

**Universidad Autónoma Metropolitana Unidad
Azcapotzalco**

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

Sistema Gestor para la Universidad Obrera de México

Proyecto Tecnológico

Trimestre 2016 Primavera

Cesar Arostegui Ramírez

Matricula: 204305462

Asesores:

Hugo Pablo Leyva

Rafaela Blanca Silva López

Yo, Hugo Pablo Leyva declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Yo, Rafaela Blanca Silva López declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Yo, Cesar Arostegui Ramírez, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



ÍNDICE GENERAL

Contenido

ÍNDICE GENERAL.....	3
ÍNDICE DE IMÁGENES.....	4
DEDICATORIA	6
ANTECEDENTES	6
INTRODUCCIÓN.....	8
RESUMEN	9
OBJETIVOS.....	11
Objetivo General	11
Objetivos Particulares	11
JUSTIFICACIÓN.	12
MARCO TEORICO.....	13
ESPECIFICACIÓN TÉCNICA	13
NUESTRAS HERRAMIENTAS.....	13
1. WAMP SERVER	13
2. WORKBENCH	15
3. JAVASCRIPT	16
ESTILOS CSS	16
DESARROLLO DEL PROYECTO.....	17
DISEÑO DE LOS MÓDULOS.....	17
CONTROL DE ACCESO.....	18
ESPACIOS FÍSICOS.....	20
ENCUESTAS.....	22
REPORTES.....	23
DISEÑO DE LA BASE DE DATOS.....	25
DIAGRAMA ENTIDAD-RELACION	25
CREANDO LA BASE DE DATOS	26
IMPLEMENTACIÓN DE LOS MÓDULOS.....	26
EL HTML Y NOTEPAd++	26
ACCESO AL SERVIDOR LOCAL	27
CREANDO EL MODULO DE CONTROL DE ACCESO.....	27

MODULO ESPACIOS FÍSICOS.....	33
ENCUESTAS.....	38
REPORTES.....	41
DETALLANDO LA INTERFAZ	42
ESTILO A EL SISTEMA.....	42
APOYO DE JAVASCRIPT.....	42
LAS CLASES CSS.....	43
VISUALIZACIÓN DE LOS RESULTADOS DE LAS ENCUESTAS	44
SIMULACIONES.....	45
REALIZANDO PRIMERAS PRUEBAS	45
REPORTE DE ERRORES.....	45
RESULTADOS	46
CONCLUSIONES	47
BIBLIOGRAFÍA.....	48
Libros.....	48
Páginas Web.....	48
APÉNDICES	49
APÉNDICE 1. Instalación de WAMP SERVER.....	49
APÉNDICE 2. Tablas de creación de MySQL	54

ÍNDICE DE IMÁGENES

Ilustración 1. Arquitectura del Sistema.....	13
Ilustración 2. Módulos del Sistema en la propuesta.....	17
Ilustración 3. Módulos del Sistema Actual.....	17
Ilustración 4.Caso de uso, Módulo de Control de Acceso.....	18
Ilustración 5. Diagrama de clases de Control de Acceso.....	19
Ilustración 6. Caso de uso, Espacios Físicos.	20
Ilustración 7.Diagrama de clases, Espacios Físicos.....	21
Ilustración 8. Caso de uso, Espacios Físicos incluyendo al Asistente.	21
Ilustración 9.Caso de uso, Encuestas.	22
Ilustración 10.Diagrama de Clases, Encuestas.	22
Ilustración 11.Casos de uso Reportes.	23
Ilustración 12.Caso de uso General del sistema.	24
Ilustración 13.Primer Diagrama Entidad-Relación.	25
Ilustración 14.Diagrama entidad-relación Final.....	26
Ilustración 15.Dirección de alojamiento en el servidor.	27
Ilustración 16.Dirección del servidor.....	27

Ilustración 17. Diagrama entidad-relación para Control de acceso.....	28
Ilustración 18. Ventana del panel inicial, index.php	28
Ilustración 19. Ventana de registro	29
Ilustración 20. Se muestra como se verifica que el apartado email tenga @.....	29
Ilustración 21. Pantallas de ingreso de cada uno de los tipos de usuarios.....	30
Ilustración 22. Los paneles que mostraría a ingresar cada uno de los tipos de usuario.....	30
Ilustración 23. Comparación de los diferentes métodos para hacer la conexión a la base de datos.....	31
Ilustración 24. Menú de administración	32
Ilustración 25. Menu de Administración.....	33
Ilustración 26. Diagrama Entidad-relación de las tablas a utilizar en el módulo de espacios físicos.	34
Ilustración 27. Menú donde se administran los espacios Físicos.....	34
Ilustración 28. Relación de tablas espacios físicos y tipo de espacios físicos.	35
Ilustración 29. Proceso de solicitud de espacio físico.	35
Ilustración 30. Mostrar Solicitudes.	36
Ilustración 31. Página pide actualización para generar horario.....	37
Ilustración 32. Tabala de horarios libres.	38
Ilustración 33. Diagrama entidad-relación para el módulo de encuestas.	39
Ilustración 34. Proceso de la creación de una nueva encuesta.	39
Ilustración 35. Proceso que sigue el alumno al contestar la encuesta.	40
Ilustración 36. Panel de administración de reportes.	41
Ilustración 37. Comparación de antes y después de los estilos CSS.	42
Ilustración 38. Script de las alertas.	42
Ilustración 39. Script que nos permite navegar en la misma ventana.....	43
Ilustración 40. Clase creada wrap y sus variantes.....	44
Ilustración 41. Así se mostraran los resultados de las encuestas.	45

DEDICATORIA

La finalización de este proyecto es una parte importante en mi vida por que culmina un ciclo de esfuerzo en terminar un largo recorrido para lograr el objetivo de culminar mi carrera universitaria, pero al final no lo hubiera logrado sin el apoyo de mi familia y el consejo de mis maestros por eso quiero agradecerles y dedicarles mi trabajo.

A mi madre Lupita.

Por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, por estar siempre pendiente de mí y haber enseñado el valor de la perseverancia pero más que nada, por su amor incondicional.

A mi novia Martha.

Por el apoyo que me dio desde el primer momento que decidimos caminar hacia el futuro juntos, el estar motivándome a seguir adelante y nunca rendirme, por estar en todo momento complicado, cuando las cosas se ponían difíciles ella siempre me dio palabras de aliento para concluir esta etapa de mi vida. Y le dedico esto sobretodo porque la amo con toda el alma.

P,G,M,C,D.

A mi familia

A mi familia que creyó en mí y que siempre confió en lograría terminar mi carrera, pero sobre todo por estar respaldándome siempre, y a mis abuelitos Angelina y Chuchito y mi tío Mario que desde el cielo sabrán que este esfuerzo lo hice también por ustedes.

A mis amigos.

Que fueron parte importante al hacer interesante mi vida universitaria y que me hicieron sentir una segunda familia en la UAM. Daniel, Mimí, Jacob, Aatzin, Montse gracias por ser mis amigos.

A mis maestros.

A todos ellos que me transmitieron sus conocimientos y me superaron inculcar ese amor por la carrera y que me hicieron superarme día con día. Pero además quiero mencionar a ciertos profesores que sin darme una sola clase me enseñaron muchas cosas que en mi vida profesional me han sido de mucho: La maestra Margarita Portilla, el Dr. Emilio Sordo, y el Dr. Alejandro Pérez Ricardo.

A mis Asesores.

Sin duda también a ellos porque sin su guía esto no hubiera sido posible, al maestro Hugo Pablo por su apoyo y por qué en sus clases aprendí muchas cosas que en este proyecto aplique gracias por to ese apoyo y también a la maestra Rafaela Blanca quien desde que tuve oportunidad de conocer creyó en mí y me dio todo su apoyo.

Y agradezco a mucha gente más que no mencione aquí pero sé que fueron y serán importantes en mi vida por siempre.

ANTECEDENTES

La Universidad Obrera de México (UOM) [1] con 79 años de historia se ha mantenido como una alternativa de conocimiento para la clase trabajadora ofreciendo diversos cursos, talleres y diplomados con diferente temática. Ha logrado permanecer por tanto tiempo debido a que ha ido evolucionando y adaptando a las necesidades de la sociedad y cada vez tiene una mayor matrícula de alumnos entre sus cursos, talleres y diplomados, por lo que conocer la opinión y sugerencias de los mismos son de suma importancia para seguir fortaleciendo la oferta educativa de la UOM. Pero también la UOM se dedica a ofrecer una difusión cultural y social por lo cual a lo largo del año realiza exposiciones, congresos, conferencias, entre otros. Todos estos eventos tanto internos como externos requieren de una administración y esto sin duda es una carga de trabajo que consume tiempo.

Por lo anterior planteamos el desarrollo de un Sistema que apoye en la gestión de los espacios físicos con los que cuenta la Universidad y también con el monitoreo de la satisfacción que tiene los alumnos. Además, con la facilidad de poder hacer reportes.

El Sistema para la gestión de espacios será muy útil en la UOM debido a que todo el trabajo que se realiza actualmente se hace de forma manual y solo utilizan programas como Excel como apoyo, por lo cual genera muchas dificultades a la hora de hacer búsquedas y actualizar la información, generando problemas con la logística y distribución de los espacios. El sistema en la parte de la gestión de espacios físicos tendrá su principal función de hacer este trabajo más rápido y tener un mayor control de esta actividad, desde que se hace una reservación de un espacio, hasta la de consultar la disponibilidad de los mismos y que además se puedan generar reportes. Para esto el sistema debe contar con una base de datos donde se almacene la información.

La otra parte del Sistema servirá para monitorear la satisfacción de los estudiantes por medio de encuestas y con ello se pueda identificar donde existen problemas, donde se debe mejorar y cuales actividades se están haciendo bien. El módulo deberá tener la facilidad de adaptarse a varios tipos de encuesta y como agrupar los resultados, la interpretación de estos quedara a cargo de la persona correspondiente.

INTRODUCCIÓN

El presente proyecto, se llevó a cabo a través de un plan de trabajo que nos llevaba a cumplir objetivos a corto plazo lo cual nos proporcionó la facilidad para su desarrollo. El desarrollo del proyecto, nos permitió aplicar los conocimientos adquiridos durante la estancia en la Universidad Autónoma Metropolitana, pero también nos brindó la experiencia de adquirir conocimientos nuevos.

En estos tiempos parecería imposible no encontrar ventajas en la implementación de sistemas informáticos y similares en organizaciones, empresas, instituciones, etc., con los cuales el trabajo puede ser más rápido y con mayor control, y es por eso que se dio a la tarea de implementar un sistema informático que resolviera un problemática en la vida cotidiana.

El proyecto se enfoca principalmente en resolver el problema de gestión que tiene la Universidad Obrera de México para distribuir sus espacios físicos y más que nada para llevar un buen orden en sus reservaciones y asignaciones, ya que esto les ha traído problemas como empalmes de eventos en un mismo espacio, además que en la diversidad de eventos provoca que no exista un control real entre los eventos externos y los internos.

Pero también se tratara de dar una herramienta que permita tener en cuenta la opinión de los alumnos con respecto a los cursos y también hacia los profesores, con el fin de tener áreas de oportunidad, tanto para mejorar como para resolver fallas que se estén presentado.

Así que en conjunto nuestro proyecto será realizar un Sistema Gestor que permite no solo administrar espacios físicos, si no también tener la posibilidad de recaudar información por medio de encuestas de la opinión de los alumnos de la UOM.

RESUMEN

El presente proyecto, se llevó a cabo a través de un plan de trabajo que nos llevaba a cumplir objetivos a corto plazo lo cual nos proporcionó la facilidad para su desarrollo. El desarrollo del proyecto, nos permitió aplicar los conocimientos adquiridos durante la estancia en la Universidad Autónoma Metropolitana, pero también nos brindó la experiencia de adquirir conocimientos nuevos.

El proyecto se enfoca principalmente en resolver el problema de gestión que tiene la Universidad Obrera de México para distribuir sus espacios físicos y más que nada para llevar un buen orden en sus reservaciones y asignaciones, ya que esto les ha traído problemas como empalmes de eventos en un mismo espacio, además que en la diversidad de eventos provoca que no exista un control real entre los eventos externos y los internos.

El Sistema está diseñado de tal manera que se puede instalaren un servidor local como en un servidor remoto por lo cual se utilizaron tecnologías conocidas como MySQL, PHP5, HTML, JavaScript y Servidor Apache. Pero antes de implementar estas herramientas habría que hacer un diseño tanto de los módulos como de la base de datos.

Se definieron los siguientes módulos: el Modulo de Control de acceso, el cual a partir de él se podrá acusar al sistema y se podrá administrar los roles de usuario, los cuales quedaron definidos en 4: Administrador, asistente, alumno y persona externa. Cada uno con diferentes funciones y algunos más complejas que otros, en si el administrador es donde más detalles se tuvieron que implementar pues en si es donde se lleva a cabo la mayor acción de operaciones posibles.

En la propuesta se planteó 2 diferentes módulos con respecto a la cuestión de los espacios físicos estos son en donde se administran y en donde se solicitan. Se puede ver de la siguiente manera en uno es donde el administrador gestiona los espacios físicos y también donde él puede validar una solicitud de espacio, sin embargo paralelamente está el modulo donde con anterioridad se realiza la solicitud y que es básicamente la única función de un usuario externo la de solicitar un espacio físico y revisar si fue aceptado o rechazado. Se hace hincapié en este módulo porque en la cuestión de verlo diseñarlo se vio que era más un bloque con dos sub-módulos.

De la misma manera en la parte de las encuestas hay dos módulos relacionados que es el de administrar las encuesta al cual tendrán acceso los usuarios administrador y asistente para editar crear, borrar y revisar resultados de las encuestas, mientras que en el otro será exclusivamente para que el usuario alumno pueda contestas la encuesta, siendo esta prácticamente la única función del alumno, aunque se esperaría que una versión posterior del sistema se puedan tener más funciones para este usuario.

Por otra parte el módulo de reportes, un módulo que interactúa con la información que los demás módulos van generando, ya que en este no solo se podrá generar información acerca de las encuestas o solicitudes sino también de la actividad de los usuarios, será operado funcionalmente por los usuarios administrado y asistente.

Una vez que su tuvo bien definido cada uno de los módulos se diseñó la base de datos, que a pesar de estar antes que el desarrollo de los módulos fue hasta el final donde quedo definida su última versión. Esta base de datos es muy importante que quedar bien definida ya que al ser en MySQL se

asegura que pudiera ser implementada en varios lenguajes esto por si en algún momento se migraba de PHP y HTML5 a distintos lenguajes por ejemplo java o totalmente en JavaScript ya que aunque cambiara toda interfaz la información de la base de datos y su diseño ayuda en demasía a que el proceso si así se diera fuera rápido en ese sentido.

Al final nuestra base de datos quedo implementado y constituida por 17 tablas que difieren de las 45 iniciales pero conforme se desarrolló el proyecto descubrimos que unas no eran necesarias ya sea por lógica o porque al final una operación programada la hacía innecesaria.

Teniendo definidos los módulos y sus funciones y teniendo ya la base de datos se desarrolló el armado del sistema.

Brevemente su función seria de la siguiente manera, lo primero es dentro de la base de datos dejar los datos del administrador ya que para la funcionalidad del sistema debe haber un administrador desde el principio esto se puede ver como cuando instalas un nuevo software y antes de verlo funcionar defines el usuario y password. Una vez que el Usuario administrador esta anexado el sistema funcionara así:

El usuario administrado se autenticara y podrá acceder a varios menús donde podrá administra a los usuarios, además en otro apartado podrá acceder a la administración de los espacios físicos, en otro más a la administración de las encuestas y por último la creación de los reportes.

