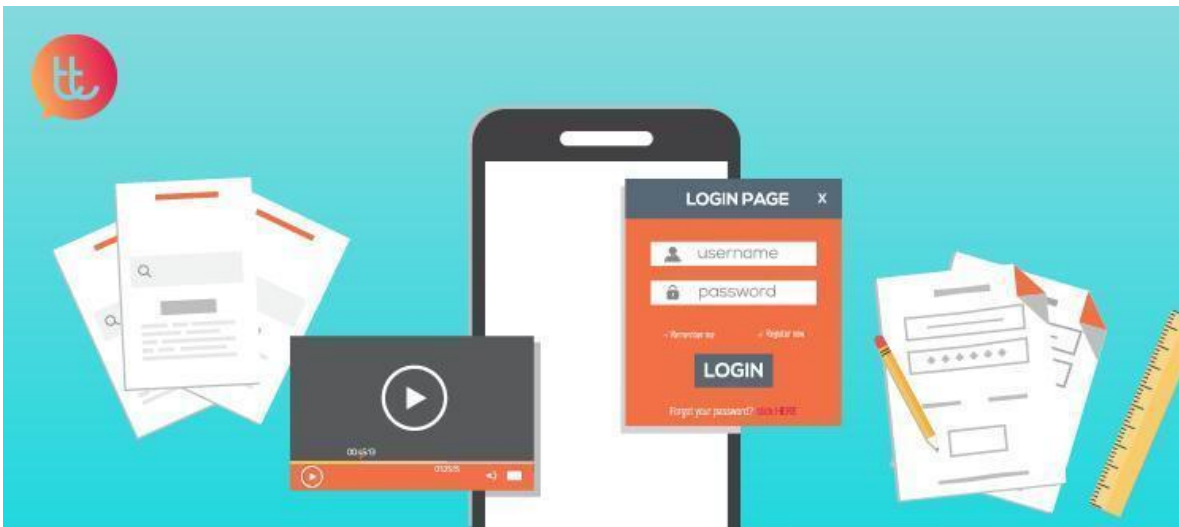


Figura 94. Frontend. www.mdirector.com

El Frontend es un vasto mundo que si bien, en los años 90 y 2000, donde sólo teníamos como conceptos de desarrollo Frontend a HTML, CSS y Javascript, actualmente cuenta con tantas herramientas y conceptos que para quién empieza en el mundo del Frontend puede darle una verdadera confusión.

Conceptos como preprocesadores, librerías y frameworks reactivos, pwa, seo, postprocesadores, web assembly y muchos más nos ayudan hoy en día a crear productos mucho más ricos e interactivos para simplificar y mejorar la vida de nuestros clientes o usuarios.

Para el lado del Frontend elegimos las herramientas a utilizar enfocados en el Desarrollo Ágil, es decir salir a Producción lo antes posible y entregando algo de valor para los usuarios, en este el día a día, caso, los profesores y administradores del **Congreso de Matemáticas**.

Para darle diseño a la interfaces propuesta en Material Design utilizamos una librería de css llamada **Vuetify**, la cual es una librería diseñada para el Framework **VueJS** que de forma muy rápida te permite estilizar la aplicación con diversos componentes como botones, cajas de texto, colores, y muchas características más

Imagen de prototipo usando Vuetify

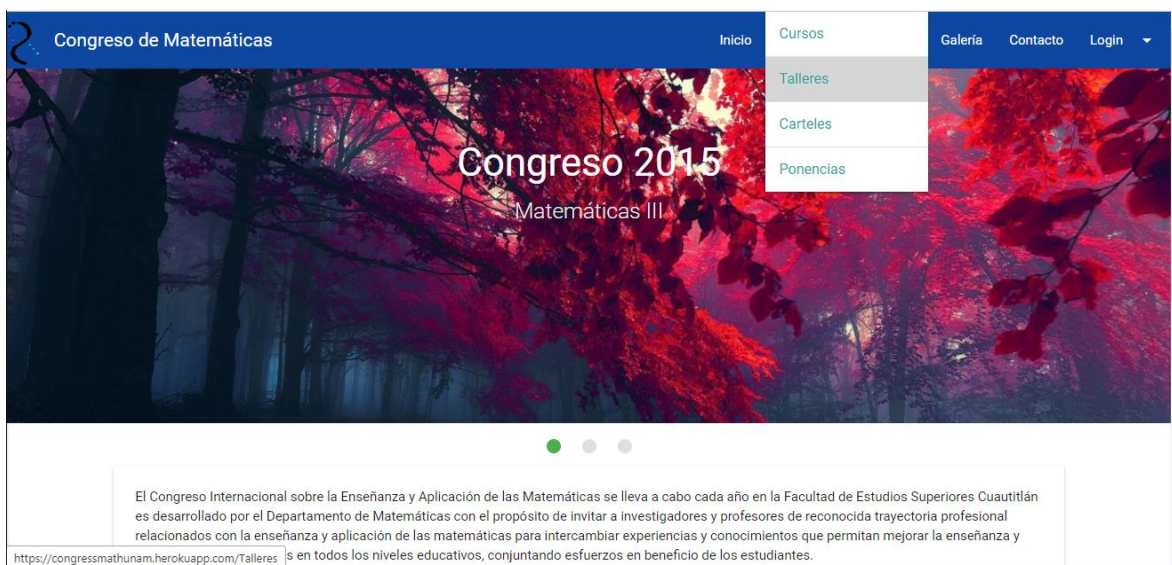


Figura 95. Usando Vuetify. Autoría Propia.

Vuetify también se puede utilizar con los preprocesadores css, los cuales no son más que software mayoritariamente realizados en lenguaje Ruby, que nos permiten extender las funcionalidades de css con conceptos de programación como variables, ciclos, condicionales. Nosotros no implementamos Preprocesadores para

esto, ya que con lo que **Vuetify** trae por default en su librería nos fue suficiente para cubrir los requerimientos.



Figura 96. Logo Vuetifyjs. vuetifyjs.com

Código de cómo se codifica en Vuetify:

```
1 <template>
2   <v-container>
3     <v-layout>
4       <v-layout
5         column
6       >
7         <v-flex
8           xs12
9           sm8
10          md6
11        >
12          <div class="text-xs-center">...
28        </div>
29        <v-card dark color="primary">
30          <v-card-title class="headline">¡Bienvenido!</v-card-title>
31        </v-card>
32
33        <v-card flat> ...
65      </v-card>
66
67    </v-flex>
68  </v-layout>
69 </v-container>
70 </template>
```

Figura 97. Código de Vuetifyjs. vuetifyjs.com

Para manejar la interacción completa del Frontend, usamos una librería llamada **VueJs** una de las 3 más populares en estos últimos 3 años para el desarrollo de aplicaciones web con alto nivel de interacción con el usuario.

