



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería Mecánica Eléctrica

## **SISTEMA DE ASIGNACIÓN DE LABORATORIOS DE ELECTRÓNICA EN LÍNEA**

**Edgar Josué Gómez Salazar**

Asesorado por la Inga. Ingrid Rodríguez de Loukota

Guatemala, agosto de 2015

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**SISTEMA DE ASIGNACIÓN DE LABORATORIOS  
DE ELECTRÓNICA EN LÍNEA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA

POR

**EDGAR JOSUÉ GÓMEZ SALAZAR**

ASESORADO POR LA INGA. INGRID RODRÍGUEZ DE LOUKOTA

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN ELECTRÓNICA**

GUATEMALA, AGOSTO DE 2015

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Narda Lucía Pacay Barrientos
VOCAL V	Br. Walter Rafael Véliz Muñoz
SECRETARIA	Inga. Lesbia Magalí Herrera López

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Murphy Olympto Paiz Recinos
EXAMINADOR	Ing. Armando Alomzo Rivera Carrillo
EXAMINADOR	Ing. Byron Odilio Arrivillaga Méndez
EXAMINADORA	Inga. María Magdalena Puente Romero
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

### **SISTEMA DE ASIGNACIÓN DE LABORATORIOS DE ELECTRÓNICA EN LÍNEA**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 30 de julio de 2013.



**Edgar Josue Gomez Salazar**

Guatemala 6 de julio de 2015

Ingeniero  
Carlos Eduardo Guzmán Salazar  
Coordinador del Área de Electrónica  
Escuela de Ingeniería Mecánica Eléctrica  
Facultad de Ingeniería, USAC.

Estimado Ingeniero Guzmán.

Me permito dar aprobación al trabajo de graduación titulado: **SISTEMA DE ASIGNACIÓN DE LABORATORIOS DE ELECTRÓNICA EN LÍNEA**, del señor **Edgar Josué Gómez Salazar**, por considerar que cumple con los requisitos establecidos.

Por tanto, el autor de este trabajo de graduación y, yo, como su asesora, nos hacemos responsables por el contenido y conclusiones del mismo.

Sin otro particular, me es grato saludarle.

Atentamente,



Inga. Ingrid Rodríguez de Loukota  
Colegiada 5,356  
Asesora

Ingrid Rodríguez de Loukota  
Ingeniera en Electrónica  
colegiado 5356



Ref. EIME 45. 2015

Guatemala, 14 de julio 2015.

Señor Director

Ing. Guillermo Antonio Puente Romero  
Escuela de Ingeniería Mecánica Eléctrica  
Facultad de Ingeniería, USAC.

Señor Director:

Me permito dar aprobación al trabajo de Graduación titulado:  
SISTEMA DE ASIGNACIÓN DE LABORATORIOS DE  
ELECTRÓNICA EN LÍNEA, del estudiante Edgar Josué Gómez  
Salazar que cumple con los requisitos establecidos para tal fin.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,  
DID Y ENSEÑAD A TODOS

Ing. Carlos Eduardo Guzmán Salazar  
Coordinador Área Electrónica



SFO



REF. EIME 45. 2015.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; EDGAR JOSUÉ GÓMEZ SALAZAR titulado: SISTEMA DE ASIGNACIÓN DE LABORATORIOS DE ELECTRÓNICA EN LÍNEA, procede a la autorización del mismo.

Ing. Guillermo Antonio Puente Romero

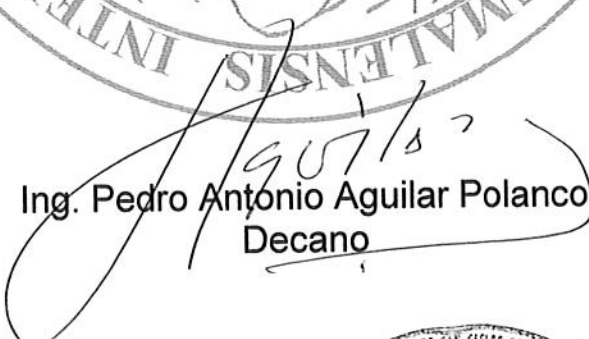


GUATEMALA, 22 DE JULIO 2,015.



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica al trabajo de graduación titulado: **SISTEMA DE ASIGNACIÓN DE LABORATORIOS DE ELECTRONICA EN LÍNEA**, presentado por el estudiante universitario: **Edgar Josué Gómez Salazar**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE.

  
Ing. Pedro Antonio Aguilar Polanco  
Decano

Guatemala, agosto de 2015





## **ACTO QUE DEDICO A:**

### **Mis padres**

Sandra Salazar y Vinicio Gómez, por el enorme esfuerzo, porque cada día lucharon para que esto fuera posible, por tomarme de la mano y acompañarme hasta aquí.

### **Mi hermano, padrino y compadre**

Marco Vinicio Gómez Salazar, por ser esa figura de admiración en mi vida, por creer en mí y en lo que soy capaz, y por alentarme hasta el fin.

### **Mi novia**

Ana Cristina Valencia, por el amor, la paciencia, la dedicación y el apoyo, pero sobre todo por ser mi compañera de viaje.

### **Lisette Mendoza, Adita Mora y Montserrat Solano**

Por haber venido a darle tantas alegrías a nuestra familia, por el amor que nos dan y por ser tan importantes para mí.

### **Mi familia**

Por el apoyo y por creer en mí.