En caso de que se requiera el acceso por parte de un usuario asistente no lo podrá hacer a menos que un usuario administrador lo registre, esta media fue implementada para evitar que un usuario con privilegios de edición fuera creado arbitrariamente.

Para los casos delos usuarios externos y alumno es más sencillo ellos pueden hacer sus registro al entrar a la aplicación para que luego realicen su autenticación, para cada uno les parecerá un menú diferente con las acciones más comunes que ellos realizaran, por ejemplo el alumno contestar una escueta, claro que para eso debe estar creada y activa pero además él debe contar con una clave de encuesta para poder acceder a ella esta medida evitara que se contesten de más una encuesta y esta clave será proporcionada por el profesor en la clase.

Por su parte el usuario externo podrá revisar los espacios físicos disponibles y con ello realizar una solicitud la cual posteriormente podrá verificar si está aprobada o rechazada, en caso de lo primero la función del sistema habrá terminado ya que el usuario tendrá asignado a una fecha y hora un espacio físico para su evento, en caso de estar rechazada el usuario podrá acceder a una apartado donde se le indica del por qué fue rechazada.

En un muy breve resumen se detalló este proyecto que al final hay una satisfacción porque se cumplieron los objetivos planteados y se pudo ver en las simulaciones que puede parecer un sistema sencillo pero con su grado de dificultad, también se está consiente que podría tener más potencial y más aplicaciones pero eso Seria cosa de otro proyecto.

OBJETIVOS

Objetivo General

Diseñar e implementar un Sistema de gestión que permita administrar el uso de los espacios físicos y monitorear la satisfacción de los estudiantes de la Universidad Obrera de México (UOM).

Objetivos Particulares

- Diseñar e implementar una Base de datos para almacenar la información de los módulos.
- Diseñar e implementar un módulo de control de acceso que realice el registro del usuario o en su caso valide sus datos.
- Diseñar e implementar un módulo que ayude a administrar la asignación de espacios físicos.
- Diseñar e implementar un módulo que permita realizar reservación de los espacios físicos.
- Diseñar e implementar un módulo para ayude a monitorear la satisfacción de los estudiantes con respecto a los cursos y talleres que se imparten en la UOM por medio de encuestas.
- Diseñar e implementar un módulo que permita realizar reportes de la gestión de espacios físicos y de las encuestas de satisfacción.

JUSTIFICACIÓN.

Actualmente la UOM no cuenta con los recursos necesarios para adquirir un software comercial que le permita gestionar los espacios y conocer la satisfacción de los usuarios. Una alternativa de solución es la realización de este proyecto de integración.

Dada la complejidad del proyecto se requiere de conocimientos relacionados con diseño y administración de bases de datos, programación de aplicaciones. Adicionalmente, la aplicación desarrollada optimizará un proceso administrativo.

Es importante resaltar que el proyecto no solo gestionará los espacios físicos de la UOM, para agilizar el trámite y optimizar los espacios disponibles en la UOM, si no también ofrecer una alternativa de evaluación de los servicios por parte del usuario final con el objetivo de identificar áreas de oportunidad que puedan mejorarse.

El Sistema generara beneficios como ahorro de tiempo, optimización de uso de espacios físicos, pero sobre todo un apoyo económico debido a que la UOM como muchas instituciones educativas consta de recursos limitados pero también tiene la necesidad de contar con un sistema de información que le ayude a agilizar y optimizar diferentes actividades administrativas, y que en caso de buscar en sistemas comerciales los precios rebasan por mucho su presupuesto.

MARCO TEÓRICO

ESPECIFICACIÓN TÉCNICA

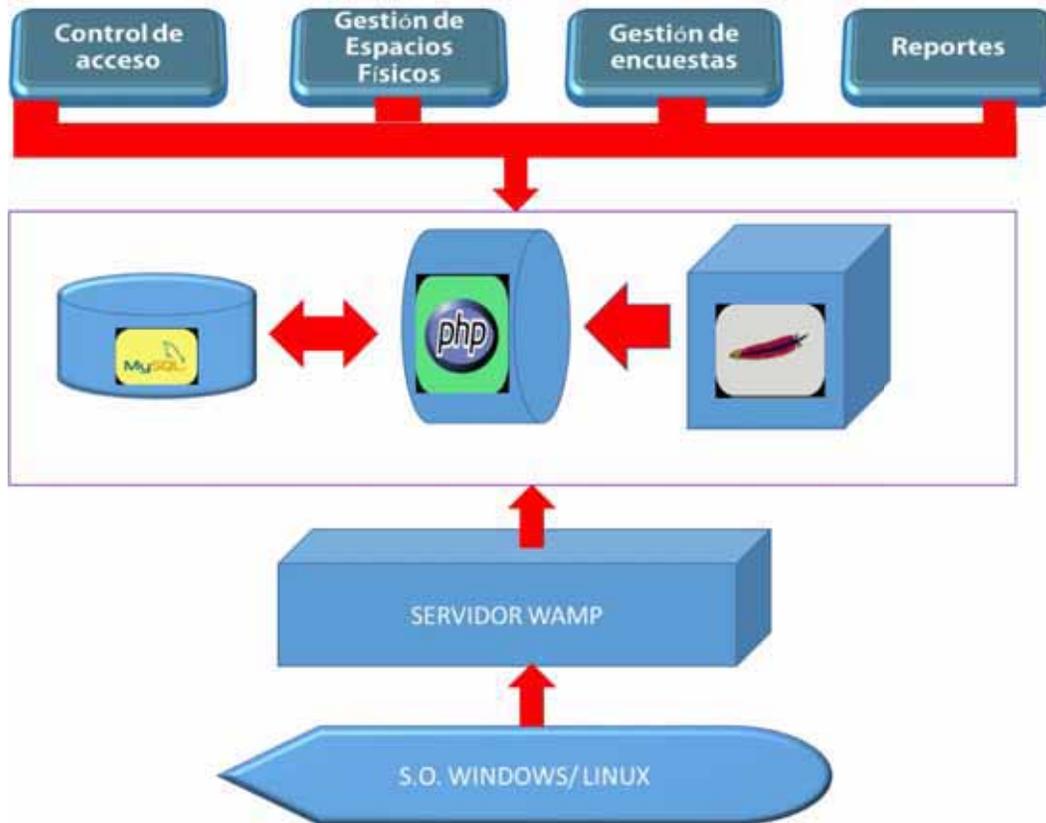


Ilustración 1. Arquitectura del Sistema

NUESTRAS HERRAMIENTAS

Se plantearon diferentes tecnologías a utilizar pero teniendo en cuenta que sería mucho mejor utilizar herramientas que fueran muy comunes y que en dado caso le pudieran dar continuidad al proyecto en un futuro, pero además se buscaron que fueran de licencia libre (open source).

A continuación se enlistara las herramientas que fueron ocupadas y además un poco de información de ellas para ir adentrándonos en el tema.

1. WAMP SERVER

WampServer es un entorno de desarrollo web para Windows en el cual se podrán crear aplicaciones web con Apache, PHP y base de datos en MySQL (*motor de base de datos*). Esta herramienta incluye además con un administrador de base de datos PHPMyAdmin con el cual podremos crear una nueva base de datos e ingresar la data de las tablas creadas en ella, realizar consultas y generar scripts SQL, como exportar e importar scripts de base de datos. WampServer ofrece a los desarrolladores herramientas

necesarias para realizar aplicaciones web de manera local, con un sistema operativo (Windows), un manejador de base de datos (MySQL), un software de programación script web PHP. WampServer se caracteriza por que puede ser usado de forma libre es decir no debemos de contar con alguna licencia el cual nos permita el uso de la misma, ya que pertenece a la corriente de "open source".¹

El elegir WAMP en lugar de otros entornos de desarrollo como "App Server" o "XAMP" no quiere decir que sean mejor o los otros peores solo fue que se tenía ya experiencia en su uso.

Dentro de WAMP server encontraremos ya incluidas las demás herramientas con las cuales se creó nuestro proyecto.

1.1 PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.²

Además de ser muy usado en varios entornos WEB se eligió utilizar PHP por su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales.

Su rápido acceso a la base de datos, su facilidad para crear sesiones de usuarios y su muy alto grado de información para su soporte, hace que utilizar PHP sea de las razones más importantes para elegir este lenguaje de programación.

La versión de PHP que se utilizó en este proyecto fue 5.5.12.

1.2 MYSQL

MySQL es un sistema de administración de bases de datos.

Una base de datos es una colección estructurada de datos. La información que puede almacenar una base de datos puede ser tan simple como la de una agenda, un contador, o un libro de visitas, o tan vasta como la de una tienda en línea, un sistema de noticias, un portal, o la información generada en una red corporativa. Para agregar, acceder, y procesar los datos almacenados en una base de datos, se necesita un sistema de administración de bases de datos, tal como MySQL.³

La versión que se utilizó de MySQL fue 5.6.17.

1.3 APACHE

Apache es un poderoso servidor web, cuyo nombre proviene de la frase inglesa "a patchy server" y es completamente libre, ya que es un software Open Source y con licencia GPL. Una de las ventajas

¹ Texto extraído de <https://codegeando.blogspot.mx/2013/03/php-wampserver-definicion-instalacion-y.html>

² <http://php.net/manual/es/intro-what-is.php>

³ <http://www.uruhost.com/x1sa10.php?c=4&i=15>

más grandes de Apache, es que es un servidor web multiplataforma, es decir, puede trabajar con diferentes sistemas operativos y mantener su excelente rendimiento.

Desde el año 1996, es el servidor web más popular del mundo, debido a su estabilidad y seguridad. Apache sigue siendo desarrollado por la comunidad de usuarios desarrolladores que trabaja bajo la tutela de Apache Software Foundation. ⁴

LA versión de apache instalad fue la 2.4.9.

Además estas 3 herramientas están incluidas en el paquete que te ofrece WAMP así que al instalarlo (apéndice A) habremos instalado al mismo tiempo PHP, MySQL y Apache.

1.4 MySQLi

¿Qué es la extensión MySQLi de PHP?

La extensión MySQLi, o como a veces se le conoce, la extensión de MySQL mejorada, se desarrolló para aprovechar las nuevas funcionalidades encontradas en los sistemas MySQL con versión 4.1.3 o posterior. La extensión MySQLi viene incluida en las versiones PHP 5 y posteriores. ⁵

La extensión MySQLi contiene numerosos beneficios, siendo estas las mejoras principales respecto a la extensión MySQL:

- a) Interfaz orientada a objetos
- b) Soporte para Declaraciones Preparadas
- c) Soporte para Múltiples Declaraciones
- d) Soporte para Transacciones
- e) Mejoradas las opciones de depuración
- f) Soporte para servidor empotrado

AL utilizar esta extensión también nos aseguramos estar al día ya que muchas funciones de PHP para conectarse a la base de datos han sido declaradas como obsoletas y sin soporte.

2. WORKBENCH

MySQL Workbench es una aplicación para el diseño y documentación de bases de datos (sucesora de la aplicación DBDesigner4) pensada para ser usada con el sistema de gestión de bases de datos MySQL. Existen dos versiones del producto, una es open source y la otra es una versión comercial. Evidentemente, la versión comercial proporciona algunas funcionalidades que pueden resultar de interés en algún ámbito, aunque la versión open source es más que suficiente para la realización de este proyecto.

Esta herramienta es sin duda un excelente apoyo de trabajo para realizar los diagramas además de hacer y probar scripts, a pesar de que todo lo que se hace en Workbench comúnmente los

⁴ <http://culturacion.com/que-es-apache/>

⁵ <http://php.net/manual/es/mysqli.overview.php>

desarrolladores lo hacen en PHPMyAdmin⁶ el entorno grafico de Workbench es de mucho más utilidad para ahorrar tiempo en el diseño de la base de datos y su administración para crear editar y eliminar tablas y campos.

3. JAVASCRIPT

JavaScript es un lenguaje de programación, al igual que PHP, si bien tiene diferencias importantes con éste. JavaScript se utiliza principalmente del lado del cliente (es decir, se ejecuta en nuestro ordenador, no en el servidor) permitiendo crear efectos atractivos y dinámicos en las páginas web. Los navegadores modernos interpretan el código JavaScript integrado en las páginas web.⁷

En un principio no se tenía contemplado utilizar JavaScript sin embargo las circunstancias llevaron a tener que integrarlo sobre todo porque es ideal para la cuestión dinámica de una aplicación WEB ya que a pesar de que en nuestro caso no es una página WEB existe muchas similitudes.

ESTILOS CSS

Los estilos se volvieron importantes y reiterando que un es una página web si es una aplicación web por lo cual al utilizar herramientas como PHP y HTML sin duda era casi instantáneo incluir los estilos CSS para darle una vista mucho más amigable a la interfaz.

SS es un lenguaje utilizado en la presentación de documentos HTML. Un documento HTML viene siendo coloquialmente “una página web”. Entonces podemos decir que el lenguaje CSS sirve para organizar la presentación y aspecto de una página web. Este lenguaje es principalmente utilizado por parte de los navegadores web de internet y por los programadores web informáticos para elegir multitud de opciones de presentación como colores, tipos y tamaños de letra, etc.

La filosofía de CSS se basa en intentar separar lo que es la estructura del documento HTML de su presentación. Por decirlo de alguna manera: la página web sería lo que hay debajo (el contenido) y CSS sería un cristal de color que hace que el contenido se vea de una forma u otra. Usando esta filosofía, resulta muy fácil cambiarle el aspecto a una página web: basta con cambiar “el cristal” que tiene delante. Piensa por ejemplo qué ocurre si tienes un libro de papel y lo miras a través de un cristal de color azul: que ves el libro azul. En cambio, si lo miras a través de un cristal amarillo, verás el libro amarillo. El libro (el contenido) es el mismo, pero lo puedes ver de distintas maneras.⁸

⁶ Paramas información de PHPMyAdmin se puede consultar la página: <https://www.inc.cl/blog/hosting/que-es-phpmyadmin>

⁷ <http://aprenderaprogramar.com/>

⁸ http://www.aprenderaprogramar.com/index.php?option=com_content&id=546:que-es-y-para-que-sirve-el-lenguaje-css-cascading-style-sheets-hojas-de-estilo&Itemid=163

DESARROLLO DEL PROYECTO

DISEÑO DE LOS MÓDULOS

Una vez definidas las herramientas de trabajo es momento de diseñar, en este caso de imaginar que es lo que va hacer nuestro sistema, así hay que plantearse el ¿Cómo?, ¿para qué? ¿De qué forma? Trabajar cada módulo.

En la imaginaria inicial siempre se tuvo la idea de una estructura inicial donde existieran 6 módulos sin embargo al final de decidió por solo considerar 4 módulos elementales en los que se basaría nuestro sistema y de ahí en adelante lo que daría pauta a esos 6 iniciales sería una cuestión de los submódulos.