VueJs tiene diferentes alcances, que puede desde sólo usarla a su mínima expresión solo en una sección o función de nuestro sistema, hasta ser VueJs el que maneje el sistema al cien por ciento.



Figura 98. Logo VueJs. vuetifyjs.com

En propuesta para este sistema, por los requerimientos y la arquitectura del Software, decidimos ocuparlo para ser el centro de nuestro **Frontend**, como arquitectura **Comparte-nada**.

Una de las funciones de Vuejs es la de mejorar la UX del usuario. Ya que en el sistema se hacen mucho CRUD (create, read, update, delete) de datos, por ello normalmente en una aplicación tradicional, al crear un nuevo elemento en base de datos el proceso suele ser:

Vista con Formulario -> Acción de CRUD -> Llamada a Servidor -> Procesa -> Recarga vista con los Datos y muestra información de éxito o error.

Esto a nivel de funcionalidad está bien, sin embargo, hacer que el sistema recargue cada vez que se hace una acción no sólo no es intuitivo, sino que a la larga se vuelve un proceso tedioso.



Figura 99. Experiencia VueJs. vuetifyjs.com

Por ello Vuejs nos ayuda a resolver esto, ya que con su sistema de Two-Way-Data-Binding nos permite refrescar los datos cada que se haga una acción de forma asíncrona (es decir sin recargar la página), esto lo hacemos complementando Vue con otra librería pequeña de Javascript llamada Axios, la cual nos permite ejecutar peticiones Http y de esa forma conectar nuestro frontend interactivo, con el Backend del sistema, del cual hablaremos más adelante a detalle, pero a grandes rasgos es el encargado de que el sistema tenga verdadera funcionalidad, y no sólo una bonita vista, que de eso se encarga el Frontend.

```
1 <template> ...
126
127 <script>
128
129 export default {
130
131   data(){
132     return {
133       dateString: '',
134       dateStringEng: '',
135       dateFormat: dateHandler.formatDate(new Date()),
136       metaTitle: document.getElementsByTagName('title')[0],
137     }
138   },
139   created(){
140     this.metaTitle.innerHTML = "Congress Test Component"
141   },
142   mounted(){
143   },
144   },
145   updated(){ ...
147   },
148   methods: { ...
268   }
269 }
270 </script>
271
272
```

Figura 99. Código VueJs-1. Autoría Propia.

Ejemplo de Componente en Vue

```
40 export default axios.create({
41   baseURL: API
42 })
```

Figura 100. Código VueJs-2. Autoría Propia.

```

327 | async getWorkShops(){
328 |   const self = this
329 |   if (!this.selectedUser || this.selectedUser === null) return 0
330 |   const user = await back.get(`chat/getWorkshops/${this.selectedUser.SESSION_ID}`)
331 |
332 |   if(this.selectedUser.NOMBRE){
333 |     if(this.selectedUser.NOMBRE.includes("Unspecified")) this.selectedUser.NOMBRE = "Usuario Web"
334 |   }
335 |
336 |   if(user.data.length != this.mensajes.length){
337 |     setTimeout(() => {
338 |       self.$scrollTo('#mensaje-final', 500, {container: '#lista-mensajes', easing: 'ease-in-out'})
339 |     },500)
340 |   }
341 |
342 |   this.mensajes = user.data
343 |   self.getWorkShops()
344 | },

```

Figura 101. Código VueJs-3. Autoría Propia.

5.3 Base de Datos

En la etapa de análisis de la base de datos determinamos que la relación de los datos será crucial, entonces lo primero que se definió, fue utilizar una base de datos relacional.

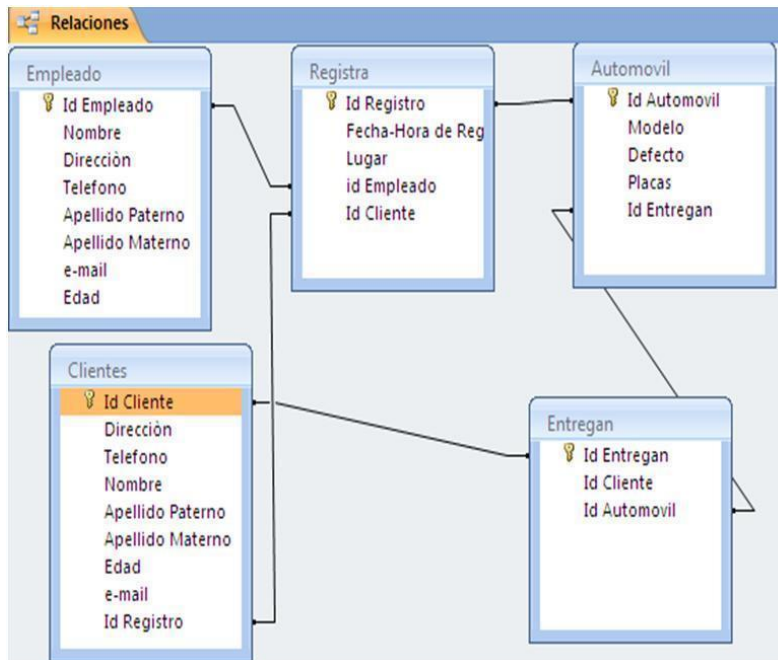


Figura 102. Base de Datos Relacional. Autoría Propia.

Y utilizar el DBMS (Database Management System) Mysql, ya que éste gestor de bases de datos es de libre uso, sencillo de manejar y eficiente para las bases de datos.



Figura 103. DBMS. www.solocube.com.

Al momento de elegir el motor que manejaría nuestra Base de datos determinamos el uso de InnoDB ya que ésta permite un mejor rendimiento de escritura y tiene una mejor optimización para la relación de llaves primarias y foráneas frente a otros motores como por ejemplo el popular MyISAM.



Figura 104. MyISAM & InnoDB. joewp.com.

Una vez establecidas las tecnologías a utilizar en la base de datos, se inició la etapa del modelado de la base de datos.

El modelado de la Base de datos se realizó con una herramienta Open Source llamada **MySQL Workbench**, En donde se puede tranquilamente diseñar la base de datos y tiene la posibilidad de exportar directamente el .sql



Figura 105. MySQL Workbench. dev.mysql.com

El diagramado dio como resultado varias tablas relacionadas entre sí que después se consumirá con el Backend y el backend le mandará la información al Frontend que es quien lo verá.