## **AGRADECIMIENTOS A:**

<b>Dios</b>	Porque sin Él, nada de esto estaría sucediendo, por permitirme llegar a este día, y por abrirme las puertas del mundo.
<b>Universidad de San Carlos de Guatemala</b>	Por ser mi alma máter, mi casa de estudios, mi hogar durante todos estos años.
<b>Facultad de Ingeniería</b>	Por hacer realidad mis ilusiones, y exigirme al máximo para llegar a ser un profesional
<b>Mis más que amigos</b>	Sergio Fuentes, Carlos Aguilar, Guille Pineda y Omar Chacón, por ser compañeros de desvelos, metas y esfuerzos.
<b>Inga. Ingrid de Loukota</b>	Por asesorarme y compartirme su vasta experiencia durante toda la carrera y en la finalización de la misma.
<b>Oscar Vicente Carrillo</b>	Por su apoyo incondicional en el desarrollo de este proyecto.
<b>Todos mis amigos y conocidos</b>	Que han sido parte de este gran logro y que estuvieron ahí para apoyarme.

# ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES .....	V
LISTA DE SÍMBOLOS .....	IX
GLOSARIO .....	XI
RESUMEN .....	XIII
OBJETIVOS.....	XV
INTRODUCCIÓN .....	XVII
1.    CONCEPTOS DE APLICACIONES WEB .....	1
1.1.    Sitio web .....	1
1.1.1.    Contenido .....	1
1.1.2.    Estructura física .....	2
1.1.3.    Estándar de nombre de archivos y carpetas .....	3
1.1.4.    Hipervínculos .....	3
1.2.    Lenguajes de desarrollo web .....	5
1.2.1.    Python .....	5
1.2.2.    Ruby .....	6
1.2.3.    HTML.....	7
1.2.4.    JSP .....	9
1.2.5.    PHP .....	11
1.2.6.    ASP.NET .....	12
1.3.    Servicios web.....	14
1.3.1.    Servicios web 1.0 .....	15
1.3.2.    Servicios web 2.0 .....	15
1.3.3.    Aplicaciones en la nube, <i>Cloud Computing</i> .....	17
1.4.    Servidores de aplicaciones web .....	18

1.4.1.	Servidor web .....	18
1.4.2.	Servidores de aplicaciones web .....	19
1.4.3.	Servidores de aplicaciones web comerciales .....	20
1.4.3.1.	Apache .....	21
1.4.3.2.	IIS .....	21
1.4.3.3.	Nginx .....	22
1.4.3.4.	Apache Tomcat .....	23
1.5.	Bases de datos y sistemas de gestión .....	23
1.5.1.	Concepto y origen de las BD y SGBD .....	24
1.5.2.	Objetivos y servicios de los SGBD .....	25
1.5.2.1.	Consultas no predefinidas y complejas .....	25
1.5.2.2.	Flexibilidad e independencia .....	26
1.5.2.3.	Integridad de los datos .....	27
1.5.2.4.	Concurrencia de usuarios.....	28
1.5.2.5.	Seguridad .....	29
1.5.3.	Modelos de BD.....	30
1.5.4.	Modelos y usuarios .....	32
1.5.5.	Tipos de datos.....	33
1.6.	<i>Framework</i> web .....	34
2.	TECNOLOGÍAS APLICABLES EN EL PROYECTO .....	35
2.1.	Diseño web adaptativo .....	35
2.1.1.	<i>Template</i> de rejilla fluida .....	36
2.1.2.	Imágenes fluidas .....	38
2.1.3.	<i>Media types</i> y <i>media queries</i> .....	40
2.2.	Arquitectura MVC (Modelo-Vista-Controlador) .....	42
2.2.1.	Patrón MVC en PHP .....	44
2.2.2.	Modelo MVC en ASP.NET .....	46

2.3.	ADO.NET Entity Framework.....	50
2.3.1.	Arquitectura y componentes .....	51
2.3.2.	Proveedores específicos .....	52
2.3.3.	Modelo de entidades .....	53
2.4.	Motor de vistas “Razor” .....	53
2.5.	Jquery.....	54
3.	EL SISTEMA ANTIGUO Y EL NUEVO .....	57
3.1.	Antecedentes.....	57
3.1.1.	Descripción del proceso .....	59
3.1.2.	El sistema de asignaciones .....	59
3.1.2.1.	El equipo.....	60
3.1.2.2.	La base de datos de asignaciones .....	60
3.1.2.3.	La aplicación de asignaciones.....	62
3.1.2.3.1.	Submenú de configuración .....	63
3.1.2.3.2.	Submenú de ingresos...	68
3.1.2.3.3.	Submenú de reportes ...	74
3.2.	El nuevo sistema .....	79
3.2.1.	Descripción del nuevo proceso.....	79
3.2.2.	El nuevo sistema de asignación .....	81
3.2.2.1.	El equipo.....	82
3.2.2.2.	La nueva base de datos .....	83
3.2.2.3.	La nueva aplicación de asignaciones ..	85
3.2.2.3.1.	Acceso como usuario restringido.....	88
3.2.2.3.2.	Acceso como auxiliar....	92
3.2.2.3.3.	Acceso como administrador.....	100

4.	IMPLEMENTACIÓN FINAL.....	103
4.1.	Cumplimiento de requisitos de sistema.....	103
4.1.1.	Cumplimiento de requisitos de hardware .....	103
4.1.2.	Cumplimiento de requisitos de software.....	105
4.2.	Configuración del servidor.....	107
4.2.1.	Configuración de base de datos.....	107
4.2.2.	Configuración de IIS.....	113
4.2.3.	Configuración de <i>firewall</i> local.....	119
4.3.	Pruebas al sistema.....	122
4.3.1.	Pruebas de rendimiento .....	122
4.3.2.	Escaneo de puertos .....	124
	CONCLUSIONES.....	127
	RECOMENDACIONES .....	129
	BIBLIOGRAFÍA.....	131
	ANEXOS.....	133

# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1.	Diferentes tipos de estructuras físicas.....	2
2.	Hipervínculos en un sitio web.....	4
3.	Ejemplo de código fuente Python.....	6
4.	Ejemplo “Hola Mundo” con Ruby .....	7
5.	Estructura básica de HTML.....	9
6.	Esquema de aplicación usando <i>servlets</i> y JSP .....	10
7.	Ejemplo básico de una página JSP.....	11
8.	Ejemplo de Programación PHP básico .....	12
9.	Diagrama de funcionamiento de las páginas ASP .....	13
10.	Ejemplo de página web básica con ASP.....	13
11.	Funcionamiento de un servidor de aplicaciones PHP .....	20
12.	Vista del sitio por defecto de IIS.....	22
13.	Relación de base de datos y su sistema de gestión .....	24
14.	Modelos de base de datos .....	31
15.	Ejemplo de un sitio web adaptativo.....	36
16.	Ejemplo de hoja de estilo con pixeles limitados .....	37
17.	Ejemplo de hoja de estilo con altura fija.....	38
18.	Parte del CSS que fija imágenes al 100 % del contenedor .....	39
19.	Código en <i>head</i> para funciones las imágenes fluidas .....	39
20.	<i>Head</i> con <i>media types</i> .....	40
21.	Ejemplo de CSS con media queries.....	41
22.	Inclusión de la hoja de estilos en el HTML.....	41
23.	Diagrama de flujo MVC .....	43

24.	Generación e implementación de la interface .....	44
25.	Modelo Vista Controlador Entity Framework.....	50
26.	Componentes principales de ADO.NET Entity Framework.....	52
27.	Diagrama del proceso de asignación de laboratorios .....	58
28.	Ventana de acceso al sistema antiguo .....	63
29.	Opciones del submenú de configuración .....	64
30.	Administración de cursos.....	65
31.	Administración de semestres .....	65
32.	Administración de rubros .....	66
33.	Administración de horarios .....	67
34.	Cambio de contraseñas .....	68
35.	Submenú de ingresos .....	69
36.	Ingreso de estudiantes.....	70
37.	Asignación de cursos.....	71
38.	Ingreso de notas por curso .....	72
39.	Ingreso de revalidaciones autorizada .....	73
40.	Submenú de reportes .....	74
41.	Asignaciones por día .....	75
42.	Asignaciones por alumno.....	76
43.	Grupos por curso .....	77
44.	Conteo de asignaciones .....	78
45.	Diagrama del nuevo proceso de asignaciones .....	80
46.	Página de inicio sitio web de asignaciones.....	86
47.	Registro de nuevos usuarios .....	87
48.	Menú de estudiante .....	88
49.	Asignación de alumnos.....	89
50.	Ver notas por semestre.....	90
51.	Calendario de laboratorios.....	91
52.	Cambio de contraseña.....	92



53.	Menú de auxiliar .....	92
54.	Administración de usuarios .....	93
55.	Menú de configuración .....	94
56.	Lista de cursos .....	94
57.	Crear curso .....	95
58.	Lista de semestres .....	96
59.	Crear semestre .....	96
60.	Lista de rubros .....	97
61.	Crear rubros .....	97
62.	Lista de horarios.....	98
63.	Crear horario .....	99
64.	Ingreso de notas .....	99
65.	Reporte de notas.....	100
66.	Lista de usuarios por role .....	101
67.	Buscar un rol.....	101
68.	Buscar usuario .....	102
69.	Cumplimiento de requisitos de ram y procesador .....	104
70.	Cumplimiento de requisitos de disco duro.....	104
71.	Sistema operativo.....	105
72.	Verificación de IIS .....	106
73.	Verificación de componentes de base de datos .....	107
74.	Explorador de objetos .....	108
75.	Restaurar base de datos .....	109
76.	Base de datos ya restaurada .....	110
77.	Crear nuevo usuario.....	111
78.	Permisos sobre la base de datos .....	112
79.	Acceso como ELAB.....	113
80.	Roles servidor .....	115
81.	Características de servidor.....	116

82.	Directorio wwwroot.....	117
83.	Permisos de IIS.....	118
84.	Vista local del sitio .....	119
85.	Estado de <i>firewall</i> .....	120
86.	Regla de tráfico HTTP .....	121
87.	Sitio web público .....	122

## TABLAS

I.	Tabla de comparación Web 1,0 vs Web 2,0 .....	17
II.	Especificaciones mínimas del equipo que aloja el sistema.....	60
III.	Especificaciones mínimas de un servidor web .....	82
IV.	Requisitos de hardware .....	103
V.	Requisitos de software.....	105
VI.	Prueba de rendimiento.....	123
VII.	Escaneo de puertos del sitio .....	125

## LISTA DE SÍMBOLOS

<b>Símbolo</b>	<b>Significado</b>
<b>BD</b>	Base de datos
<b>CSS</b>	Hoja de estilos en cascada utilizado para presentación de sitios web.
<b>KB</b>	<i>Kilo Bytes</i>
<b>FTP</b>	Protocolo de transferencia de archivos
<b>SQL</b>	Lenguaje de acceso a base de datos
<b>Mbps</b>	Taza de transferencia de datos (megabits por segundo).
<b>MVC</b>	Marco de trabajo que utiliza el Modelo-Vista-Controlador.
<b>S</b>	Segundos



## GLOSARIO

<b><i>Applet</i></b>	Componente de una aplicación que se ejecuta en el contexto de otro programa.
<b><i>Caching</i></b>	Acto de almacenar en memoria caché algún componente que será utilizado posteriormente.
<b><i>Drag and Drop</i></b>	Acción de mover con el ratón objetos de una ventana a otra o entre partes de una misma ventana.
<b>Encriptar</b>	Proceso de volver ilegible información considerada importante y únicamente será legible con una clave.
<b>Linux</b>	Término utilizado para referirse a la combinación del núcleo libre similar a Unix.
<b>Loguear</b>	Conectarse en una página web o sistema informático de forma que el usuario quede identificado.
<b><i>Padding</i></b>	Establece la anchura de algunas o todas las zonas de relleno de elementos web.
<b>Prototipado</b>	Acción de crear prototipos.
<b>Proxy</b>	Servidor que sirve de intermediario en las peticiones de recursos que realiza un cliente a otro servidor.