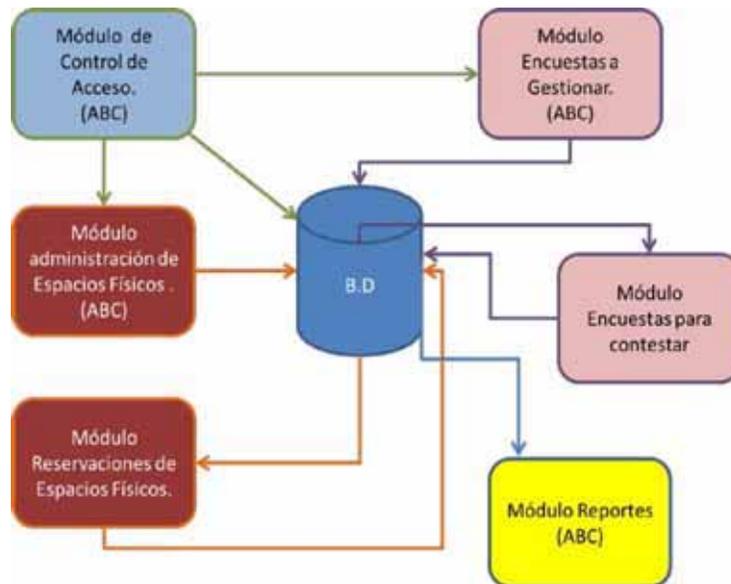


Ilustración 2. Módulos del Sistema en la propuesta.



Ilustración 3. Módulos del Sistema Actual.

CONTROL DE ACCESO

La funcionalidad del primer módulo considera el registro del usuario y la validación de sus datos, así como las funcionalidades ABC que se refiere a poder realizar Altas, Bajas y Cambios en la información de los usuarios.

Se definieron 4 Perfiles de Usuario:

Administrador: Este perfil de usuario tendrá todos los permisos, para manejar el sistema y administrarlo.

Asistente Administrativo: Este perfil de usuario tendrá acceso a toda la información pero tendrá restricciones para editar la información.

Alumnos: A este perfil de usuario solo se le conceden privilegios para consultar cierta información, dentro de sus permisos podrá realizar encuestas. Además tendrá que ser validado por el usuario Administrador.

Usuario Externo. Este perfil de usuario solo tendrá la opción de hacer una solicitud de un espacio físico.

El módulo control de acceso tendrá como función principal autenticar a los usuarios y así saber qué rol manejan y dirigirlos a su panel correspondiente, pero además antes de autenticar será necesario estar registrado en el sistema.

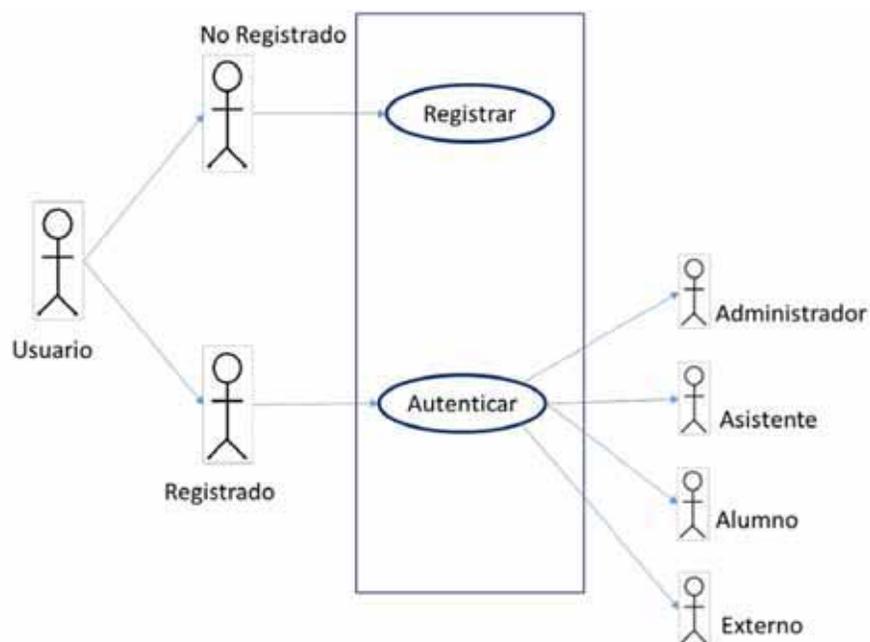


Ilustración 4. Caso de uso, Módulo de Control de Acceso.

Para poder asignarles un rol a los usuarios primero definiremos los componentes de un usuario pero para eso llamaremos a esto como cuando se usa modelado UML⁹ la entidad usuario.

Usuario prácticamente es nuestra entidad principal para la creación de este módulo, en esta entidad se almacenara la información general del usuario, como Nombre, Apellidos, Email, Teléfono, y se pensaría que en otra entidad se almacena la información como nombre de usuario y la contraseña sin embargo son datos que todos los usuarios tendrán así que serán atributos de la entidad. El rol del usuario o como lo hemos mencionado el tipo de usuario podría ser que otros sistemas se definiera con más atributos , me refiero a por ejemplo que el alumno cuente con datos muy específicos como matricula, carrera o campus, y en el caso de asistente y administrador sería algo como numero de trabajador, área y departamento; en el caso del usuario externo podría ser nacionalidad u ocupación sin embargo para este sistemas esos datos no serán necesarios ya que al analizar lo que se requiere de esa información no están necesaria para entender mejor esto explicaremos como se conforma el área que estaría ocupando el sistema.

El área de la UOM a la cual está dirigido este proyecto cuenta con un solo Coordinado que en nuestro sistema sería un administrador, en el área solo laboran 2 asistentes, uno por turno pero que este no es fijo de ahí el no considerar turno como parte de la información del asistente. Ahora los alumnos, por el momento no es necesario mucha información académica de ellos si no lo que sería necesario es que contesten la encuesta. EL usuario externo también no tiene mucho que ver su información personal y los detalles de las solicitudes que realice podrían estar incluido en la parte de los espacios físicos.

Así que después de entender mejor que la información básica del contacto es suficiente, aunque todavía estaría el detalle cómo definir como los diferenciaremos y la respuesta es haciendo una entidad que guarde los diferentes roles o tipo de usuario y tenga un identificador así construimos esta relación.



Ilustración 5. Diagrama de clases de Control de Acceso.

⁹ Lenguaje unificado de modelado.

ESPACIOS FÍSICOS

En este módulo habrá dos funciones fundamentales y que involucra a dos tipos de usuario (Administrador y usuario externo) exclusivamente en cuestión de cambio de información y uno más para solo consulta. Así que las funciones son las siguientes

Administrar el uso de los espacios físicos.

Aquí se administrará los espacios físicos se darán de alta, baja y podrán realizar cambios a la información que se almacenara en la base de datos para que se obtenga un catálogo de espacios disponibles y permita que el usuario externo realice la reservación y esperar que sea autorizada o denegada. Se administraran las reservaciones solicitadas, para aprobarlas o rechazarlas. Con esto se tendrá la función de evitar que se empalmen eventos tanto por día como por hora. Una vez autorizado el espacio físico el módulo tendrá una sección donde se almacenará información como la fecha, hora en que se ocupará, además del tipo de evento (externo o interno), por cuanto tiempo se ocupara el espacio y una breve descripción del evento que se realizara. En caso de ser rechazada el administrador podrá indicar el motivo por el cual fue rechazado y dejar una sugerencia para el usuario externo si así lo requiere solicitar otro espacio.

Reservación de espacios físicos.

En este módulo se tendrá el visible el catalogo actualizado en el módulo anterior y que podrá consultar el usuario externo (previa autenticación) para solicitar un espacio físico, después se le generara un folio con el cual posteriormente podrá consultar si se autorizó o no.

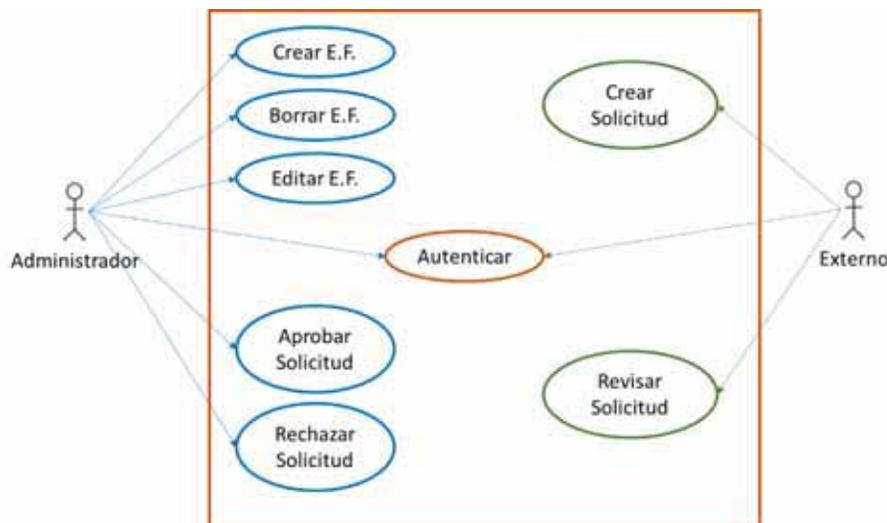


Ilustración 6. Caso de uso, Espacios Físicos.

Las principales entidades que en este módulo se identificaron fueron las siguientes: Espacio Físico donde la información de estos se almacenará para que la puedan consultar y su caso del administrador los administrará, la otra entidad es solicitudes la cual el usuario externo operará sin embargo el estatus será el que cambie el administrador.



Ilustración 7. Diagrama de clases, Espacios Físicos.

Se dejó claro que en este módulo la interacción de los usuarios administrador y externo es claro en que uno solicita y el otro valida esa solicitud, sin embargo también tenemos al usuario asistente que su función será mera consulta a la información almacenada, para verificar cuantas solicitudes en espera hay y cuantas ya fueron aprobadas o rechazadas, además también podrá consultar cuantos espacios físicos hay en el catálogo.

Así que en la ilustración 7 se anexa al usuario asistente, en la ilustración 7 se puede observar que las dos funciones parecerían exclusivas del asistente y no es que así sea pero tratamos de dar una noción del funcionamiento dentro de este módulo y cuando los demás usuarios realizan las diferentes funciones se sobreentiende que hace una consulta tanto al catálogo de espacios físicos como a las solicitudes, en el caso del usuario externo solo a las que el allá realizado.

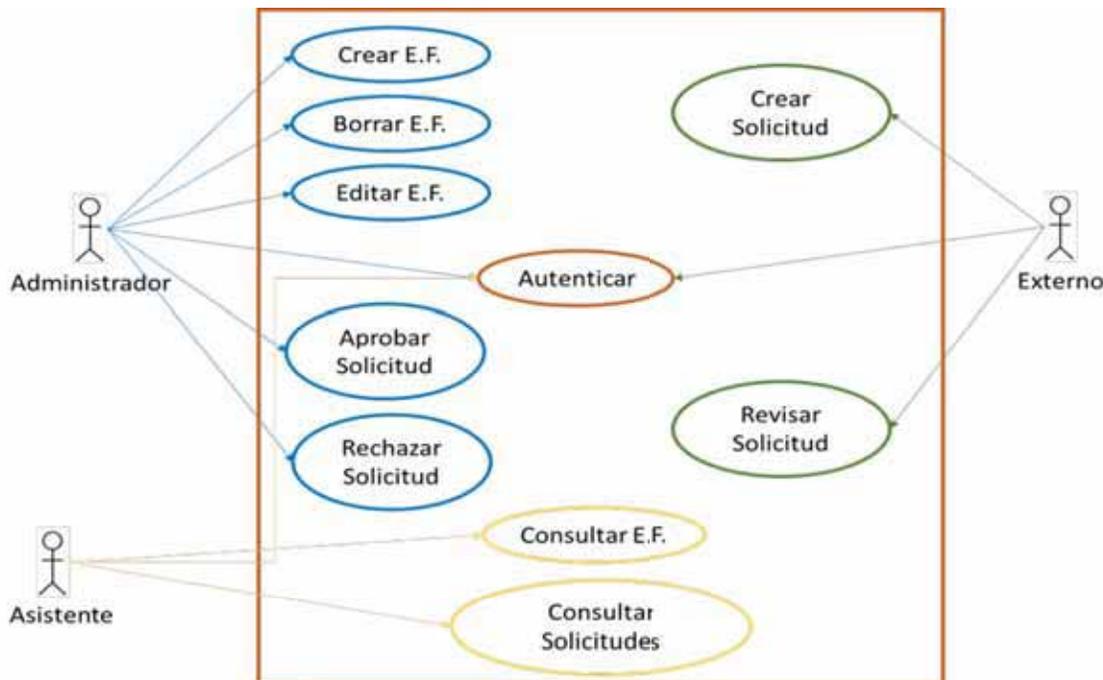


Ilustración 8. Caso de uso, Espacios Físicos incluyendo al Asistente.

ENCUESTAS

Este módulo es muy parecido al anterior debido a que será en su mayoría operado por los usuarios administrador y alumno pero la diferencia que aquí el usuario asistente sin duda tendrá un mayor grado de interacción ya que si podrá hacer acciones que hagan un cambio dentro del módulo y no solo consultar.

Las encuestas serán creadas tanto por el administrador como el asistente y también ambos podrán realizar la activación o desactivación de la encuesta así como su borrado y edición.

Los alumnos simplemente la contestaran sin llegar a ver los resultados.

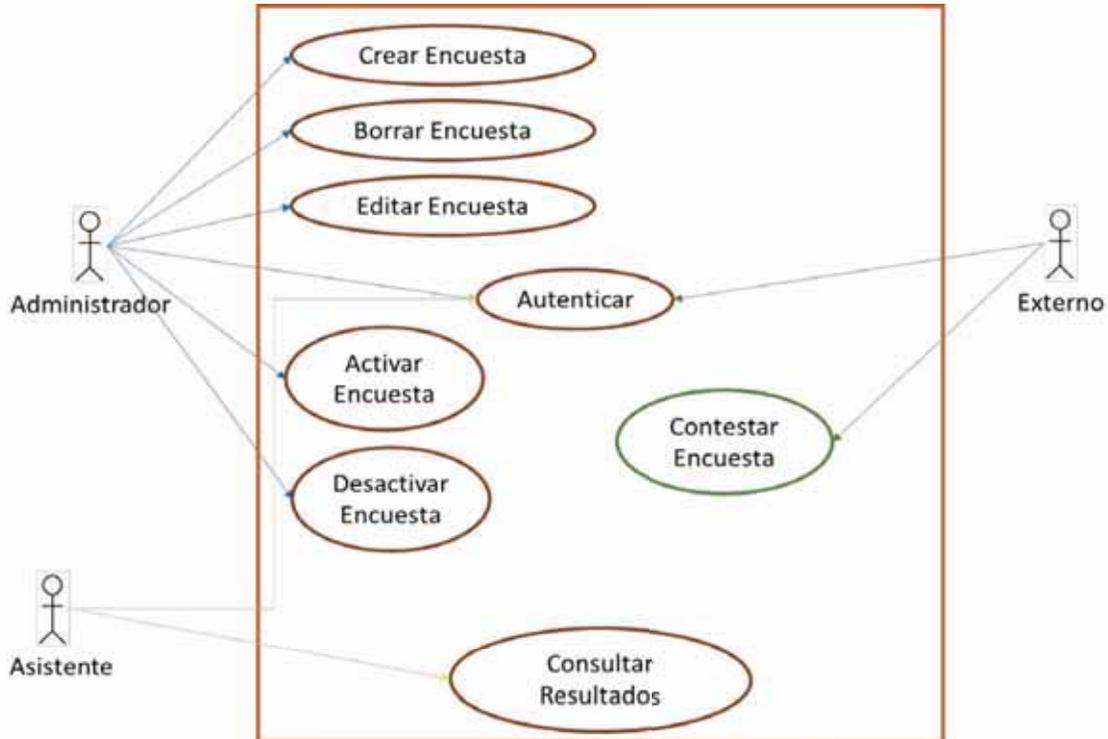


Ilustración 9.Caso de uso, Encuestas.