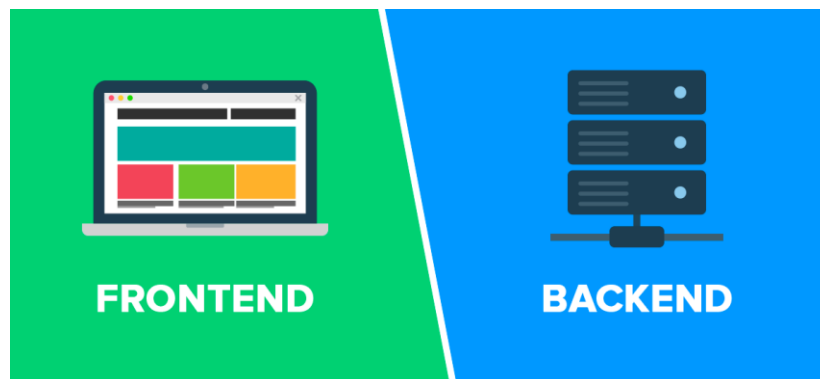


Figura 106. Frontend Backend. www.devcode.la

API

Una API (Application Programming Interface) es una parte en la arquitectura de software que se encarga de comunicar datos de acuerdo a las necesidades de un software.

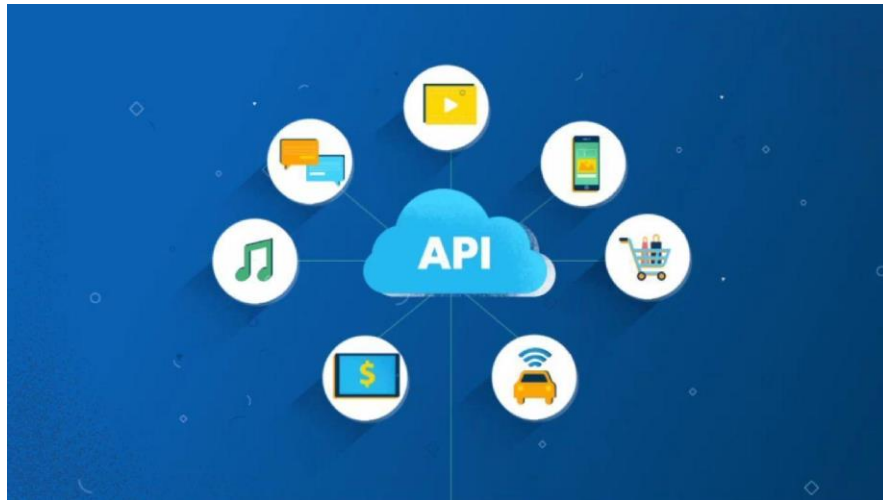


Figura 107. API. Autoría Propia.

Se propone desarrollar una mini API, con el propósito de que nuestro Frontend que mencionamos antes en algunas partes de nuestra aplicación donde necesitaba realizar muchas funciones del tipo CRUD, utilizando Vue para procesarlas, por ello la API nos permite enviar y recibir la información entre el Front y el Back, para poder hacer verdadera funcionalidad asíncrona.

La API propuesta es sencilla ya que sólo consiste en verificar en los encabezados de la petición si existe una autenticación por parte del usuario quién está efectuando esta petición.

Si esto es verdad, entonces leemos que tipo de petición es, nosotros ocupamos los 4 métodos HTTP básicos, los cuáles son **GET, POST, PUT y DELETE.**

GET	/pet/{petId}	Find pet by ID
PUT	/pet	Update an existing pet
DELETE	/pet/{petId}	Deletes a pet
POST	/pet/{petId}/uploadImage	uploads an image

Figura 108. Métodos HTTP. Autoría Propia.

Backend

Para el desarrollo se propone utilizar el lenguaje de programación **Php** en su versión **7**. Las razones para elegir este lenguaje sobre otros como NodeJs, Python, Ruby, Go, ASP, etc. Fueron:

- Lenguaje amigable que garantiza la mantenibilidad del código
- Lenguaje Open Source y gratuito sin restricciones
- Fácil integración a los servidores de la escuela
- Fácil integración con la Base de datos MySql



Figura 109. PHP Versión 7. www.php.net

Aunado a ello, también seleccionamos el Framework de Php **CodeIgniter** en su versión **3**, esto debido a que de ésta forma logramos un desarrollo ágil pues nos facilita cosas como:

- Refuerzo en Seguridad como:

- MySql Injection
 - Xss
 - Reflect Attack
 - DDOS
- Estructura de archivos ordenada en MVC(Model-View-Controller)
 - Manejo de sesiones, cookies y peticiones
 - Query Builder: Para facilitar las consultas con la Base de datos



Figura 110. CodeIgniter. codeigniter.com.

En un principio tuvimos en cuenta otros Frameworks que ofrecían los mismos beneficios como:

- Laravel
- Simphony

Pero al final elegimos CodeIgniter por ser el más minimalista, ya que en nuestro alcance no necesitamos un framework excesivamente robusto.

La estructura de CodeIgniter es simple bajo el modelo MVC, sin embargo, en nuestra propuesta lo usamos tanto de MVC como de Rest API.

The image shows a mobile application interface for 'Congreso de Matemáticas'. At the top, there is a blue header with a hamburger menu icon, a logo, and the text 'Congreso de Matemáticas'. Below this is a blue banner with the word 'Contacto' in white. Underneath the banner is a yellow bar with the text 'Teléfono:'. The main content area is white and contains a form titled 'Formulario:'. The form has four input fields: 'Nombre', 'Email', 'Asunto', and 'Mensaje'. At the bottom of the form is a yellow button with the text 'ENVIAR' and a right-pointing arrow. The bottom of the screen shows a blue bar with the text 'Congreso de Matemáticas'.

Figura 111. MVC API. Autoría Propia.

5.4 Seguridad

En el desarrollo del proyecto se consideran distintos mecanismos de seguridad para evitar el robo de información, la interrupción del sistema, así como comprometer su correcto funcionamiento.

CORS

Para prevenir que sitios web distintos al Frontend puntual que hará la petición intenten obtener información del sistema, se implementa el funcionamiento de CORS (Cross Origin Request Source) el cual es deniega el acceso a los endpoints del backend en caso de que el origen de un HTTP Request sea distinto del que nosotros puntualmente configuramos (La Página del Congreso).