<b>Stand-Alone</b>	Aplicación que puede ser utilizada por una computadora sin necesidad de depender de un servidor.
<b>SGBD</b>	Sistema de gestión de bases de datos.
<b>SMTP</b>	Protocolo simple de transferencia de correo.
<b>SO</b>	Sistema operativo.
<b>Usabilidad</b>	Es la facilidad con la que personas pueden utilizar una herramienta en particular.
<b>Windows</b>	Sistema operativo de la empresa Microsoft comercializado desde 1985.
<b>Wizard</b>	Interface asistente para guiar la instalación o configuración de un sistema o programa.

## **RESUMEN**

En el presente trabajo de graduación se documenta el proceso de diseño e implementación del sistema de asignación de laboratorios de electrónica en línea.

En el primer capítulo se aclaran los conceptos necesarios sobre aplicaciones web, los cuales serán utilizados en el desarrollo de este trabajo de graduación, estos son necesarios para comprender cada una de las aristas del mismo.

En el segundo capítulo, se detalla cada una de las tecnologías que serán utilizadas para el diseño, desarrollo e implementación del sistema, así como, el razonamiento para la elección de las mismas.

En el tercer capítulo se presenta el sistema antiguo de asignación y se realiza una comparación con el nuevo sistema, mostrando los beneficios adquiridos, dirigido tanto al personal de laboratorio, como a los estudiantes.

En el cuarto capítulo, se documenta la implementación del nuevo sistema de asignación de laboratorios en línea, así como las pruebas de rendimiento del sistema y una prueba final de vulnerabilidad y seguridad del mismo.





## **OBJETIVOS**

### **General**

Implementar un sistema de asignación de laboratorios en línea, desarrollado para el Laboratorio de Electrónica de la Escuela de Ingeniería Mecánica Eléctrica.

### **Específicos**

1. Presentar los conceptos que rodean a los aplicativos web, para facilitar la comprensión de la necesidad de migración del sistema antiguo.
2. Presentar las tecnológicas que son aplicables al nuevo sistema, y porque este fue desarrollado sobre las mismas.
3. Presentar el sistema antiguo de asignaciones, con sus fortalezas y debilidades, para luego ser comparado con el nuevo sistema, haciendo énfasis en la usabilidad del mismo y el beneficio asociado.
4. Documentar la implementación del sistema de asignación de laboratorios en línea, para facilitar su posterior administración y mantenimiento.



## INTRODUCCIÓN

Debido a la dificultad del proceso de asignación de laboratorios de los cursos de la carrera de Ingeniería Electrónica, y debido a que dentro del proceso de asignación es necesario que cada alumno encuentre su respectivo grupo en el momento de la asignación, se ha identificado la necesidad de actualizar este proceso a las tecnologías actuales, las cuales que permiten mayor flexibilidad en el método de asignación, beneficiando así al alumnado y a su vez a el equipo de auxiliares de laboratorio.

En el presente trabajo de graduación se muestra el razonamiento de la renovación del proceso de asignación y el desarrollo de un nuevo sistema que permite el seguimiento de dicho proceso, facilitando también la utilización en todas sus facetas.

Este sistema de asignación de laboratorios en línea, se desarrolló en un ambiente web y reside en un servidor al cual tiene acceso el personal de laboratorio para su adecuada administración, esta administración es mínima ya que el sistema fue desarrollado de manera que el mismo es capaz de manejar por sí solo, la mayoría de los pasos del proceso de asignación.



# **1. CONCEPTOS DE APLICACIONES WEB**

## **1.1. Sitio web**

Realizando la investigación y teniendo varios conceptos en debate, se puede decir que “Un sitio web es un conjunto de páginas web relacionadas entre sí. Se entiende por página web tanto el fichero que contiene el código HTML como todos los recursos que se emplean en la página (imágenes, sonidos, Código JavaScript, entre otros).

### **1.1.1. Contenido**

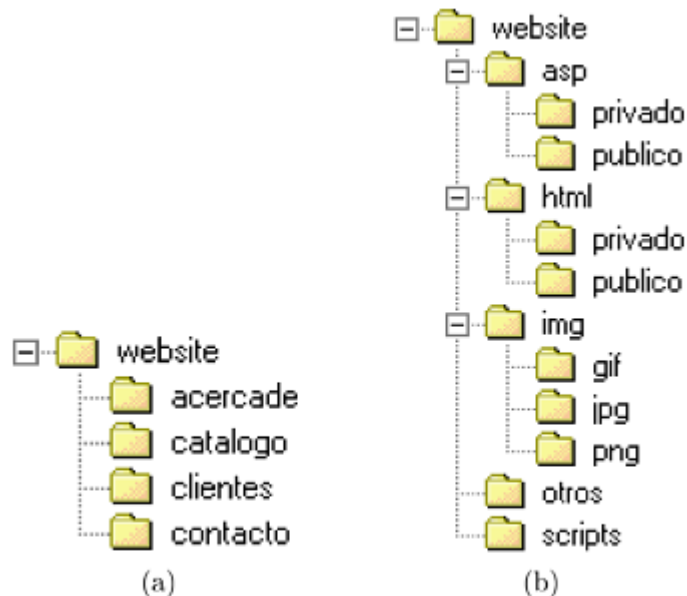
Se necesita definir lo que debe estar contenido en un sitio web, el contenido web suele organizarse en una serie de secciones para facilitar la búsqueda y localización, en este caso, es el de un centro educativo por lo cual el sitio web debe contar como mínimo con lo siguiente:

- Personal
- Profesores
- Estudiantes
- Planes de estudio
- Noticias
- Otros

### 1.1.2. Estructura física

La estructura física de un sitio web es la forma en la que se almacenan los recursos que forman parte del mismo, esta estructura indica la manera en la que se organizarán las carpetas y subcarpetas desde donde se recogerán los ficheros, imágenes, videos, entre otros. No existe una manera mejor o peor de organizar las carpetas dentro de un servidor web, la manera que estas carpetas se estructuren, está relacionada con el estándar que el desarrollador lo prefiera y así también con la manera que le sea más sencillo encontrar los objetos y direccionarlos dentro del código, se puede organizar por el tipo de archivo que contienen, o por la parte del sitio web a la que corresponda.