Este módulo tiene un funcionamiento de la siguiente manera, ya sea el administrador o el asistente crear a las encuestas, estas contendrán una clave que ayudara a que esa encuesta estará dirigida a un grupo, curso o taller en específico esta clave se le proporcionara a el profesor para que ellos se la otorguen a sus alumnos y así cuando ellos entren al sistema para contestar la encuesta primero ingresan la clave, para evitar que cualquiera conteste la encuesta y otra medida de seguridad será que se pueda activar o desactivar la encuesta así que esta podrá estar activa solo por un tiempo y después ya no la podrán contestar y así almacenar los resultados obtenidos.

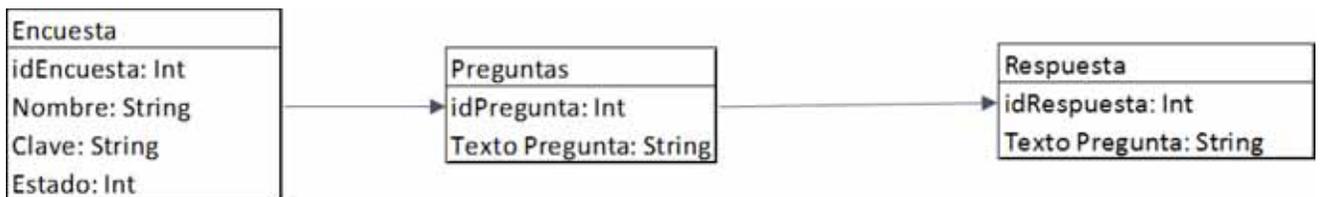


Ilustración 10.Diagrama de Clases, Encuestas.

REPORTES

Al final tendremos los reportes que será un módulo sencillo pero que trabajara con la información que se almacene de los demás módulos, pero si mencionamos esto será muy fácil pensar que entonces solo era una acción más a realizar sin embargo su estructura como nos da para un módulo completo, ya que en el tendremos varias funciones como el generar reportes de los usuarios, como la cantidad de ellos en cada tipo, y su acceso al sistema, además de ver los resultados de las encuestas y poderlos agrupar según las necesidades.

En este módulo solo podrán acceder los usuarios administrador y asistente.

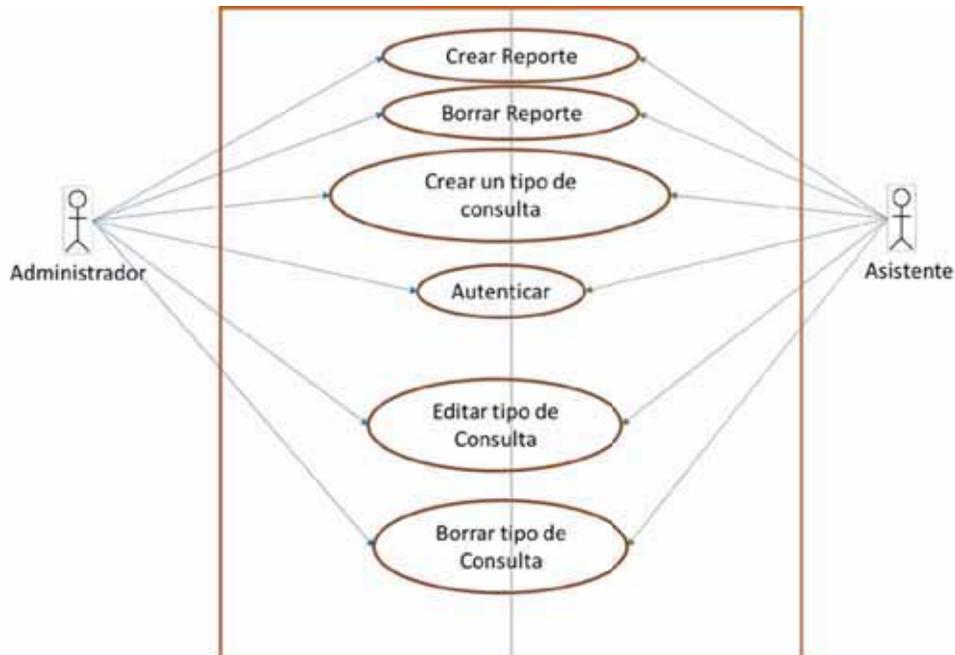


Ilustración 11. Casos de uso Reportes.

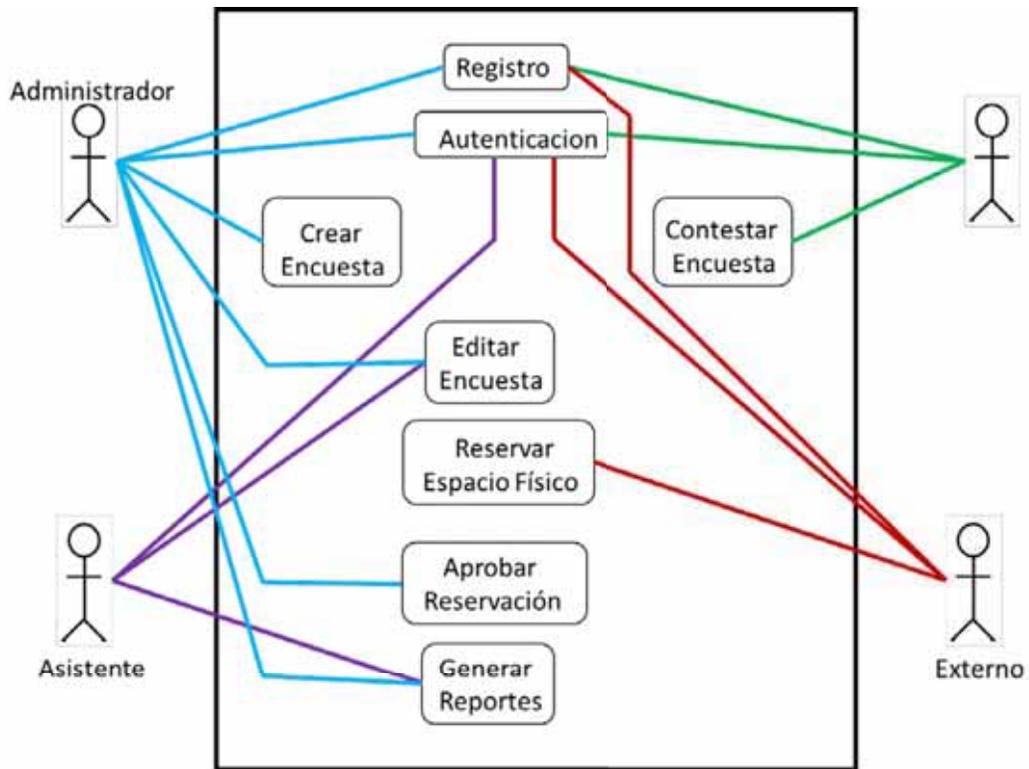


Ilustración 12. Caso de uso General del sistema.

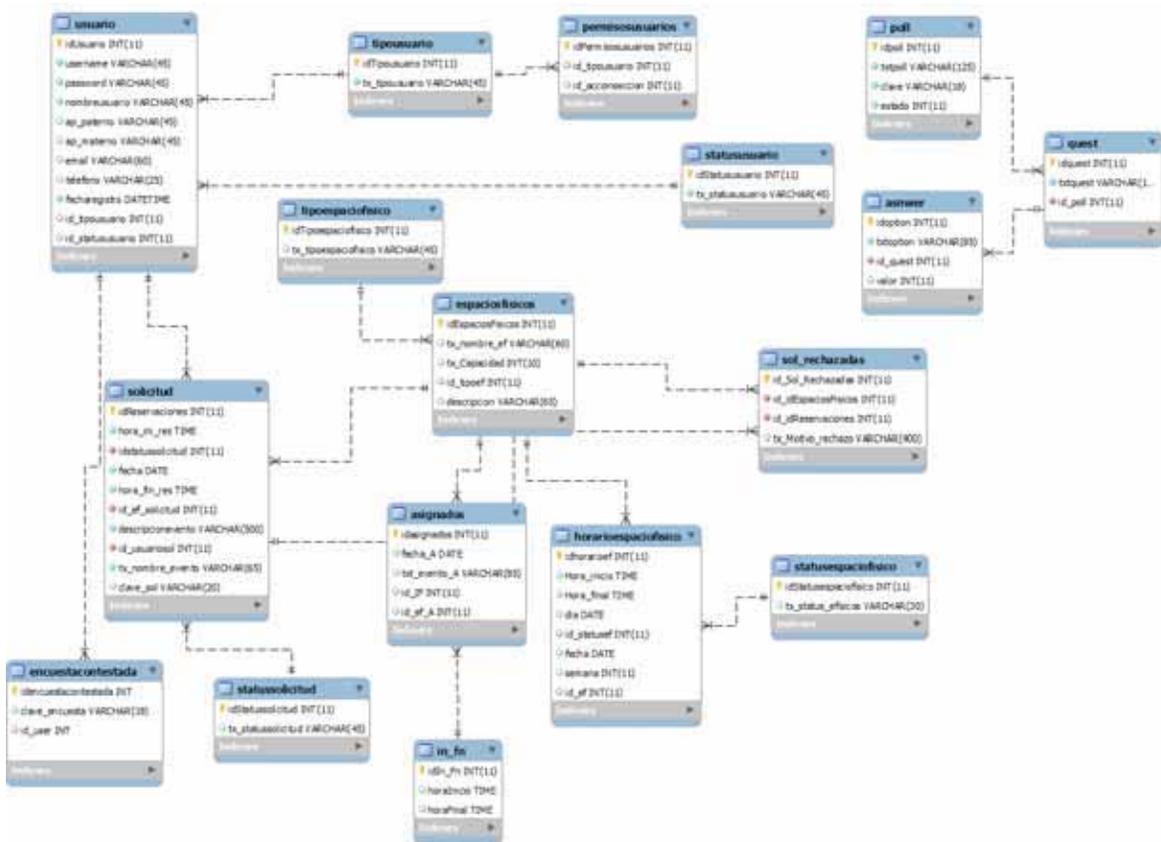


Ilustración 14. Diagrama entidad-relación Final.

CREANDO LA BASE DE DATOS

El código para la creación de la base de datos así como sus tablas se puede consultar en el [apéndice 2](#).

IMPLEMENTACIÓN DE LOS MÓDULOS

EL HTML Y NOTEPAD++

En el apartado de herramientas se mencionaron varias que se utilizaron en el desarrollo de este proyecto sin embargo también es buen mencionar a dos apoyos que sin duda fueron fundamentales uno es HTML que es el lenguaje que se emplea para el desarrollo de páginas de internet. Está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla, para muchas personas del medio de la computación la consideran obsoleta pero la versión 5 que fue la que utilizamos ofrece varias ventajas que haremos saber más adelante. Por su parte necesitamos un editor de texto y uno que nos diera muchas funcionalidades por el PHP, el HTML y el JavaScript por eso decidimos elegir Notepad++ por su versatilidad en varios lenguajes.

Decidimos colocar estos apoyos en esta parte por el hecho de que fue después de definir los módulos y crear la base de datos que nos dimos cuenta de la necesidad de dos herramientas como estas y por eso aquí es donde las incluimos.

ACCESO AL SERVIDOR LOCAL

Antes que nada es importante indicar todas las tareas importantes que se llevaron a cabo para crear este proyecto y este es importante.

Antes de empezar a tirar código como se dice en el ámbito de la programación, crearemos el lugar a donde por medio de un navegador en nuestro caso elegido es Google Chrome, podremos acceder a nuestros archivos y verlos gráficamente.

WAMP se instala en la raíz del computador así que a partir de ahí nos dirigimos a la carpeta con ese nombre (WAMP), ahí abrimos la carpeta www y dentro de ella creamos una carpeta que será la de nuestro proyecto, la nombramos en este caso PI-I16.



Ilustración 15. Dirección de alojamiento en el servidor.

Ahora cuando hagamos referencia a ella dentro del navegador y no poner toda la ruta como aparece en el navegador del explore haremos la siguiente sintaxis

127.0.0.1/PI-I16/

En nuestro navegador se verá algo así

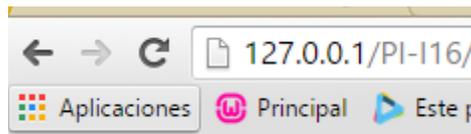


Ilustración 16. Dirección del servidor.

CREANDO EL MODULO DE CONTROL DE ACCESO

Lo elemental que tendrá que hacer nuestro modulo será crear la parte que valide el acceso de los usuarios teniendo en cuenta esta información. Vemos que solo consideramos 2 tablas pero vemos que hay una referencia como explicamos en la creación de la BD creamos una tabla del status del usuario para que se puedan bloquear usuarios si así se requiriera pero en este momento no lo consideraremos.

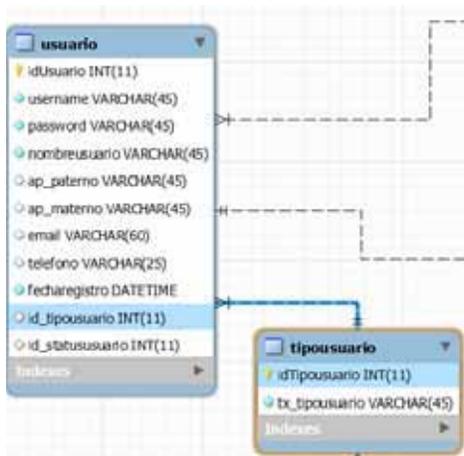


Ilustración 17. Diagrama entidad-relación para Control de acceso.

Antes continuar se hará la aclaración que no se incluirá código de programación pero si habrá una referencia en que parte de la carpeta del código se encuentra.

Primero tendremos nuestro index¹⁰ donde se encontrar la opción de ingresar o registrar y tendríamos algo así:



Ilustración 18. Ventana del panel inicial, index.php

Una vez que tenemos esto continuamos con el archivo de registrar

¹⁰ SE encontrara en la carpeta PI-116/index.php

SISTEMA DE GESTIÓN DE LA UNIVERSIDAD OBRERA DE MÉXICO

Ingrese los datos de autenticación

Nombre:

Apellido Paterno:

Apellido Materno:

Nombre de Usuario:

Contraseña:

Email:

Telefono:

Tipo de Usuario

Alumno Externo

Ilustración 19. Ventana de registro

Aquí analizaremos varios aspectos como el que no aparece idUsuario pero no aparece porque ese es un índice auto incrementable que MySQL cuando lo ocupemos lo incrementará conforme se llenen los registros. Pero los demás datos que nos planteamos claro que se encuentran pero con una nota importante que en el tipo de usuario no aparece administrador ni asistente y esos usuarios serán creados cuando el administrador esté autenticado para eso antes nosotros llenaremos sus datos y ya cuando esté en el sistema él podrá crear usuarios de cualquier tipo.

Ahora aquí lo que mencionamos, el por qué utilizar HTML 5 y es que nos da la oportunidad de que como esta que en los formularios ya se pueda establecer de qué tipo son los datos de entrada por ejemplo en email.

SISTEMA DE GESTIÓN DE LA UN

Ingrese los datos de autenticación

Nombre:

Apellido Paterno:

Apellido Materno:

Nombre de Usuario:

Contraseña:

Email: as

Telefono:

Tipo de Usuario

Alumno Externo

Incluye un signo "@" en la dirección de correo electrónico. La dirección "as" no incluye el signo "@".