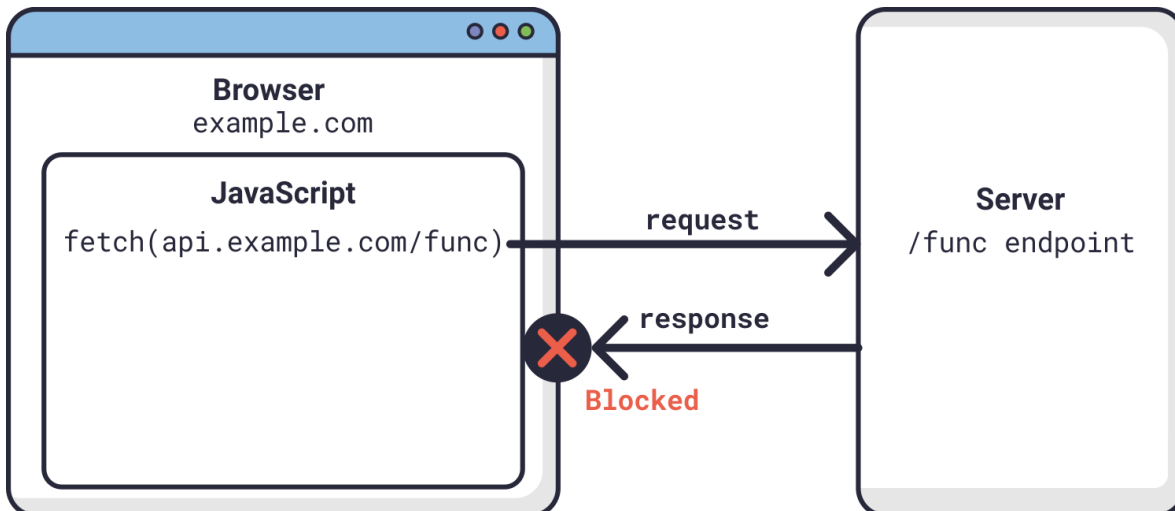


Figura 112. CORS. [Haproxy.com](https://haproxy.com).

JWT

JSON Web Tokens o abreviado como JWT, es un mecanismo de autenticación con el cual seremos capaces de validar desde el Frontend, que quién está haciendo ciertas acciones como dar de alta alumnos en cierto taller, tiene los permisos para hacerlo. Ya que actualmente la forma de desarrollar Aplicaciones Web es Backend For Frontend, esto quiere decir el Frontend y Backend están desacoplados, al no usar técnicas como Cookies o sesiones, utilizamos JWT como técnica de inicio de sesión.

Los JSON Web Tokens (JWT) son un estándar que nos permite representar los permisos o requerimientos entre dos partes, es decir, el servidor de nuestra aplicación concede los permisos para un usuario generando un token, este token se envía al cliente o navegador donde utilizamos el token para realizar peticiones al servidor sin que éste necesite verificar a los usuarios, simplemente recibe los JWT y devuelve la información a la que el usuario tiene permiso según cada token.

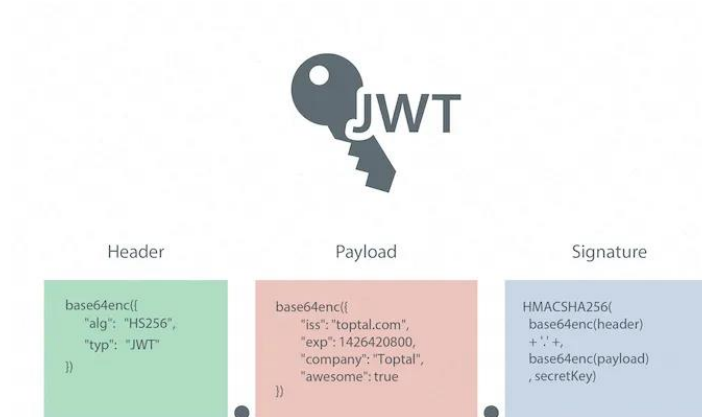


Figura 113. JWT. <https://ffontanesf.es>.

Sanitizadores

En el sistema algo muy común será realizar escrituras en Base de datos, y los datos vendrán desde el Cliente, es decir el Frontend, aunque nosotros consideramos mecanismos de limpieza para enviar los datos del Frontend a la parte del Backend de la forma más íntegra posible, en el cliente siempre es posible manipular lo que se envía, para prevenir ataques de tipo XSS(Cross Site Scripting) del lado del Backend se realiza la implementación de sanitizadores de datos con el fin de validar que la data que será guardada en Base de datos sea lo más íntegra posible, utilizando las librerías que ya tiene el framework Codeigniter se realizan limpiezas de XSS eliminando todo código Html y Javascript que pudiera existir en un campo de por ejemplo “nombre”.

Además de ello se hacen validaciones de tipos de datos para comprobar que un correo sea en realidad un correo (con patrones de RegEXP), un teléfono sea un teléfono y cada caso en particular tenga un correcto formato.

Prevención de Ataques por fuerza bruta

Hay ocasiones que se pueden ejecutar intentos de inicio de sesión desde el navegador de forma automatizada, evadiendo el mecanismo de CORS por medio de Aplicaciones desarrolladas por ejemplo con Puppeteer de NodeJS que permite tomar el control del navegador y de esa forma podrían intentar infinitamente combinaciones de usuario y contraseña para así dar con una clave.

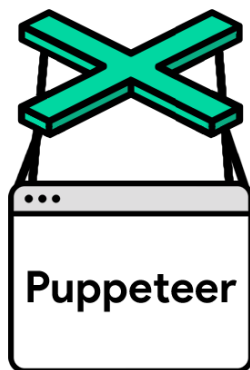


Figura 114. Puppeteer. <https://pptr.dev>.

Si bien en el código desarrollado se contempla una seguridad de passwords de hash 256, no puede evitarse en ocasiones que una contraseña débil creada por un usuario estudiante o un miembro del staffing del congreso aparezca. Por ello se implementará una funcionalidad en donde después de 5 intentos de inicio de sesión desde una dirección IP, se bloqueará por 2 horas el acceso al inicio de sesión, dejando al posible robot, en espera para seguir intentándolo.

Aunado a ello se introducirá el sistema de Google reCaptcha el cual es de uso gratuito y analiza el comportamiento de un usuario en el Frontend y así se detecta posible actividad sospechosa de navegación, empezará a solicitar que se resuelvan ciertos rompecabezas para comprobar que no es un robot.