Figura 1. Diferentes tipos de estructuras físicas



Fuente: LUJAN, Sergio. *Programación de aplicaciones web*. p. 66.

### **1.1.3. Estándar de nombre de archivos y carpetas**

Es necesario tener en cuenta el estándar de nombres que se utilizarán en los archivos y directorios del sitio, ya que esto permitirá mantener el orden en el mismo, tanto para el desarrollo como para la navegación por el mismo.

En el momento de nombrar los archivos y carpetas, es importante tener cuidado con los caracteres que se utilizan, debido a que dichos caracteres pueden no ser permitidos en los distintos sistemas operativos, y esto puede generar errores en la navegación por el sitio.

En el caso de los sistemas Windows, estos no distinguen entre mayúsculas y minúsculas, por lo cual los nombres de los ficheros pueden contener caracteres de ambas formas, a diferencia de los sistemas Unix que sí hacen distinción entre mayúsculas y minúsculas, esta ambigüedad puede generar problemas a la hora de acceder a los componentes desde diferentes sistemas operativos, por tanto, se busca convenir en escribir el nombre de todos los elementos en minúsculas para evitar este tipo de problemas.

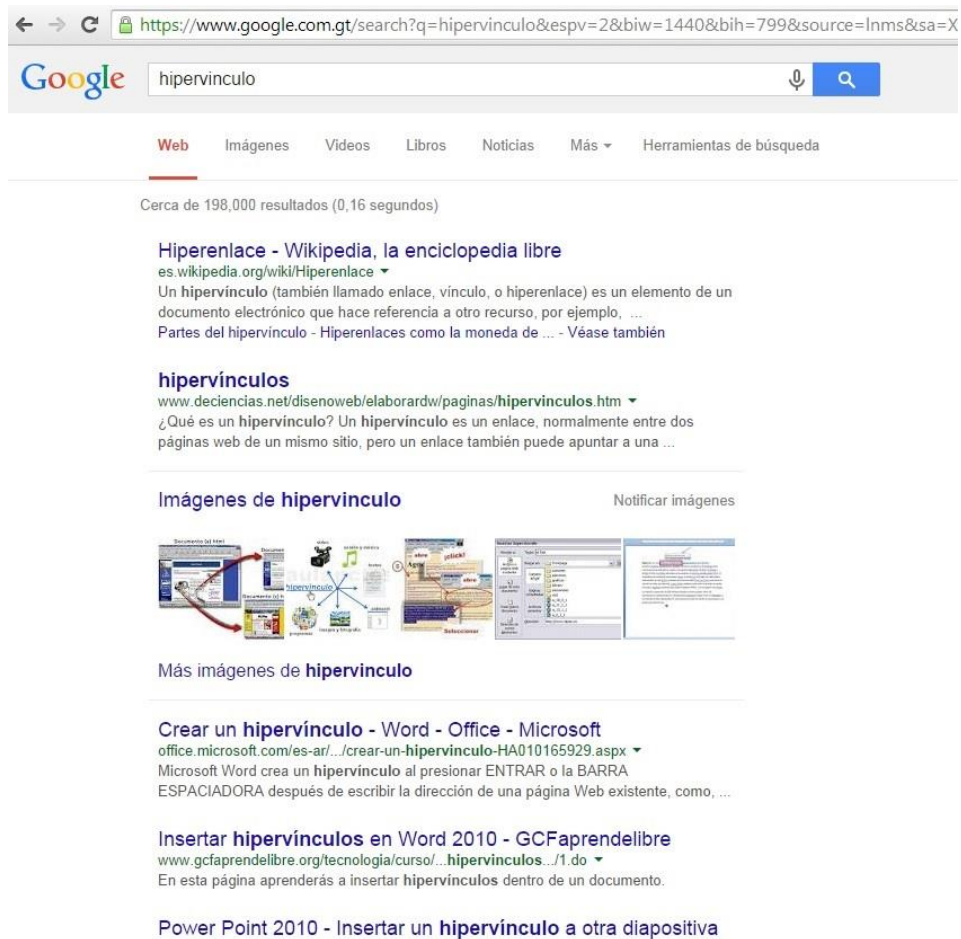
### **1.1.4. Hipervínculos**

Los hipervínculos son los encargados de crear el hipertexto. Existen dos tipos básicos de hipervínculos: los que hacen referencia a recursos del propio sitio web, y lo que hacen referencia a sitios web externos.

Un hipervínculo es de utilidad cuando se necesita referenciar alguna información contenida en un lugar diferente al que se está leyendo y el usuario pueda consultar esta misma después sin perder la atención en el tema mismo.

Los hipervínculos usualmente son introducidos en un sitio web de forma que el usuario únicamente haga clic sobre ellos y así puedan ser dirigidos tanto al sitio en mención, un ejemplo muy claro, es el mostrado en la figura 2, el cual corresponde a una búsqueda en el buscador Google.

Figura 2. **Hipervínculos en un sitio web**



Fuente: elaboración propia.



## **1.2. Lenguajes de desarrollo web**

En la actualidad existen varios lenguajes de desarrollo web, los mismos han surgido paulatinamente siguiendo ciertas tendencias y necesidades tanto de los usuarios como las plataformas, a continuación se detallan cada uno de los lenguajes más utilizados para desarrollo web en la actualidad.