Ilustración 20. Se muestra como se verifica que el apartado email tenga @.

Una vez que el registro sea exitoso se puede ir a la parte de registro.

Ahora la otra parte de este módulo es que nos dirija a lo deseado pero primero la autenticación.

Habrán datos por default ya pregrabados en el caso de usuario Admin nos enviara la administración, al Asis casi la misma pantalla pero con limitaciones y así sucesivamente a los otros dos tipos de usuario.

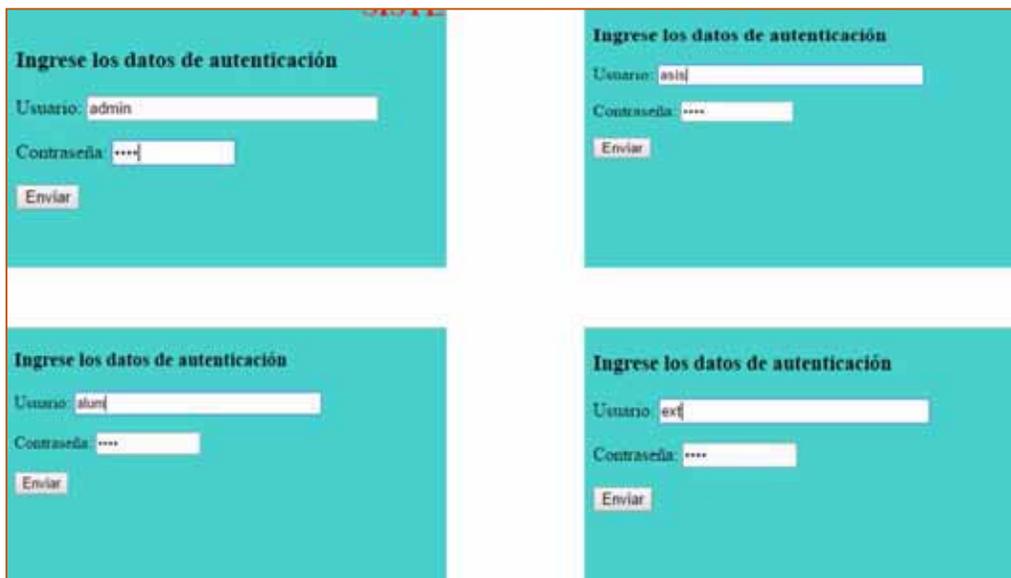


Ilustración 21. Pantallas de ingreso de cada uno de los tipos de usuarios.

Y la respuesta es la siguiente:

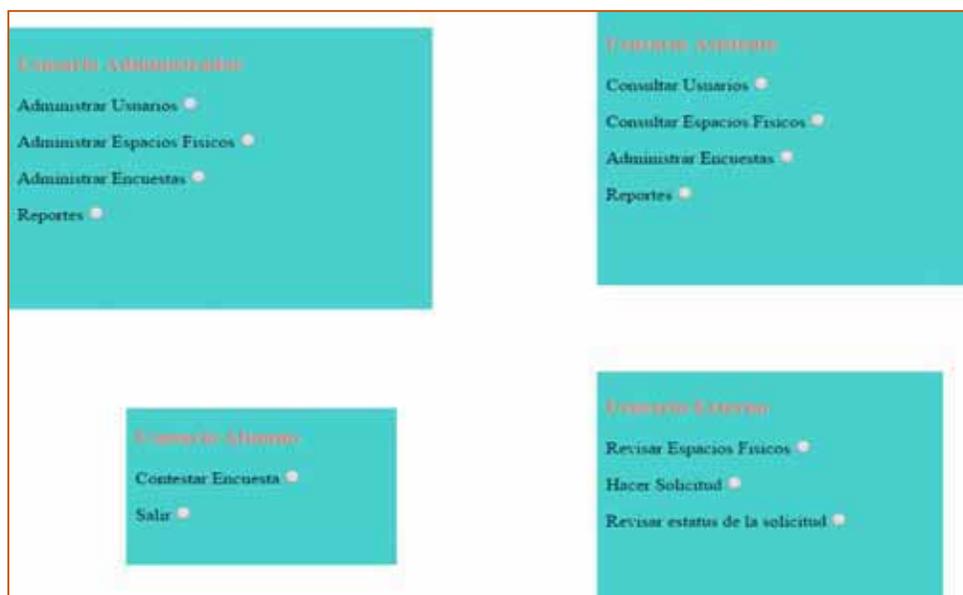


Ilustración 22. Los paneles que mostraría a ingresar cada uno de los tipos de usuario.

Hasta el momento ha sido básicamente HTML es hora de subir de nivel y meterle PHP pero al mismo tiempo MySQL.

Entonces tendremos que ir estructurando por así que se crearon dos carpetas en PI-I16 una se le nombro MySQL y la otra PHP.

En la carpeta MySQL se guardan nuestros query's¹¹ importantes y en la otra carpeta los script PHP P que necesitemos. También vamos a ir dándole un poco más de formato a nuestra página con respecto a la primera pero todavía sin entrar de lleno a los CSS pero aun así también creamos su carpeta de nombre CSS.

Bueno en la carpeta MySQL se creó el script para conexión de base de datos y desde aquí hay cambios importantes como y mostramos los dos métodos de conexión de base de datos una tradicional pero que ya es obsoleto y el nuevo que está basado en MySQLi. Aunque aún es muy utilizada la versión anterior al utilizarla nos arrojará un mensaje de warning indicando que ya está descontinuada esa función y recomendamos la nueva, es por eso que se eligió MySQLi que nos da mucha maniobra la trabajar con objetos.

Opción Tradicional	Opción MySQLi
<pre><code><?php // datos para la conexión a mysql define('DB_SERVER','localhost'); define('DB_NAME','exem'); define('DB_USER','root'); define('DB_PASS',''); \$connex = mysql_connect (DB_SERVER,DB_USER,DB_PASS); mysql_select_db(DB_NAME,\$connex);</code></pre>	<pre><code><?php function conectarse() { \$servidor="localhost"; \$usuario="root"; \$password=""; \$db="mydb"; \$conectar= new MySQLi(\$servidor,\$usuario, \$password, \$db) or die ("no se pudo conectar con el servidor BD"); return \$conectar; } \$connexion = conectarse(); ?></code></pre>

Ilustración 23. Comparación de los diferentes métodos para hacer la conexión a la base de datos.

Ahora enlistaremos todas las validaciones que haremos algunas desde el llenado del formulario y otras que haremos buscando en la BD.

- I. Que la contraseña sea alfanumérica con al menos una mayúscula
- II. Un email valido: de estructura mínima email@correo.com.
- III. Antes de enviar el formulario que ningún campo este vacío.
- IV. Verificar que el usuario no exista ya en la base de datos.

Algunas de las funciones HTML5 importantes por su ayuda fueron como la ya mencionada de correo y otras dos, la primera es pattern que nos permite el uso de expresiones regulares¹² y la segunda required que hace que un campo sea obligatoria y no se envíe vacío.

¹¹ (Query) Una petición para información específica tomada de una base de datos.

¹² Una expresión regular, a menudo llamada también regex, es una secuencia de caracteres que forma un patrón de búsqueda, principalmente utilizada para la búsqueda de patrones de cadenas de caracteres u operaciones de sustituciones.

Una vez que el usuario quedo se registrado correctamente se podrá autenticar y en esta parte también hay varias validaciones:

- I. Verificar que ningún campo este vacío.
- II. Verificar que los datos como nombre de usuario y contraseña coincidan.

Nuestro primer módulo estaría funcionando correctamente, aun se ve muy feo esto



Ilustración 24. Menú de administración

Así que desde de aquí vamos configurando nuestros estilos CSS y que servirá para que en los demás módulos se vean bien.

CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne.

```
h1 {color: red;}
```

h1 es el selector

{color: red;} es la declaración.

Creamos nuestra hoja de estilo que estará en la carpeta PI-116/css/estilos.css

Así se ve nuestra imagen ahora



Ilustración 25.Menu de Administración

Se ve mucho mejor pero aún le faltan varios estilos sin embargo esos fueron detallados más adelante mientras podemos ver que tenemos un menú de navegación tanto arriba como abajo esto aparecerá exclusivamente en los paneles de administrador y asíntete debido a que ellos realizan varias actividades y los otros dos tipos de usuario solo una o dos.

MODULO ESPACIOS FÍSICOS

Una vez que ya nuestro módulo de control de acceso funciona correctamente veremos cómo funcionara nuestro sistema de espacios físicos.

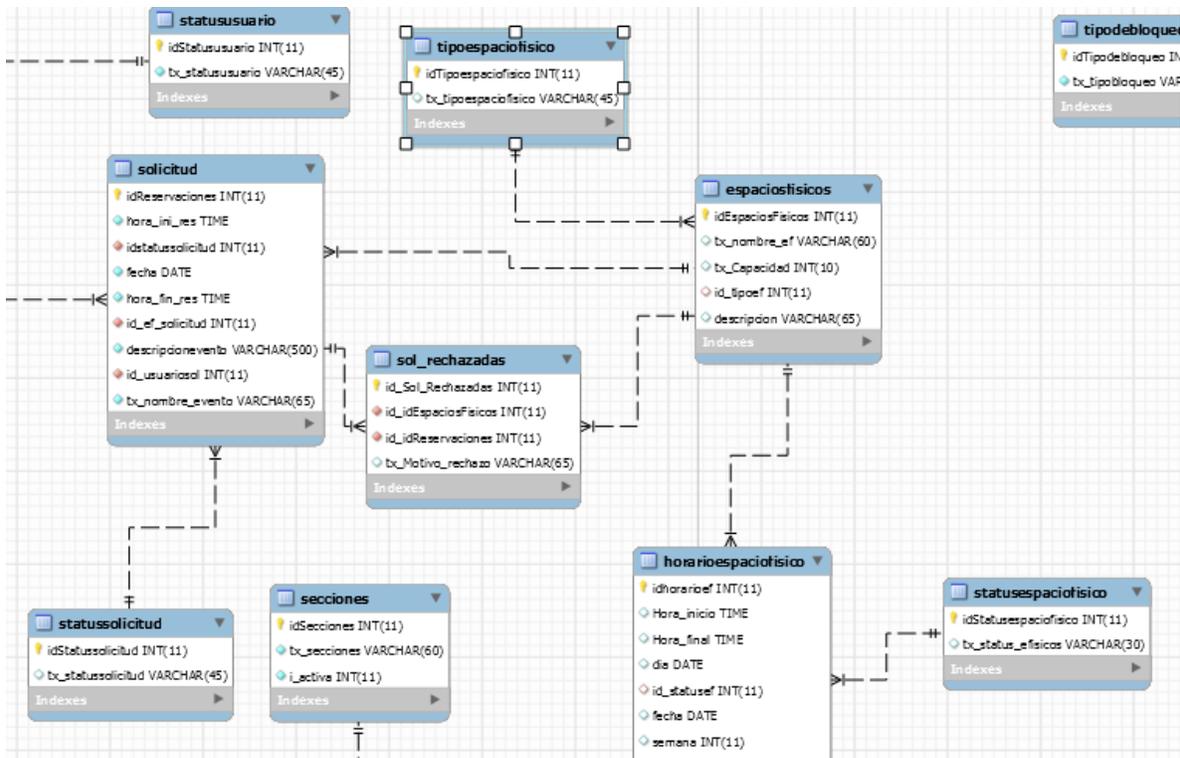


Ilustración 26. Diagrama Entidad-relación de las tablas a utilizar en el módulo de espacios físicos.

Si comparamos esta imagen con la primera donde había muchas tablas dedicadas para cómo realizar la reservación vemos que se redujo bastante y explicaremos su funcionamiento.



Ilustración 27. Menú donde se administran los espacios Físicos.

Como se ven la imagen el Administrador se encargara de dar de alta los espacios físicos que estarán disponibles la información como se ve en la imagen del modelo entidad –relación es el id_ef que se genera en automático con el ingreso exitoso de cada registro, el nombre del espacio físico capacidad, su tipo (que esta esta almacena en una tabla de tipo de espacios) donde es más fácil solo ligar con su id_tipoef. Este procedimiento lo ejemplificamos en la imagen siguiente.

idEspaciosFisicos	tx_nombre_ef	tx_Capacidad	id_tipoef	descripcion	idTipoespaciofisico	tx_tipoespaciofisico
6	SALON 1	30	1	aaaaa	2	AUDITORIO
7	SALON 4	20	1	xxxxx	3	BIBLIOTECA
8	AUDITORIO 1	150	2	bbbb	4	ESPLANDA
9	BIBLIOTECA	50	3	ccccc	1	SALON
10	ESPLANDA 3	140	4	ddddd		

Ilustración 28. Relación de tablas espacios físicos y tipo de espacios físicos.

Aunque aquí en el nombre de usuario parece que se repite como su el tipo de espacio es porque son datos de ejemplos sin embargo se espera que en lugar de llamarse auditorio1 debería llevar un nombre como Incalli Ixcahuicopa, Vicente Guerrero, Lombardo Toledano etc. Cualquier nombre que le ponen a los espacios.

Ya una vez que se tiene cargados los espacios físicos el usuario externo podrá verificar cuales existe para posteriormente hacer la solicitud.

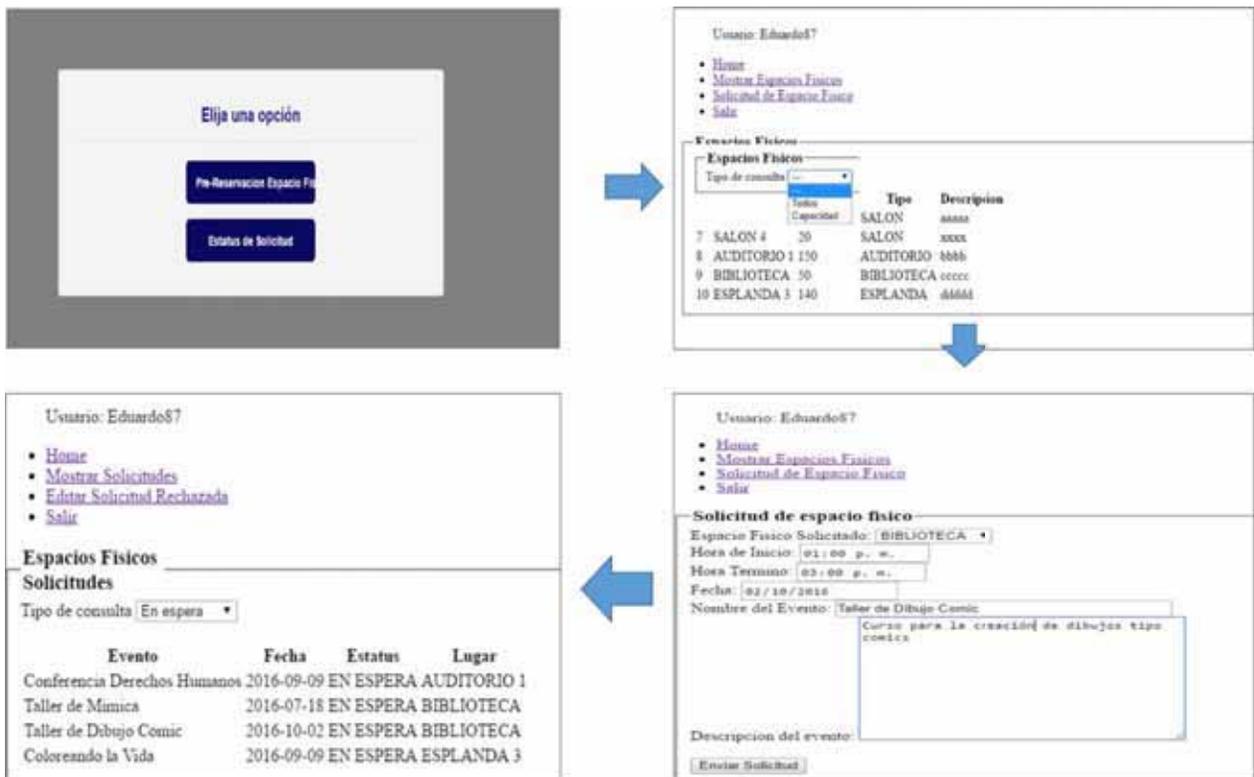


Ilustración 29. Proceso de solicitud de espacio físico.