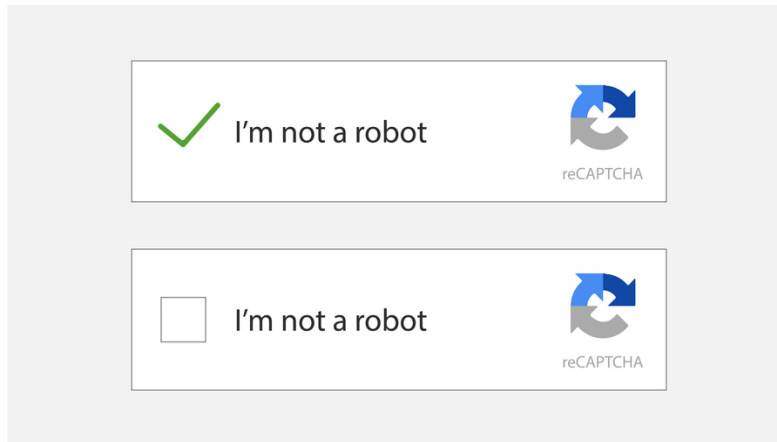


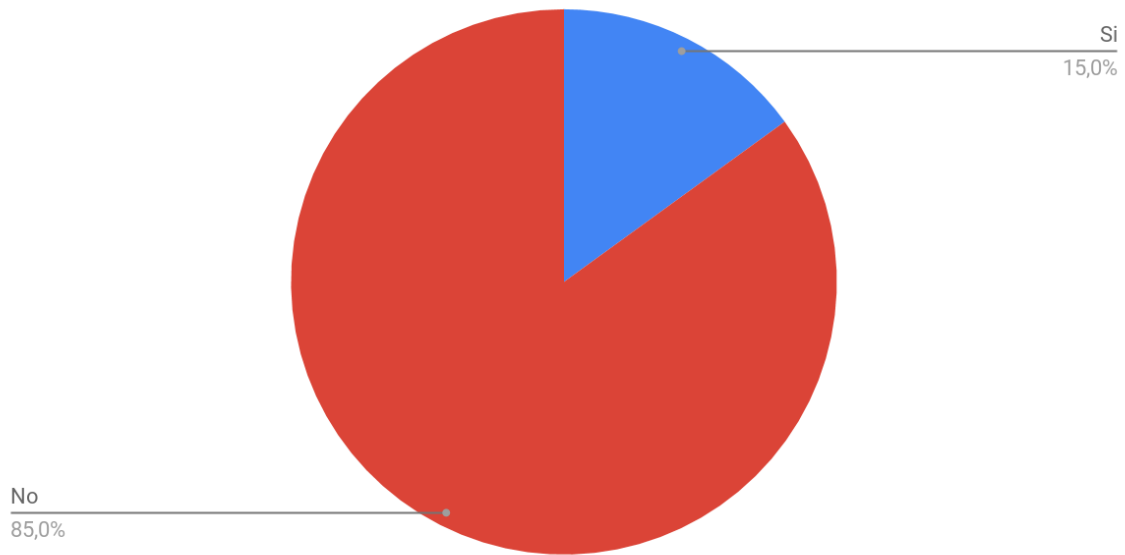
Figura 115. Recaptcha. <https://fossbytes.com>

5.5 Análisis de Resultados

Después de pasar por las diferentes etapas de la investigación, recopilar datos para la propuesta, se realizaron una serie de preguntas para obtener una serie de resultados que favorecieron la propuesta desarrollada a lo largo de este trabajo de investigación.

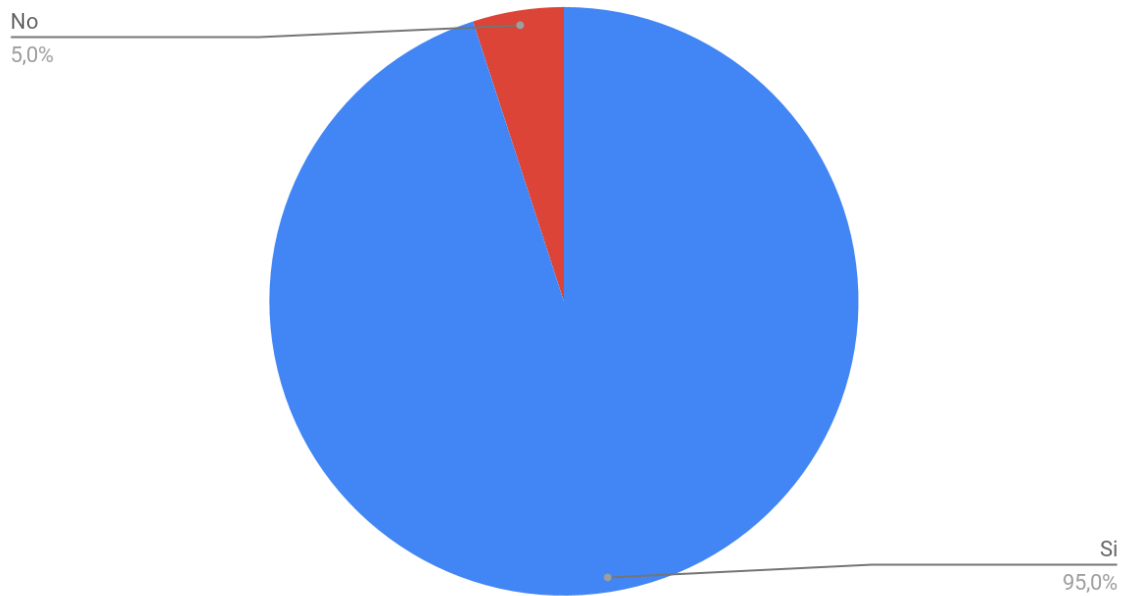
Lo cual nos lleva a las siguientes gráficas:

¿Sabías que existe una página web para el Congreso de Matemáticas?