### **1.2.1. Python**

Es un lenguaje de programación diseñado por el holandés Guido van Rossum, a finales de los 80. Python tiene licencia de código abierto (PSFL) compatible con GPL de GNU a partir de la versión 2.1.1.

Se usa en programación de sistemas, cálculo numérico, desarrollo web, software para dispositivos móviles (Symbian, Android), desarrollo de aplicaciones de escritorio, educación, simulación, prototipados, GIS, entre otros.

Es un lenguaje de programación de propósito general, de muy alto nivel, Python es un lenguaje interpretado (no es necesaria compilación), dinámico (no necesita identificar explícitamente los tipos de datos para inicializar variables, de modo que los tipos se validan durante la ejecución del programa) y fuertemente tipado (no pueden mezclarse tipos, es necesario hacer conversiones).

Es un lenguaje multiplataforma (Windows, Mac, Linux, entre otros), multiparadigma (imperativo, orientado a objetos y en menor medida funcional) y con gestión automática de memoria.

Por último cabe destacar que es un lenguaje de programación con una sintaxis clara y sencilla, fácil de aprender, donde se pueden mezclar los diferentes paradigmas de programación de los que dispone, ampliamente documentado, extensible, que intenta obligar al desarrollador de software a programar de la manera correcta en el menor tiempo posible.

Figura 3. **Ejemplo de código fuente Python**

```
def add5(x):
    return x+5

def dotwrite(ast):
    nodename = getNodename()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print '    %s [label="%s' % (nodename, label),
    if isinstance(ast[1], str):
        if ast[1].strip():
            print '= %s"' % ast[1]
        else:
            print ''
    else:
        print ''
        children = []
        for n, childenumerate(ast[1:]):
            children.append(dotwrite(child))
        print ', ' % ast[0] -> {' % nodename
        for n, namechildren:
            print '%s' % name,
```

Fuente: *Código Python con coloreado de sintaxis*. es.wikipedia.com.

### 1.2.2. Ruby

Es un lenguaje de programación interpretado, reflexivo y orientado a objetos, creado por el programador japonés Yukihiro "Matz" Matsumoto, quien comenzó a trabajar en Ruby en 1993, y lo presentó públicamente en 1995. Combina una sintaxis inspirada en Python, Perl con características de programación orientada a objetos similares a Smalltalk. Comparte también funcionalidad con otros lenguajes de programación como Lisp, Lua, Dylan y

CLU. Ruby es un lenguaje de programación interpretado en una sola pasada y su implementación oficial es distribuida bajo una licencia de software libre.

Una de las características especiales de Ruby es que todo es un objeto, se combinan las capacidades de los lenguajes funcionales con las de los lenguajes imperativos orientados a objetos, de esta manera cada elemento en Ruby tiene un compartimiento similar, pero personalizado para las conveniencias de cada usuario.

Figura 4. **Ejemplo “Hola Mundo” con Ruby**

```
1 require 'rubygems'
2 require 'cramp/controller'
3
4 class WelcomeAction < Cramp::Controller::Action
5   on_start :send_hello_world
6
7   def send_hello_world
8     render "Hello World"
9     finish
10  end
11 end
12
13 Rack::Handler::Thin.run WelcomeAction, :Port => 3000
```

Fuente: *Basic “Hello World” example*. [www.rubyinside.com](http://www.rubyinside.com) .

### 1.2.3. HTML

HTML es un lenguaje para desarrollo web, artificial, el cual las computadoras son capaces de interpretar y el cual está diseñado para que los programadores redacten instrucciones que los navegadores ejecutan para originar el sitio web, en otras palabras HTML es un lenguaje de programación, o un “idioma que la máquina entiende y procesa para dar la respuesta”.

Las siglas de HTML significan HyperTextMackupLanguage (lenguaje de máscaras de híper texto). El hipertexto en una computadora es texto que posee referencias a otro texto, dicho concepto surgió debido a la necesidad de los científicos de procesa la información mecánicamente entre computadoras conectadas.

En esencia, HTML sirve para estructurar documentos, pero no describe la apariencia o el diseño de un documento, mas bien ofrece las herramientas necesarias para dar formato según la capacidad del servidor web en el que se almacenan las páginas web y la capacidad del navegador, tamaño de la pantalla, fuentes que tiene instaladas, entre otros.

El lenguaje HTML tiene dos ventajas que lo hacen prácticamente imprescindible a la hora de diseñar una presentación web: su compatibilidad y la facilidad que plantea su aprendizaje debido al reducido número de etiquetas en las que se apoya.

Básicamente la sintaxis de HTML consiste en etiquetas, atributos y los elementos que conforman la página, tanto texto como las imágenes, tablas, gráficos, entre otros. Esto llevado a la práctica, se plasma en comandos estructurados de la forma en que se muestra en la figura 5.

Figura 5. Estructura básica de HTML



Fuente: *Estructura Básica HTML*. [www.ecalli.com](http://www.ecalli.com). Consulta: 23 de abril de 2014.

Este lenguaje ha evolucionado en sus versiones dado que han surgido nuevas etiquetas y así como también han desaparecido algunas otras, actualmente la última versión es la versión HTML5.