En la figura podemos observar cómo va el proceso de la solicitud, se verifica los espacios disponibles y se pueden verificar por su capacidad, posteriormente se realiza una solicitud y aquí otro punto a HTML5 es que nos permite hacer en formularios una caja de tipo fecha y tipo hora facilitándonos la captura de los datos. También antes de enviar la solicitud habrá varias restricciones como:

- I. Evitar que algún campo vaya vacío.

- II. Cuando se envía los datos verificar que la fecha se superior a la actual.
- III. Que las horas estén en un rango de 8 a 19 horas (7:00pm) para el inicio del evento y de 9:00 a 20:00 (8:00pm) para el final.

Como se observa en el último recorrido del proceso de solicitud esta en mostrar las solicitudes que ha realizado el usuario externo en turno y vemos que puede hacer varias y cada una se pondrá en automático en espera.

El proceso continua con el usuario administrador

Mostrar Solicitudes

Tipo de consulta

 Todas
 Fecha

Usuario	Espacio Físico Solicitado	Fecha	Hora Inicio	Hora Final	Evento	Solicitud	Gestionar
Rex_98	SALON 4	2016-09-29	16:00:00	18:00:00	Exposición de Fotografías	EN ESPERA	Validar
Eduardo87	AUDITORIO 1	2016-09-09	10:00:00	12:00:00	Conferencia Derechos Humanos	EN ESPERA	Validar
Eduardo87	BIBLIOTECA	2016-07-18	15:30:00	18:10:00	Taller de Mímica	EN ESPERA	Validar
Gerardo_09	BIBLIOTECA	2017-02-02	14:00:00	16:06:00	Presentación de Libro	EN ESPERA	Validar
Eduardo87	BIBLIOTECA	2016-10-02	13:00:00	15:00:00	Taller de Dibujo Comic	EN ESPERA	Validar
Eduardo87	ESPLANDA 3	2016-09-09	09:00:00	12:00:00	Coloreando la Vida	EN ESPERA	Validar

Ilustración 30. Mostrar Solicitudes.

Como vimos en las imágenes anteriores se ve muy rustico el diseño de las tablas aquí ya detallamos mas eso para darle un estilo diferente. El administrador desde su panel podrá acceder al apartado de espacios físicos donde al elegir solicitudes vera todas solicitudes realizadas. Ahí él se encargara de validar y esta acción se refiere a aprobar o rechazar la solicitud eso podrá ser tomando en cuenta varios factores pero ya dependerá del usuario administrador y no tanto del sistema.



Ilustración 31. Página pide actualización para generar horario.

La imagen de la pantalla anterior es interesante ya que se elige validar una solicitud de la cual no se a generado un horario de eleccion, este horario de eleccion servira para verrificar que exista lugar disponible para la solicitud o no. La pagina nos solicita que recargemos y la magia se haraya que se generara un nuevo horario con los horarios disponibles entonces el usuario administrador podra asignar el horario solicitado y aprobar la solicitud, cabe señalar que los bloques de logistica estan marcados por horas en caso de que el solitante requiera horarios intermedios se aprobara ocn un horario mayor al soliitado o en otro caso se rechazaria la solitud y se indicaria en el motivo la cuestion de los horarios.

Administración de Espacios Físicos

Detalles de la solicitud

Nombre del Evento: Taller de Dibujo Comic
Descripción: Curso para la creación de dibujos tipo comics
Fecha: 2016-10-02
Hora de Inicio: 13:00:00 --- Hora de Terminó: 15:00:00
Espacio Solicitado: BIBLIOTECA

NOTA: Solo deje seleccionados los horarios a asignar

Horarios del "BIBLIOTECA" del día

2016-10-02			
Option	Hora Inicial	Hora Final	Evento
1	06:00:00	09:00:00	Libre
2	09:00:00	10:00:00	Libre
3	10:00:00	11:00:00	Libre
4	11:00:00	12:00:00	Libre
5	12:00:00	13:00:00	Libre
6	13:00:00	14:00:00	Libre
7	14:00:00	15:00:00	Libre
8	15:00:00	16:00:00	Libre
9	16:00:00	17:00:00	Libre
10	17:00:00	18:00:00	Libre
11	18:00:00	19:00:00	Libre

Aprobar Solicitud

Rechazar Solicitud

Ilustración 32. Tabla de horarios libres.

ENCUESTAS

El módulo de encuestas funciona aún más fácil que el de espacios físicos pero aun así su diseño de funciones constó más sobre todo para saber cómo almacenar la información y que se hiciera dinámicamente.

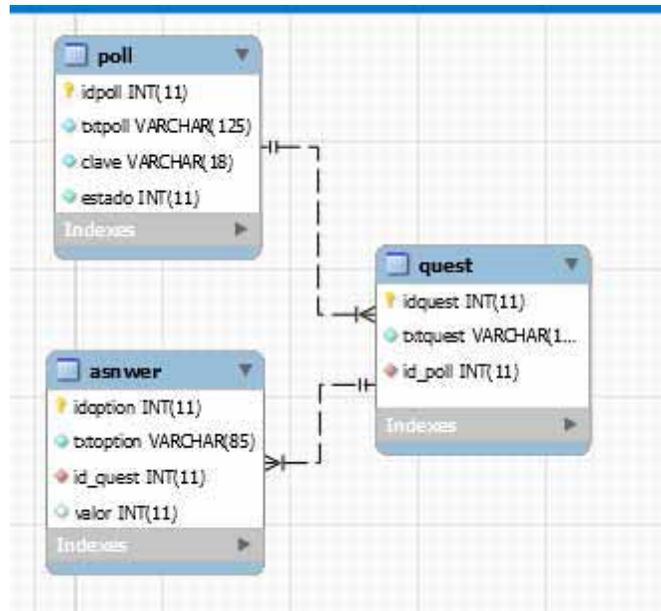


Ilustración 33. Diagrama entidad-relación para el módulo de encuestas.

Pero aun cuando parecía que era complicado decidimos trabajar muchas cosas con variables ***tipo _SESSION de PHP para guardar datos momentáneos que posteriormente no nos servirían y así evitar estar creando tablas momentánea o llenar tablas con datos que solo se usan una sola vez.

EL procedimiento es casi y muy similar al de espacio físico en el sentido que el administrador primero creara la encuesta y posteriormente el alumno la contestara. Hablando de la creación de las encuestas se recuerda que usuario asistente también podrá realizar esta acción.



Ilustración 34. Proceso de la creación de una nueva encuesta.

El alumno su única función en el sistema es el de contestar la encuesta para necesitar previamente que se otorga la clave una vez ingresado con una clave correcta el podrá continuar y al termina de contestar en el sistema se guardan sus datos y se relacionan con la encuesta realizada esto evita que conteste más de una vez la encuesta.

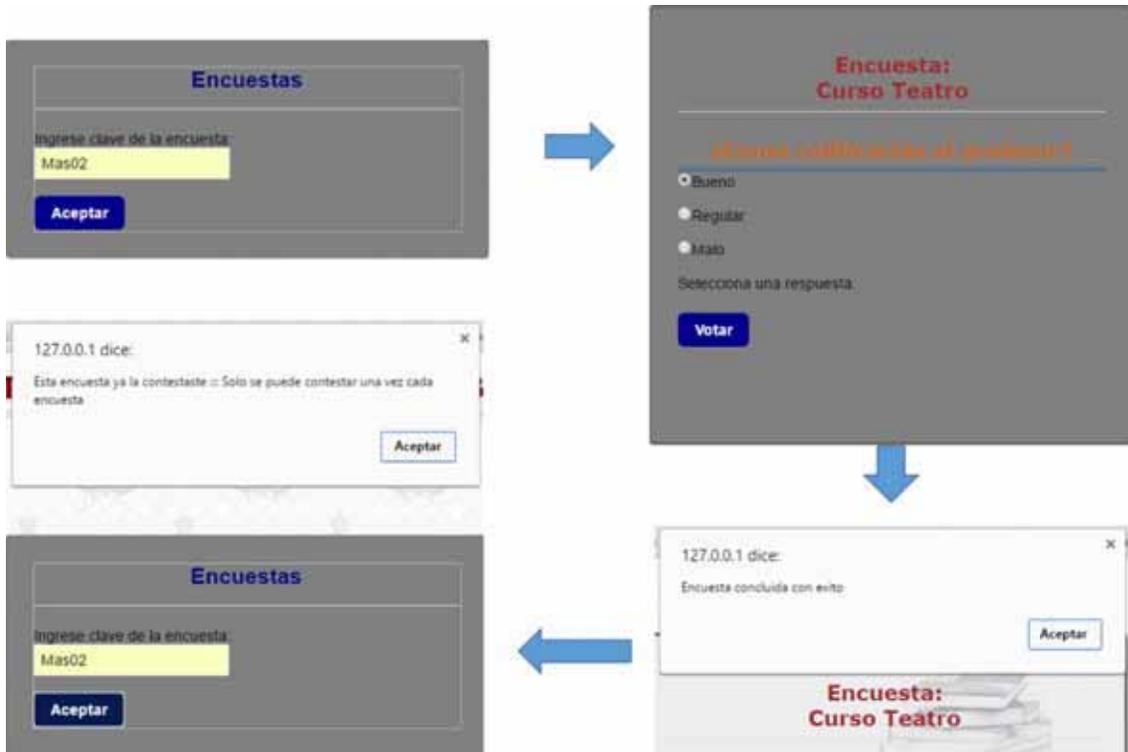


Ilustración 35. Proceso que sigue el alumno al contestar la encuesta.

REPORTES

En la parte de reportes podremos generar reportes, aquí se pueden hacer varios tipos de consultas para después exportar a un archivo PDF.



Ilustración 36. Panel de administración de reportes.

DETALLANDO LA INTERFAZ

ESTILO A EL SISTEMA

Después de ver imágenes muy simples y sin mucho diseño estético decidimos darle un toque mucho más amigable a la interfaz que vamos a utilizar ahí investigamos mucho de CSS viendo que se personalizan todo desde la forma, hasta los párrafos y por supuesto los fondos, pero en donde nos ayudó más los estilos fueron en las tablas, ya que por variante tamaño en varios se tuvo que limitar para que el contenido no se viera muy desproporcionado.

Mostramos algunas imágenes donde se puede notar este cambio:

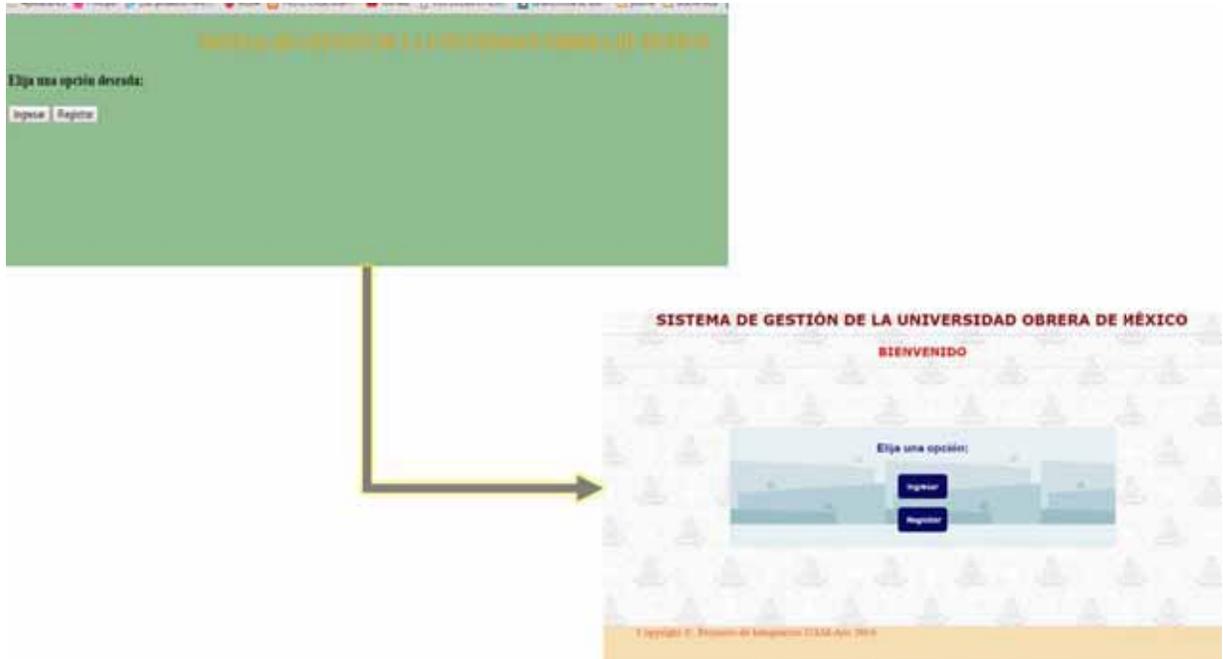


Ilustración 37. Comparación de antes y después de los estilos CSS.

APOYO DE JAVASCRIPT

Sin duda queremos hacer mención a el uso de JavaScript porque muchas veces hay esa decisión de que o se usa PHP o JavaScript pero sin embargo este proyecto sirvió para darnos cuenta que ambos pueden funcionar al mismo tiempo de una manera muy buena.

No fueron muchos los scripts que se utilizaron pero si muy útiles aquí los mostramos:

Primero las útiles alertas que nos ayudan para indicarle al usuario si algo se realizó correctamente o incorrectamente y así él sepa hacer su siguiente acción.

```
<script lenguaje='javascript'>alert('Usuario o Contrase\u00F1a Incorrecta ::: Verifique sus Datos '); location.href = 'auteticacion.php';</script>
```

Ilustración 38. Script de las alertas.

Lo interesante que es un script muy sencillo y ya con un poco más de conocimientos hasta se puede cambiar el diseño por cuestiones de tiempo se decidió dejar el diseño por default.

Otros fueron los “Window.location” que sin duda fueron útiles a la ahora de ahorrar más y más saltos de páginas y con ellos pudimos mejorar la navegación.

```
<script>
window.onload = function()
{
  var lista = document.getElementById("consulta-lista-sol");
  lista.onchange = function()
  {
    window.location = "?op=Solicitud&consulta-sol="+lista.value;
  };
}
</script>
```

Ilustración 39. Script que nos permite navegar en la misma ventana.

LAS CLASES CSS

Una clase es una definición de un estilo que en principio no está asociado a alguna etiqueta HTML, pero que podemos asociar a etiquetas concretas.

Para ello, en primer lugar definimos la clase:

```
.nombre_de_nuestra_clase {valor1: valor; valor2:valor, etc..}
```

Es decir, escribiendo un punto seguido del nombre que le queramos dar a la clase, y definiendo el estilo como lo definimos para cualquier otra etiqueta: para propiedad: valor separados por punto y coma y encerrados entre llaves. Además, podremos definir cuantas clases necesitemos.¹³

Incluimos el texto anterior porque fue de gran ayuda ya que esta parte ayuda a darle forma a nuestro sistema para elementos similares detallarlos de manera diferente sin estar creando mas hojas de estilo si no simplemente creado las clases correspondiente para cada forma utilizada.