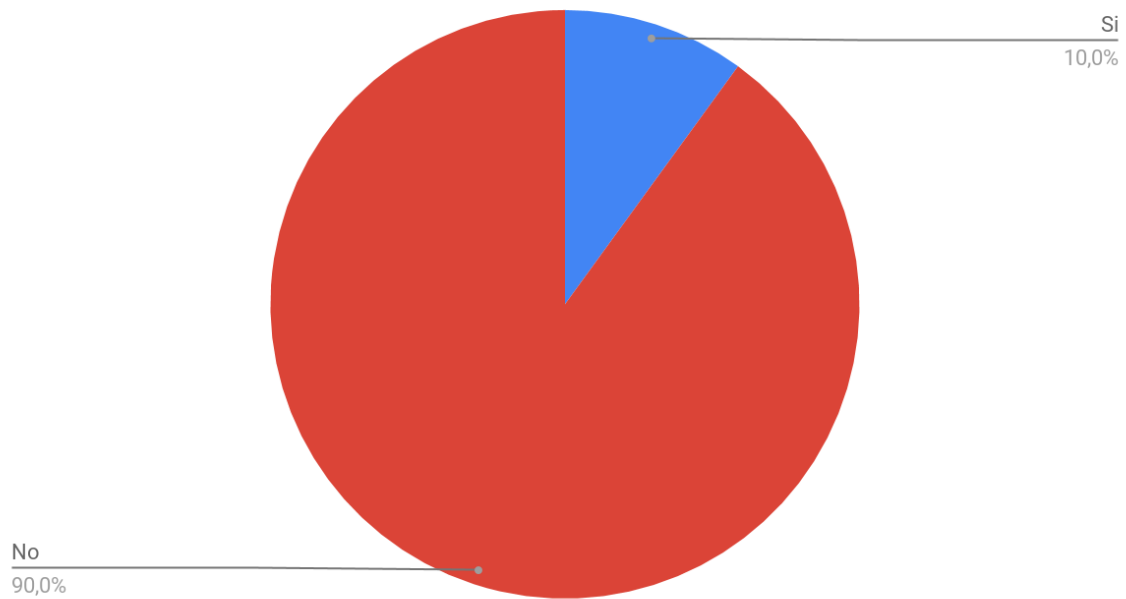
Gráfica 1. Resultado de encuesta. Autoría Propia.

¿Te gusto la interfaces de la Pagina Web?



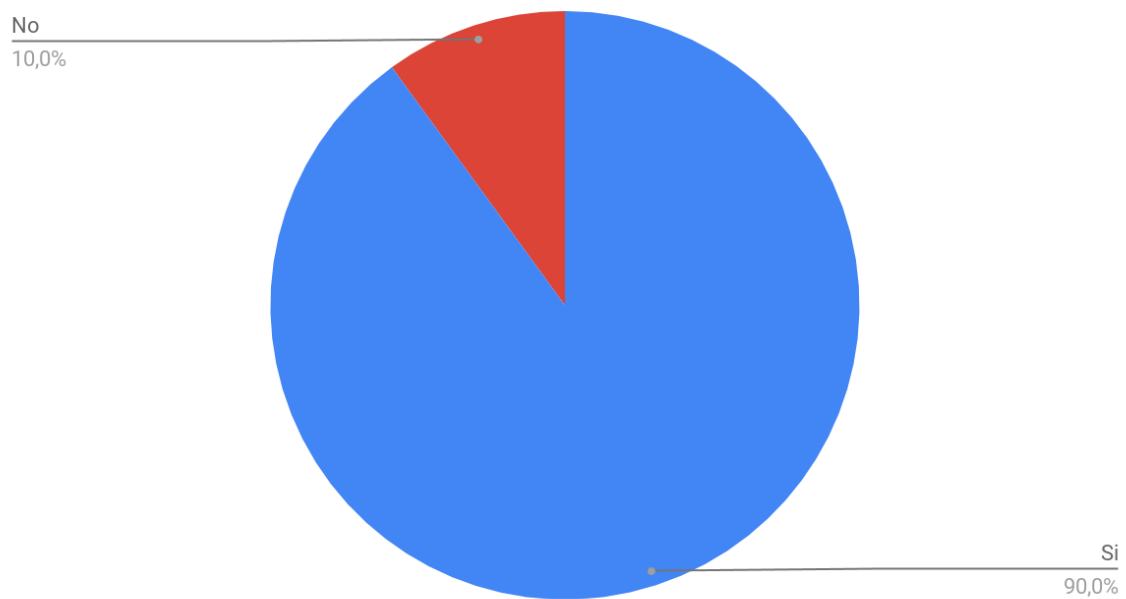
Gráfica 2. Resultado de encuesta. Autoría Propia.

¿Crees que la distribución del Contenido es adecuado?



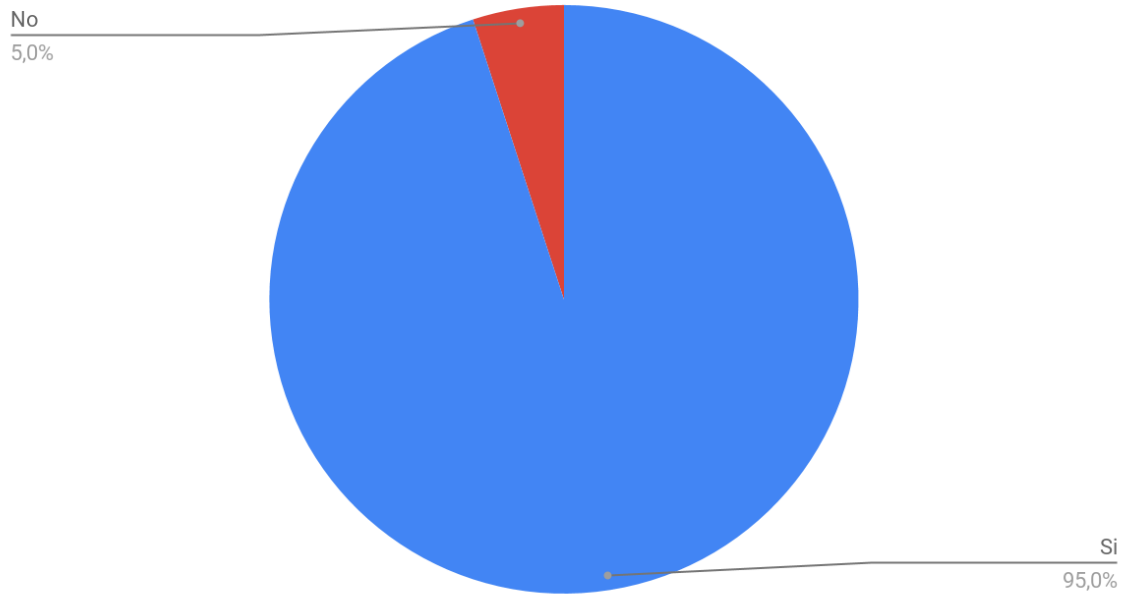
Gráfica 3. Resultado de encuesta. Autoría Propia.

¿Te gustaría que se generara una página web mas moderna?