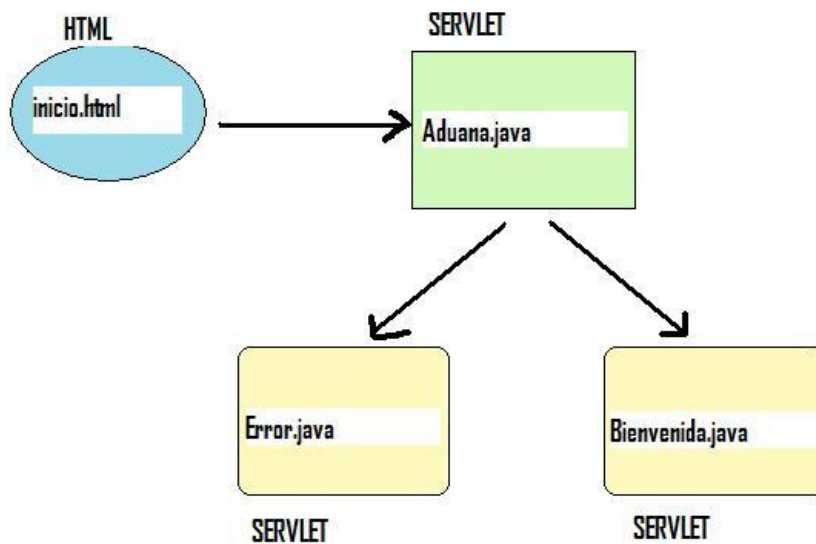
#### 1.2.4. JSP

Es el acrónimo de Java Server Pages, que en español significa Páginas de Servidor Java.

El lenguaje JSP es un lenguaje Java orientado a la conexión cliente-servidor, con la ventaja de que las tecnologías residen del lado del servidor, esto evita que el usuario deba tener instalados componentes específicos para

que funcione cada programa, JSP es similar a ASP y PHP los cuales se verán más adelante. Los *servlets* son equivalentes a los *applets*, pero en el lado del servidor.

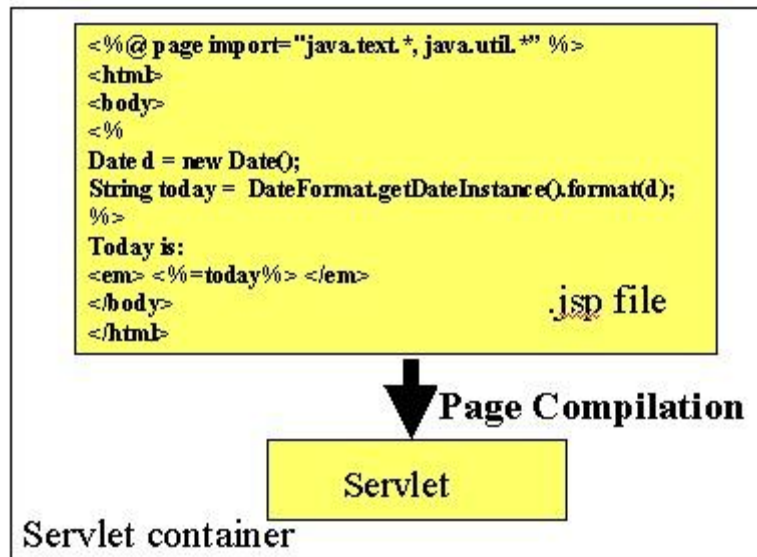
Figura 6. **Esquema de aplicación usando *servlets* y JSP**



Fuente: *La redirección de peticiones con servlets*. programandojava.webnode.es. Consulta: 23 de abril de 2014.

En JSP se crean páginas de manera que se generan archivos con extensión .jsp que incluyen, dentro de la estructura de etiquetas HTML, las sentencias Java a ejecutar en el servidor. Antes de que sean funcionales los archivos, el motor JSP lleva a cabo una fase de traducción de esa página en un *servlet*, implementado en un archivo class. Esta fase de traducción se lleva a cabo habitualmente cuando se recibe la primera solicitud de la pagina.jsp, aunque existe la opción de pre compilar en código para evitar ese tiempo de espera la primera vez que un cliente solicita la página.

Figura 7. Ejemplo básico de una página JSP



Fuente: [www.desarrolloweb.com](http://www.desarrolloweb.com). Ejemplo de página JSP.

### 1.2.5. PHP

PHP (acrónimo de "PHP: HypertextPreprocessor") es un lenguaje "open source" interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor.

Lo que distingue a PHP de la tecnología Javascript, la cual se ejecuta en la máquina cliente, es que el código PHP es ejecutado en el servidor. Si se tuviera un *script* similar al del ejemplo en el servidor, el cliente solamente recibiría el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido. El servidor web puede ser incluso configurado para que procese todos los ficheros HTML con PHP.

Lo mejor de usar PHP es que es extremadamente simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales. No se debe leer la larga lista de características de PHP, en poco tiempo se puede empezar a escribir los primeros *scripts*.

Figura 8. **Ejemplo de Programación PHP básico**

```
1 <HTML>
2 <HEAD>
3     <TITLE>Etiquetas PHP</TITLE>
4 </HEAD>
5 <BODY>
6 <CENTER>
7 <?php
8     $mensaje="Estas etiquetas son las más adecuadas para embeber código PHP";
9     echo($mensaje);
10 ?>
11 </CENTER>
12 </BODY>
13 </HTML>
```

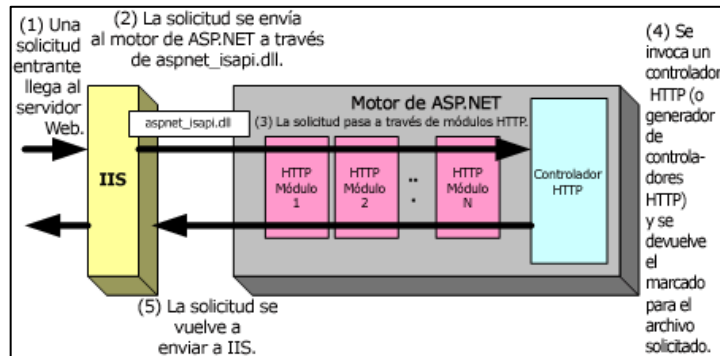
Fuente: *Programación en PHP I*. [www.adrformacion.com](http://www.adrformacion.com). Consulta: 26 de abril de 2014.