¹³ http://www.usabilidad.tv/hojas_de_estilos_css/clases.asp

```

.wrap1{
  padding: 35px;
  margin: 0 auto;
  width: 555px;
  background: #F3F3F3;
  border-radius: 4px;
  margin-top: 110px;
  border: 1px solid #E0E0E0;
}
.wrap2{
  padding: 25px;
  margin: 0 auto;
  width: 655px;
  background-image: url(../img/fonbasic.jpg);
  border-radius: 5px;
  margin-top: 110px;
  border: 1px solid #E0E0E0;
}
.wrap3{
  padding: 25px;
  margin: 0 auto;
  width: 405px;
}

```

Ilustración 40. Clase creada wrap y sus variantes.

VISUALIZACIÓN DE LOS RESULTADOS DE LAS ENCUESTAS

Una parte en la que se trabajó mucho porque queríamos que quedara lo mejor posible es la visualización de los resultados de las encuestas para que no fueran una información básica de texto o en una tabla sino que también mostraran algo de estadística desde su visualización para que así en verdad puedan útiles a la hora de revisar estos resultados.

Por eso se atendió con cuidado esta parte y en la imagen siguiente se puede observar como quedo la visualización de los resultados donde no solo muestra el número de respuestas elegidas de una determinada opción si no que su porcentaje.



Ilustración 41. Así se mostraran los resultados de las encuestas.

SIMULACIONES

REALIZANDO PRIMERAS PRUEBAS

Que paso con las primeras pruebas con el sistema completo, en general función bastante bien pero tuvo su detalle s más en cuestión grafica de que las tablas no quedaban justas, algunas impresione de “echo¹⁴”.

REPORTE DE ERRORES

Varios errores se presentaron durante las pruebas, entre los más comunes eran errar al hacer una llamado a la BD y no elegir bien la variable. Otros errores s eran que se vieron eran los originados por la utilización de un mal Query lo que al hacer la consulta te mostraba resultados.

¹⁴ Con este comando se puede imprimir texto en PHP

RESULTADOS

Se hicieron varias pruebas y se obtuvieron varios resultados como los siguientes:

El sistema se conectó en red para que varios alumnos pudieran contestar la encuesta, en total fueron 23 alumnos conectados simultáneamente y no hubo ningún inconveniente.

En todo momento en el que los alumnos estuvieron conectados se encimo información o se mostró información de ellos en otro ordenador que pudiera ser vulnerabilidad de seguridad.

El sistema al conectarse se probó en máquinas que tienen Ubuntu, Windows 7 y Windows 10. La conexión se manejó así, 4 laptops quedaron conectadas por medio de WIFI y las 4 contaban con Windows 7, otras 3 mini laptop se conectaron por medio de Ethernet y contaban con Windows 10, 13 computadoras tipo PC 5 con Windows 7 y 8 con Windows 10 fueron las que estuvieron conectadas y 3 computadoras también tipo pc pero sistema Ubuntu 14 se conectaron y pudieron acceder sin mayor problemas, ya que la conexión la hacían a través del navegador.

En cuestión de la reservación de los espacios físicos también se obtuvieron buenos resultados se conectó el sistema al servidor local y se abrió para que remotamente se hiciera una reservación de espacio físico y posteriormente se notificara que había sido aprobada...

Varias cuestiones graficas solo pudieron verse en todo su esplendor en navegadores más utilizados como Chrome, Mozilla, Opera, Safari y Edre pero no en otros como internet explorer 11 y Maxthon.

Otra cuestión de seguridad y gracias a la SESSION de PHP se logró que a 5 min sin absolutamente nada de actividad en el sistema este te solicita nuevamente tu autenticación, en este punto lo que si vemos como una mejora a futuro es que si estabas en un procedimiento como la reservación de un espacio físico el sistema borrar la información que tenías hasta antes del bloque por inactividad lo que origina que vuelva a ingresar los datos.

Sin duda los resultados fueron satisfactorios y se pudo cumplir la meta planteada en la propuesta.

CONCLUSIONES

Fue un trabajo muy interesante, porque desde el principio tuvimos cambios en cuestión de las herramientas y sobre la marcha incluso descubrimos y aprendimos el funcionamiento de algunas que ayudaron en el proyecto.

Se concluyen varios aspectos como el que se lograron varias metas que se pusieron en la propuestas, una de ellas los objetivos de crear el sistema con sus dos funcionalidades principales. Entre los objetivos particulares se detallaba cada uno de los módulos los cuales quedaron desarrollados completamente como se planteó. Las especificaciones funcionales tal vez no quedaron resultas al 100% y es que no se pudo probar el sistema con 200 usuarios al mismo tiempo, pero con las pruebas realizadas podemos concluir que el sistema los soportara sin problemas.

Se puede concluir también que muchas veces se puede menospreciar las herramientas comunes sin embargo crear aplicaciones con ellas tiene su grado de dificultad y que sean muy utilizadas no tiene para nada que ver con que sean fáciles. Y aunque algunas de ellas tiene muchos beneficios que nos ayudan a ahorrar tiempo no quiere decir que el trabajo se reduce si no simplemente los esfuerzo se enfocan a otra parte del proyecto.

El sistema no es perfecto y no está cerca en serlo pero es funcional sin embargo talvez pueda tener un punto flaco y que no depende mucho de él y es que será mucho más rápido dependiendo el hardware que se utiliza para hacerlo funcionar, pero además también tiene limitaciones en software y es que en navegadores como internet explored no explota todas sus características al igual que versiones antiguas de los navegadores más utilizados.

Como conclusión final se puede decir que fue un proyecto que hizo sacar los conocimientos más empolvados adquiridos en la carrera desde las matemáticas, hasta las materias de computación y algoritmos dejándome en lo particular muy satisfecho por mi trabajo y el apoyo de mis asesores.

BIBLIOGRAFÍA

Libros

MERCER, Dave W.; **KENT**, Allan; **NOWICKI**, Steve D.; **MERCER**, David; **SQUIER**, Dan y **CHOI Wankyu**. Fundamentos PHP 5 (1ª Edición), 2005; Madrid España; Ed. Anaya Multimedia.

LOPEZ QUIJANO, Jose; Domine PHP y MYSQL (2da. Edición), 2009; México, D.F.; Editorial Alfaomega RA-Ma.

LUJÁN Mora, Sergio. Programación en Internet: Clientes Web (1ª edición), 2001; Editorial Club Universitario.

TORRES Remon, Manuel Ángel; Desarrollo de Aplicaciones Web con PHP Y MYSQL (1ª Edición). 2016; Castilla España; Ed. S.A. MARCOMBO.

FLANAGAN, David; JavaScript: The Definitive Guide (6.ª edición), 2011; United States of America; Ed. O'Reilly.

LAURA, Lemay; Aprendiendo HTML para WEB en una semana (3ra. Edición), 1995; Sams/PHI.

BEATI, Hernán HTML5 y CSS3 Para diseñadores (1ª Edición), 2015; Editorial Alfaomega.

Páginas Web.

<https://www.um.es/atica/documentos/html.pdf>

<https://secure.php.net/manual/es/index.php>; 2001-2016 The PHP Group

http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=75&Itemid=203; 2006-2016 aprenderaprogramar.com3

APÉNDICES

APÉNDICE 1. Instalación de WAMP SERVER

Para empezar debemos dar doble clic sobre el instalador Aparece la ventana y se selecciona Next, que significa siguiente...



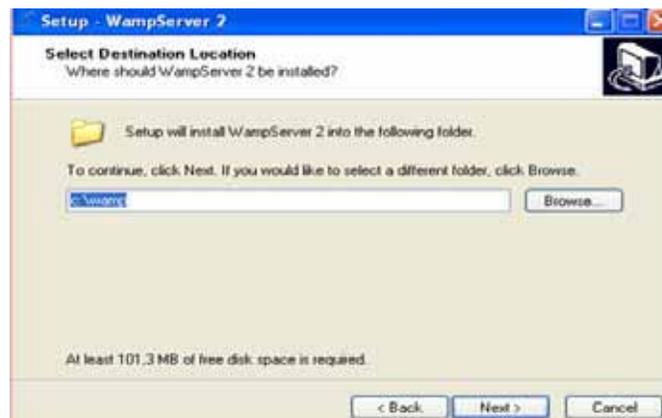
Debemos aceptar los términos de uso.



Cuando aceptamos los términos se activa la ventana Next, (siguiente).

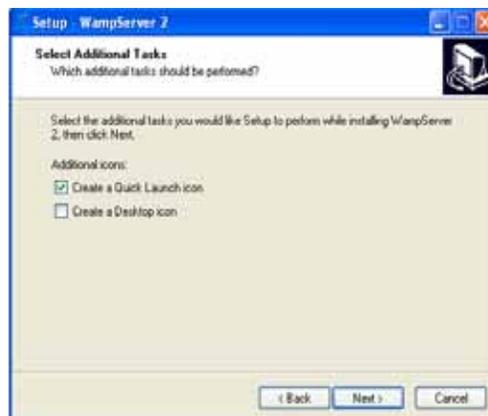


Clic en Next y seguimos con el proceso de instalación, nos aparece la siguiente ventana.



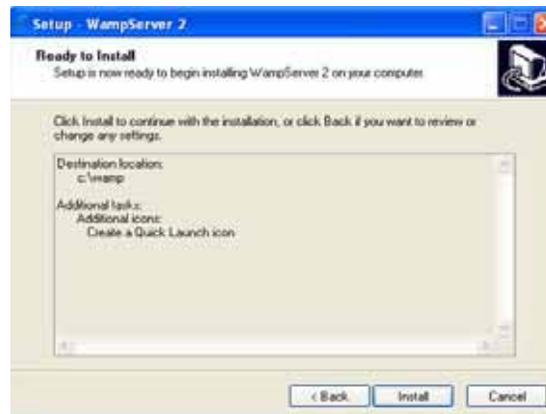
Esta ventana nos indica la carpeta en donde se va instalar el programa

Le damos next y aparece la siguiente ventana.

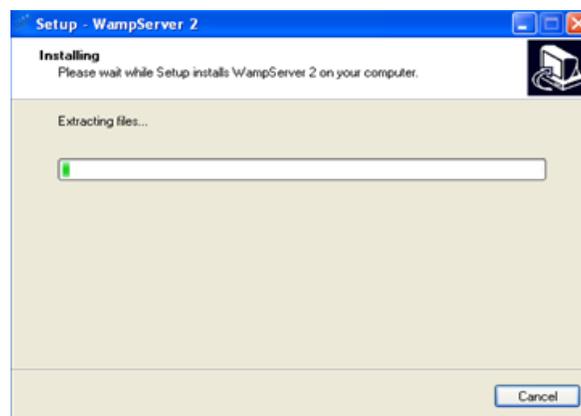


En esta nos dice que debemos seleccionar un botón stop para que lo arranquemos Cuantas veces queramos; Y luego clic en siguiente.

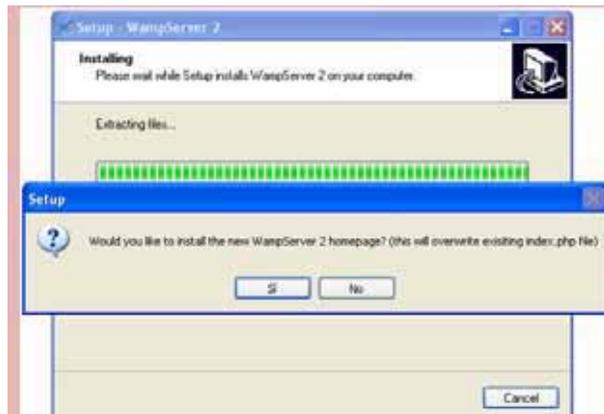
Aparece esta ventana en la que nos muestra el proceso de instalación damos clic en install.



Se está realizando ya el proceso de instalación del programa.



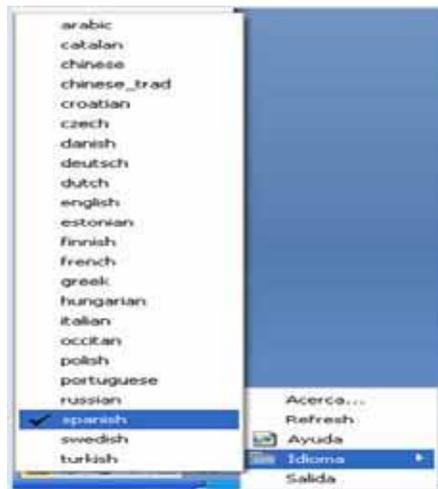
En esta ventana nos pregunta que si queremos instalar el xampp de wampserver Oprimimos si y pasa a la siguiente ventana.



Y para finalizar nos muestra la venta donde mantenemos tildada la opción Launch WampServer, y le damos finish y comienza a iniciarse el servidor.



Se tiene que seleccionar el idioma



Luego se busca el programa que se llama wampserver y se ejecuta la opción LOCALHOST



Y el Primer pantallazo que nos muestra el programa para empezar a ejecutarlo



APÉNDICE 2. Tablas de creación de MySQL

```
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
```

```
USE `mydb` ;
```

```
-----  
-- Table `mydb`.`acciones`  
-----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`acciones` (  
`idAcciones` INT(11) NOT NULL AUTO INCREMENT,  
`tx_accionseccion` VARCHAR(200) NULL DEFAULT NULL, PRIMARY KEY (`idAcciones`),  
UNIQUE INDEX `idAccionesSecciones` UNIQUE (`idAcciones` ASC)) ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

```
-----  
-- Table `mydb`.`secciones`  
-----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`secciones` (  
`idSecciones` INT(11) NOT NULL AUTO INCREMENT,  
`tx_secciones` VARCHAR(60) NOT NULL,  
`i_activa` INT(11) NOT NULL, PRIMARY KEY (`idSecciones`),  
UNIQUE INDEX `tx_secciones` UNIQUE (`tx_secciones` ASC), UNIQUE INDEX  
`idSecciones` UNIQUE (`idSecciones` ASC))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

-- Table `mydb`.`accionseccion`

```
CREATE TABLE IF NOT EXISTS `mydb`.`accionseccion` (  
  `idaccionseccion` INT(11) NOT NULL AUTO INCREMENT,  
  `as_seccion` INT(11) NULL DEFAULT NULL,  
  `as_accion` INT(11) NULL DEFAULT NULL,  
  `i_activa` INT(11) NOT NULL, PRIMARY KEY (`idaccionseccion`),  
  UNIQUE INDEX `idaccionseccion_UNIQUE` (`idaccionseccion` ASC), INDEX `asseccion_idx`  
  (`as_seccion` ASC),  
  INDEX `asaccion_idx` (`as_accion` ASC), CONSTRAINT `asaccion`  
  FOREIGN KEY (`as_accion`)  
  REFERENCES `mydb`.`acciones` (`idAcciones`)  
  ON DELETE NO ACTION ON UPDATE NO ACTION,  
  CONSTRAINT `asseccion` FOREIGN KEY (`as_seccion`)  
  REFERENCES `mydb`.`secciones` (`idSecciones`)  
  ON DELETE NO ACTION ON UPDATE NO ACTION)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

-- Table `mydb`.`in_fn`