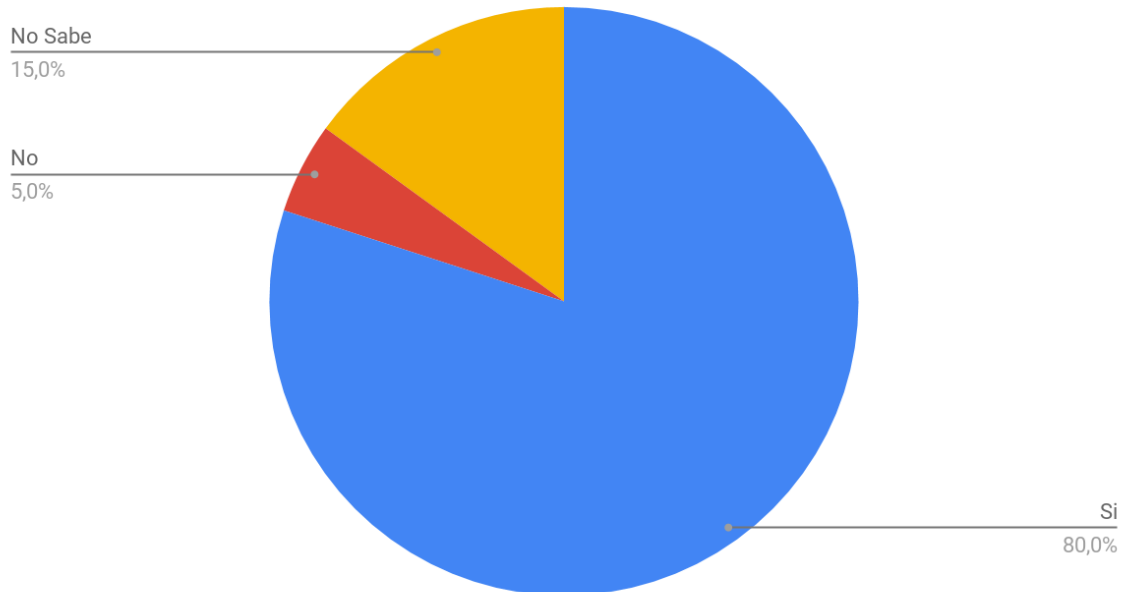
Gráfica 4. Resultado de encuesta. Autoría Propia.

¿Te gusto la propuesta de mejora de la página web que se mostró?



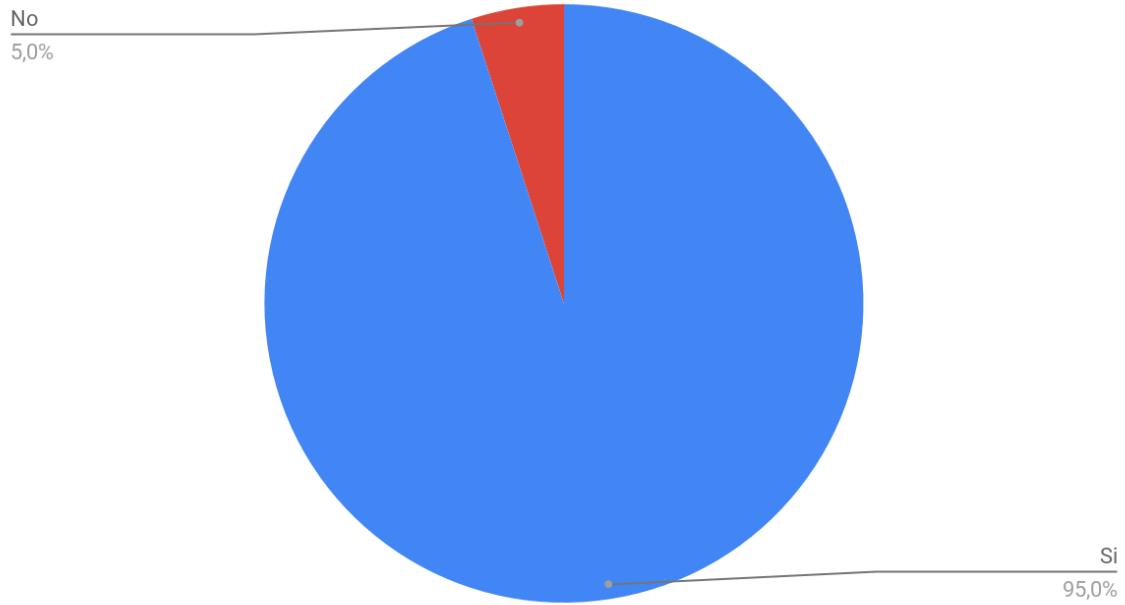
Gráfica 5. Resultado de encuesta. Autoría Propia.

¿Votarías a favor de la propuesta de mejora para la página web?



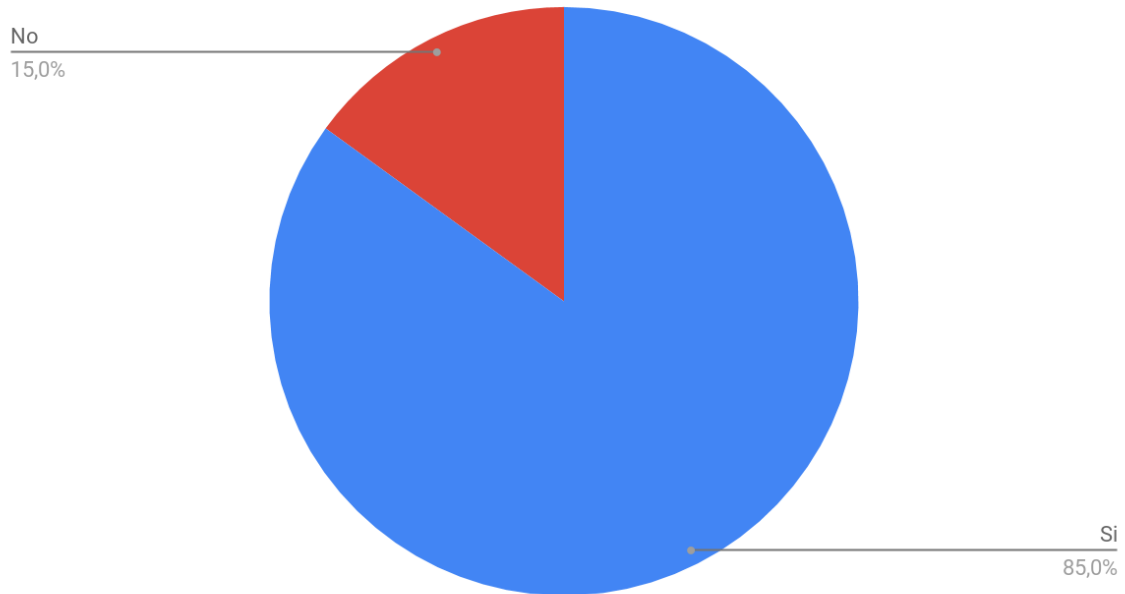
Gráfica 6. Resultado de encuesta. Autoría Propia.