```
CREATE TABLE IF NOT EXISTS `mydb`.`in_fn` (  
  `idIn_Fn` INT(11) NOT NULL AUTO INCREMENT,  
  `horaInicio` TIME NULL DEFAULT NULL,  
  `horaFinal` TIME NULL DEFAULT NULL, PRIMARY KEY (`idIn_Fn`))  
ENGINE = InnoDB AUTO INCREMENT = 12
```

DEFAULT CHARACTER SET = utf8;

-- Table `mydb`.`tipoespaciofisisco`

CREATE TABLE IF NOT EXISTS `mydb`.`tipoespaciofisisco` (
`idTipoespaciofisisco` INT(11) NOT NULL AUTO INCREMENT,
`tx_tipoespaciofisisco` VARCHAR(45) NULL DEFAULT NULL, PRIMARY KEY (`idTipoespaciofisisco`),
UNIQUE INDEX `tx_tipoespaciofisisco` UNIQUE (`tx_tipoespaciofisisco` ASC)) ENGINE = InnoDB
AUTO INCREMENT = 5
DEFAULT CHARACTER SET = utf8;

-- Table `mydb`.`espaciosfisiscos`

CREATE TABLE IF NOT EXISTS `mydb`.`espaciosfisiscos` (
`idEspaciosFisiscos` INT(11) NOT NULL AUTO INCREMENT,
`tx_nombre_ef` VARCHAR(60) NULL DEFAULT NULL,
`tx_Capacidad` INT(10) UNSIGNED NULL DEFAULT NULL,
`id_tipoef` INT(11) NULL DEFAULT NULL,
`descripcion` VARCHAR(65) NULL DEFAULT NULL, PRIMARY KEY (`idEspaciosFisiscos`),
INDEX `tipoid_idx` (`id_tipoef` ASC), CONSTRAINT `tipoid`
FOREIGN KEY (`id_tipoef`)
REFERENCES `mydb`.`tipoespaciofisisco` (`idTipoespaciofisisco`)

ON DELETE NO ACTION ON UPDATE NO ACTION)

ENGINE = InnoDB AUTO INCREMENT = 11

DEFAULT CHARACTER SET = utf8;

-- Table `mydb`.`asignados`

CREATE TABLE IF NOT EXISTS `mydb`.`asignados` (

`idasignados` INT(11) NOT NULL AUTO INCREMENT,

`fecha` A` DATE NULL DEFAULT NULL,

`txt_evento` A` VARCHAR(85) NULL DEFAULT NULL,

`id_IF` INT(11) NULL DEFAULT NULL,

`id_ef` A` INT(11) NULL DEFAULT NULL,

PRIMARY KEY (`idasignados`), INDEX `IF_idx` (`id_IF` ASC), INDEX `EFS_idx` (`id_ef` A` ASC),

CONSTRAINT `IF`

FOREIGN KEY (`id_IF`)

REFERENCES `mydb`.`in_fn` (`idIn Fn`)

ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `EFS`

FOREIGN KEY (`id_ef` A`)

REFERENCES `mydb`.`espaciosfisicos` (`idEspaciosFisicos`)

ON DELETE NO ACTION ON UPDATE NO ACTION)

ENGINE = InnoDB AUTO INCREMENT = 78

DEFAULT CHARACTER SET = utf8;

-- Table `mydb`.`poll`

```
-----  
CREATE TABLE IF NOT EXISTS `mydb`.`poll` (  
  `idpoll` INT(11) NOT NULL AUTO INCREMENT,  
  `txtpoll` VARCHAR(125) NOT NULL,  
  `clave` VARCHAR(18) NOT NULL,  
  `estado` INT(11) NOT NULL, PRIMARY KEY (`idpoll`))  
ENGINE = InnoDB AUTO INCREMENT = 3  
DEFAULT CHARACTER SET = utf8;
```

```
-----  
-- Table `mydb`.`quest`  
-----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`quest` (  
  `idquest` INT(11) NOT NULL AUTO INCREMENT,  
  `txtquest` VARCHAR(135) NOT NULL,  
  `id_poll` INT(11) NOT NULL, PRIMARY KEY (`idquest`),  
  INDEX `pq_idx` (`id_poll` ASC), CONSTRAINT `pq`  
  FOREIGN KEY (`id_poll`)  
  REFERENCES `mydb`.`poll` (`idpoll`)  
  ON DELETE CASCADE ON UPDATE CASCADE)  
ENGINE = InnoDB AUTO INCREMENT = 7  
DEFAULT CHARACTER SET = utf8;
```

```
-----  
-- Table `mydb`.`answer`  
-----
```

```

-----
CREATE TABLE IF NOT EXISTS `mydb`.`answer` (
  `idoption` INT(11) NOT NULL AUTO INCREMENT,
  `txtoption` VARCHAR(85) NOT NULL,
  `id_quest` INT(11) NOT NULL,
  `valor` INT(11) NULL DEFAULT '0',
  PRIMARY KEY (`idoption`),
  INDEX `go_idx` (`id_quest` ASC), CONSTRAINT `go`
  FOREIGN KEY (`id_quest`)
  REFERENCES `mydb`.`quest` (`idquest`)
  ON DELETE CASCADE ON UPDATE CASCADE)
ENGINE = InnoDB AUTO INCREMENT = 19
DEFAULT CHARACTER SET = utf8;

```

```

-----
-- Table `mydb`.`encuestas`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`encuestas` (
  `id` INT(11) NOT NULL AUTO INCREMENT,
  `titulo` VARCHAR(50) NOT NULL,
  `fecha` DATE NOT NULL, PRIMARY KEY (`id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `mydb`.`statusespaciofisico`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`statusespaciofisico` (
  `idStatusespaciofisico` INT(11) NOT NULL AUTO INCREMENT,
  `tx_status_efisicos` VARCHAR(30) NULL DEFAULT NULL, PRIMARY KEY (`idStatusespaciofisico`),
  UNIQUE INDEX `tx_statusefisicos` UNIQUE (`tx_status_efisicos` ASC)) ENGINE = InnoDB
  DEFAULT CHARACTER SET = utf8;

```

```

-----
-- Table `mydb`.`horarioespaciofisico`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`horarioespaciofisico` (
  `idhorarioef` INT(11) NOT NULL AUTO INCREMENT,
  `Hora_inicio` TIME NULL DEFAULT NULL,
  `Hora_final` TIME NULL DEFAULT NULL,

  `dia` DATE NULL DEFAULT NULL,
  `id_statusef` INT(11) NULL DEFAULT NULL,
  `fecha` DATE NULL DEFAULT NULL,
  `semana` INT(11) NULL DEFAULT NULL,
  `id_ef` INT(11) NULL DEFAULT NULL,
  PRIMARY KEY (`idhorarioef`), INDEX `idef_idx` (`id_ef` ASC),
  INDEX `idstatus_idx` (`id_statusef` ASC), CONSTRAINT `idef1`
  FOREIGN KEY (`id_ef`)
  REFERENCES `mydb`.`espaciosfisicos` (`idEspaciosFisicos`)
  ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `idstatus1`
  FOREIGN KEY (`id_statusef`)

```

REFERENCES `mydb`.`statusespaciofisico` (`idStatusespaciofisico`)

ON DELETE NO ACTION ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- Table `mydb`.`tipousuario`

CREATE TABLE IF NOT EXISTS `mydb`.`tipousuario` (

`idTipousuario` INT(11) NOT NULL AUTO INCREMENT,

`tx tipousuario` VARCHAR(45) NOT NULL, PRIMARY KEY (`idTipousuario`),

UNIQUE INDEX `idTipousuario_UNIQUE` (`idTipousuario` ASC), UNIQUE INDEX

`tx tipousuario_UNIQUE` (`tx tipousuario` ASC))

ENGINE = InnoDB AUTO INCREMENT = 5

DEFAULT CHARACTER SET = utf8;

-- Table `mydb`.`permisosusuarios`

CREATE TABLE IF NOT EXISTS `mydb`.`permisosusuarios` (

`idPermisosusuarios` INT(11) NOT NULL AUTO INCREMENT,

`id tipousuario` INT(11) NULL DEFAULT NULL,

`id accionseccion` INT(11) NULL DEFAULT NULL, PRIMARY KEY (`idPermisosusuarios`),

INDEX `id tipousuario_p_idx` (`id tipousuario` ASC), INDEX `id accionseccion p_idx`

(`id accionseccion` ASC), CONSTRAINT `id accionseccion p`

FOREIGN KEY (`id accionseccion`)

REFERENCES `mydb`.`accionseccion` (`idaccionseccion`)

ON DELETE NO ACTION ON UPDATE NO ACTION,

```
CONSTRAINT `id_tipousuario_p` FOREIGN KEY (`id_tipousuario`)  
REFERENCES `mydb`.`tipousuario` (`idTipousuario`)  
ON DELETE NO ACTION ON UPDATE NO ACTION)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

```
-----  
-- Table `mydb`.`preguntas`  
-----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`preguntas` (  
`idPreguntas` INT(11) NOT NULL AUTO INCREMENT,  
`id_encu` INT(11) NULL DEFAULT NULL,  
`tx texto` VARCHAR(45) NULL DEFAULT NULL,  
PRIMARY KEY (`idPreguntas`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

```
-----  
-- Table `mydb`.`statusolicitud`  
-----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`statusolicitud` (  
`idStatusolicitud` INT(11) NOT NULL AUTO INCREMENT,  
`tx statusolicitud` VARCHAR(45) NULL DEFAULT NULL, PRIMARY KEY (`idStatusolicitud`))  
ENGINE = InnoDB AUTO INCREMENT = 4  
DEFAULT CHARACTER SET = utf8;
```

```

-----
-- Table `mydb`.`statususuario`
-----

CREATE TABLE IF NOT EXISTS `mydb`.`statususuario` (
  `idStatususuario` INT(11) NOT NULL AUTO INCREMENT,
  `tx_statususuario` VARCHAR(45) NOT NULL, PRIMARY KEY (`idStatususuario`),
  UNIQUE INDEX `tx_statususuario_UNIQUE` (`tx_statususuario` ASC), UNIQUE INDEX
  `idStatususuario_UNIQUE` (`idStatususuario` ASC))
ENGINE = InnoDB AUTO INCREMENT = 3
DEFAULT CHARACTER SET = utf8;

```

```

-----
-- Table `mydb`.`usuario`
-----

CREATE TABLE IF NOT EXISTS `mydb`.`usuario` (
  `idUsuario` INT(11) NOT NULL AUTO INCREMENT,
  `username` VARCHAR(45) NOT NULL,
  `password` VARCHAR(45) NOT NULL,
  `nombreusuario` VARCHAR(45) NOT NULL,
  `ap_paterno` VARCHAR(45) NULL DEFAULT NULL,
  `ap_materno` VARCHAR(45) NULL DEFAULT NULL,
  `email` VARCHAR(60) NULL DEFAULT NULL,
  `telefono` VARCHAR(25) NULL DEFAULT NULL,
  `fecharegistro` DATETIME NOT NULL,
  `id_tipousuario` INT(11) NULL DEFAULT NULL,
  `id_statususuario` INT(11) NULL DEFAULT NULL, PRIMARY KEY (`idUsuario`),

```

UNIQUE INDEX `idUsuario` UNIQUE` (`idUsuario` ASC), UNIQUE INDEX `user` UNIQUE` (`username` ASC),

INDEX `id_tipousuario_idx` (`id_tipousuario` ASC), INDEX `id_statususuario_idx` (`id_statususuario` ASC), CONSTRAINT `id_statususuario`

FOREIGN KEY (`id_statususuario`)

REFERENCES `mydb`.`statususuario` (`idStatususuario`)

ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `id_tipousuario` FOREIGN KEY (`id_tipousuario`)

REFERENCES `mydb`.`tipousuario` (`idTipousuario`)

ON DELETE NO ACTION ON UPDATE NO ACTION)

ENGINE = InnoDB AUTO INCREMENT = 11

DEFAULT CHARACTER SET = utf8;

-- Table `mydb`.`solicitud`

CREATE TABLE IF NOT EXISTS `mydb`.`solicitud` (

`idReservaciones` INT(11) NOT NULL AUTO INCREMENT,

`hora ini res` TIME NOT NULL,

`idstatusolicitud` INT(11) NOT NULL,

`fecha` DATE NOT NULL,

`hora fin res` TIME NOT NULL,

`id_ef_solicitud` INT(11) NOT NULL,

`descripcionevento` VARCHAR(500) NOT NULL,

`id_usuariosol` INT(11) NOT NULL,

`tx_nombre_evento` VARCHAR(65) NOT NULL,

```

`clave_sol` VARCHAR(20) NULL DEFAULT NULL,
PRIMARY KEY (`idReservaciones`),
UNIQUE INDEX `clave_sol` (`clave_sol` ASC), INDEX `idstatus_idx` (`idstatusolicitud` ASC), INDEX
`id_ef_idx` (`id_ef_solicitud` ASC),
INDEX `id_user_sol_idx` (`id_usuariosol` ASC), CONSTRAINT `idstatus`
FOREIGN KEY (`idstatusolicitud`)
REFERENCES `mydb`.`statusolicitud` (`idStatusolicitud`)
ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT `id_ef`
FOREIGN KEY (`id_ef_solicitud`)
REFERENCES `mydb`.`espaciosfisicos` (`idEspaciosFisicos`)
ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT `id_user_sol`
FOREIGN KEY (`id_usuariosol`)
REFERENCES `mydb`.`usuario` (`idUsuario`)
ON DELETE NO ACTION ON UPDATE NO ACTION)
ENGINE = InnoDB AUTO INCREMENT = 8
DEFAULT CHARACTER SET = utf8;

```

```

-----
-- Table `mydb`.`sol rechazadas`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`sol rechazadas` (
`id Sol Rechazadas` INT(11) NOT NULL AUTO INCREMENT,
`id idEspaciosFisicos` INT(11) NOT NULL,
`id idReservaciones` INT(11) NOT NULL,
`tx Motivo rechazo` VARCHAR(400) NULL DEFAULT NULL, PRIMARY KEY (`id Sol Rechazadas`),
INDEX `id idEF` (`id idEspaciosFisicos` ASC), INDEX `id idRes` (`id idReservaciones` ASC),
CONSTRAINT `id idEF`

```

```
FOREIGN KEY (`id_idEspaciosFisicos`)  
REFERENCES `mydb`.`espaciosfisicos` (`idEspaciosFisicos`)  
ON DELETE NO ACTION ON UPDATE NO ACTION,  
CONSTRAINT `id_idRes`  
FOREIGN KEY (`id_idReservaciones`)  
REFERENCES `mydb`.`solicitud` (`idReservaciones`)  
ON DELETE NO ACTION ON UPDATE NO ACTION)  
ENGINE = InnoDB AUTO INCREMENT = 3  
DEFAULT CHARACTER SET = utf8;
```

```
-----  
-- Table `mydb`.`encuestacontestada`  
-----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`encuestacontestada` (  
`idencuestacontestada` INT NOT NULL AUTO INCREMENT,  
`clave_encuesta` VARCHAR(18) NULL,  
`id_user` INT NULL,  
PRIMARY KEY (`idencuestacontestada`), INDEX `ecus_idx` (`id_user` ASC), CONSTRAINT `ecus`  
FOREIGN KEY (`id_user`)  
REFERENCES `mydb`.`usuario` (`idUsuario`)  
ON DELETE NO ACTION ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
SET SQL MODE=@OLD SQL MODE;
```

SET FOREIGN KEY CHECKS=@OLD FOREIGN KEY CHECKS; SET
UNIQUE CHECKS=@OLD UNIQUE CHECKS;