¿Crees que la propuesta de mejora ayudaría a la navegación por la página web?



Gráfica 7. Resultado de encuesta. Autoría Propia.

¿Crees que el contenido de la propuesta esta bien distribuido?



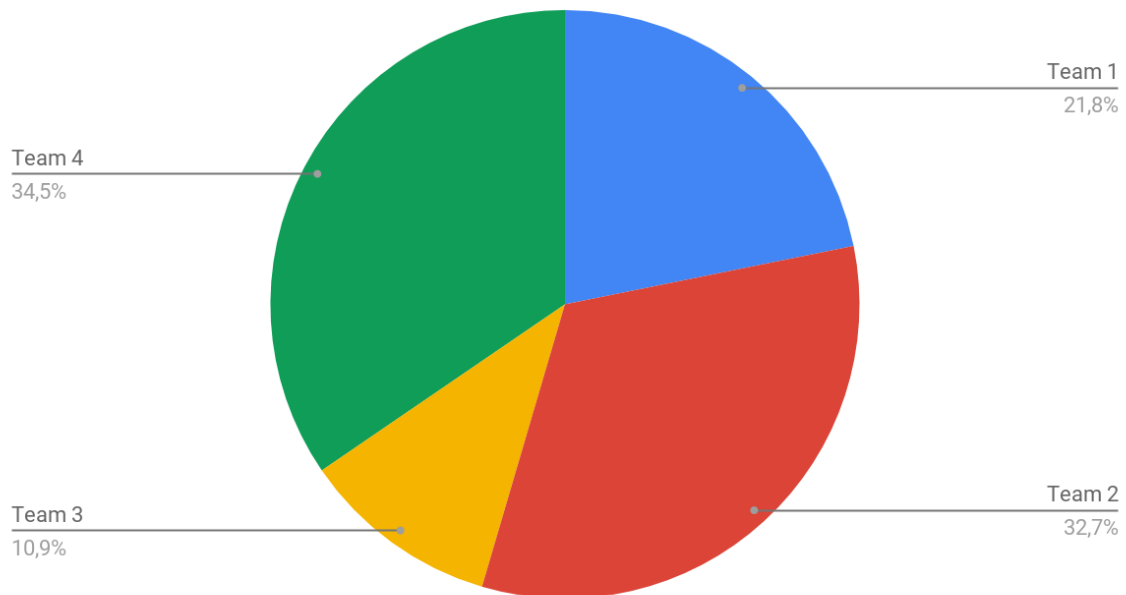
Gráfica 8. Resultado de encuesta. Autoría Propia.

¿Es importante que la página sea adecuada a cada dispositivo?



Gráfica 9. Resultado de encuesta. Autoría Propia.

Points scored



Gráfica 10. Resultado de encuesta. Autoría Propia.

6. CONCLUSIONES

En base a toda la información recopilada a lo largo de este proyecto de investigación que abarca la **“Propuesta de mejora para el sistema del Congreso de Matemáticas de la Facultad de Estudios Superiores Cuautitlán”**, se puede llegar a las siguientes conclusiones:

- Después de dar una meticulosa revisión al sistema actual con el que cuenta el La Facultad de Estudios Superiores Cuautitlán, en el área de Matemáticas, se encontraron áreas de oportunidad con la cual se puede trabajar para poder optimizar ciertos factores, para aumentar los siguientes puntos:
 - Interfaces de diseño
 - Mejor distribución del Contenido
 - Aumentar el tiempo de Sesión del usuario
 - Seguridad
 - Ahorro de Tareas Administrativas

Gracias a los estudios, pruebas y resultados obtenidos con la investigación, basada en la propuesta de mejora del sistema, se llega a la conclusión que la Propuesta de mejora para el sistema, es totalmente viable, segura y se aprovecharía al cien el sistema para poder reducir tareas, así como llevar un control más automatizados de los futuros congresos, facilitando cada actividad para el personal que se encuentre involucrado con el evento.

7. RECOMENDACIONES

Una parte que implica la propuesta de mejora del sistema del **Congreso Internacional sobre la enseñanza y aplicación de las matemáticas**, para poder optimizar al máximo este sistema y que se tengan los resultados adecuados, tomando en cuenta que pueden variar las actividades para cada congreso anual, por lo cual se plantean las siguientes recomendaciones:

- Se debe realizar un cronograma para plasmar las fechas de mantenimiento al sistema.

- Actualizaciones de información, esto con el fin de que se pueda promocionar las actividades que se tendrán, así como para los congresos futuros.

- Actualizar, modificar o borrar los registros de las bases de datos para evitar una saturación en el sistema de datos.

- Realizar planificación de interfaces para verificar si se necesita cambiar, agregar o quitar, según sea la necesidad de la Facultad de Estudios Superiores Cuautitlán, con anticipación para que no se vea afectado el tiempo de sesión de los usuarios en el sistema.

8. BIBLIOGRAFÍA

1. Sampieri, R. (2010). Metodología de la Investigación. México: McGraw Hill.
2. Kendall, E. & Kendall, J. (2005). Análisis y Diseño de Sistemas. México: Utograficaingr Amex SA
3. Corcoles E. & Montero F. (2012). Diseño de Interfaces. España: Ra-Ma Editorial, S.A.
4. Pressman, R. (2005). Ingeniería del software. Un enfoque práctico. México: McGraw Hill.
5. Behar, D. (2008). Metodología de la investigación. México: Shalom.
6. Corcoles Tintero J. Eduardo, Montero Simarro Francisco (2012). Diseño de Interfaces Web 1º edición. España: Ra-Ma Editorial, S.A.
7. Mercedes Marqués. (2011). Base de Datos. España: Castelló de la Plana.
8. Hueso Ibáñez Luis. (2012). Gestión de Base de Datos 2º Edición. España: Ra-Ma Editorial S. A.
9. Martínez López Francisco J. Gallegos Ruiz Alma. (2017). Programación de Bases Relacionales. España: Ra-Ma Editorial S.A.
10. Pavon Puertas Jacobo. Llanera Borges Ezequiel. (2015). Creación de un Sitio Web con PHP y MYSQL. España: Ra-Ma Editorial, S.