### 1.2.6. ASP.NET

ASP es la tecnología de Microsoft que permite desarrollar aplicaciones web que ejecuten en el servidor HTTP de Microsoft, el Internet Information Server (IIS). El desarrollo de aplicaciones utilizando páginas ASP consiste, básicamente, en intercalar macros o fragmentos de código dentro de los documentos HTML que sirven para crear las interfaces de usuario de las aplicaciones web. Los fragmentos de HTML proporcionan la parte estática de lo que ve el usuario mientras que los fragmentos de código generan la parte dinámica. Esto suele conducir a mezclar los detalles de la interfaz con la lógica de la aplicación, algo que, no suele ser demasiado recomendable.



Figura 9. Diagrama de funcionamiento de las páginas ASP



Fuente: *Uso de controladores y módulos http. F4. msdn.microsoft.com*. Consulta: 13 de mayo de 2014.

Igual que sucede con las demás tecnologías basadas en lenguajes interpretados, para dejar la aplicación web a disposición del usuario basta con escribir las páginas ASP y guardarlas en algún directorio al que se pueda acceder a través del Internet Information Server, sin tener que compilarlas previamente.

Figura 10. Ejemplo de página web básica con ASP

```
Client Objects & Events (No Events)
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:Label ID="Label1" runat="server" Text="Mi sitio Web para Maestros del Web"
        Font-Names="Tahoma" Font-Size="14pt"></asp:Label>
      <br />
      <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
      </asp:ContentPlaceHolder>
    </div>
  </form>
</body>
</html>
```

Mi sitio Web para Maestros del Web

Fuente: *Tutorial de ASP.NET. www.maestrosdelweb.com*. Consulta: 13 de mayo de 2014.

En la plataforma .NET se ha incluido una versión mejorada de ASP denominada ASP.NET. Entre los principales inconvenientes asociados a las versiones de ASP anteriores a ASP.NET, a las que se suele hacer referencia por "ASP Clásico", se encuentra el hecho de que ASP suele requerir escribir código muy largo. Por ejemplo, es necesario escribir código para mantener el estado de la página ASP cuando, por cualquier motivo, el usuario ha de volver a ella. Omitir dicho código provoca la frustrante sensación que todos han experimentado alguna vez cuando, tras rellenar un extenso formulario HTML, se informa de un error en uno de los campos y se ve forzado a volver a rellenar el formulario completo.

Por otro lado, el código incluido en las páginas ASP resulta, además de excesivamente extenso, poco legible y difícil de mantener al estar mezclado con el HTML de la interfaz gráfica. En ASP, el fragmento de código ha de colocarse justo en el sitio donde se requiere que su salida aparezca, lo que hace prácticamente imposible separar los detalles de la interfaz de usuario de la lógica de la aplicación. Este hecho, no solo dificulta la reutilización del código y su mantenimiento, sino que también complica el soporte de la aplicación para múltiples navegadores por ejemplo, JavaScript y JScript no son exactamente iguales.

### **1.3. Servicios web**

En este apartado se hará hincapié en todo lo relacionado a los servidores que se dedican a desplegar las aplicaciones web, que son mostradas ya sea localmente o remotamente, pero de manera específica que están comandadas por la estructura cliente-servidor, pero específicamente, en el ambiente web.

### **1.3.1. Servicios web 1.0**

Hoy en día se llaman así a los servicios que ofrecían los sitios web clásicos. En realidad, la inmensa mayoría de sitios actuales siguen siendo web 1.0 puesto que solo utilizan tecnologías clásicas. La realidad es que web 1.0 se define como lo que no es web 2.0, una definición vaga pero que en el punto siguiente se pretende aclarar mejor.

### **1.3.2. Servicios web 2.0**

Se trata de la forma de denominar a las páginas web que ofrecen servicios orientados al usuario. Las páginas web tradicionales ofrecían la misma información para todos los usuarios, ahora varían para cada usuario, le permiten un manejo más rico e incluso le hacen partícipe del contenido.

Inicialmente la web era simplemente textos e imágenes junto con los hipervínculos que permiten saltar hacia otros contenidos. Ahora el tipo de contenido es mucho más variado: animaciones, vídeo, sonido, juegos, aplicaciones ofimáticas, entre otros. Este tipo de páginas empezaron a proliferar tras el estallido de la burbuja punto com y es en el 2004 cuando la empresa O'Reilly Media le utiliza en una conferencia e incluso llega a registrarlo y lo hace popular.

Hoy en día es difícil de definir, pero sin duda hace referencia a un tipo de servicios web muy concretos. Hay tres pilares que conforman las páginas web 2.0:

- Aplicaciones ricas de internet. Es quizá el término más tecnológico relacionado con la web 2.0. también conocidas con las siglas RIA, aquellas páginas web que ofrecen servicios que las asemejan con las aplicaciones de escritorio. Permiten utilizar la web como si se estuviera utilizando una aplicación con toda su potencia en un ordenador local. Inicialmente estas aplicaciones web requerían que el usuario instalara *plugins* (como Java o Flash por ejemplo) para utilizarlas; pero hoy en día con AJAX, basta con el propio navegador.
- Arquitectura orientada al servicio (SOA). Otra de las claves, se trata de una tecnología que permite diseñar aplicaciones basándose en peticiones a un determinado servicio. De esta forma se puede crear pequeños elementos software muy reutilizables y además independientes del lenguaje con el que fueron creados.
- Web social. Es la parte más evidente y entendible de la web 2.0. El término se refiere a que el usuario posee una interacción mucho mayor en la web siendo partícipe de lo que en ella ocurre.

Las páginas web 2.0 tienen estas características:

- El contenido no solo es creado por los propietarios de las mismas, sino que es creado y modificado por los propios usuarios.
- Las páginas permiten controles más potentes de manejo por los usuarios.
- Los diferentes servicios ofrecidos por distintas páginas tienen facilidad para interaccionar (se puede colocar un mapa de Google o acceder a Facebook para compartir un detalle de la página, entre otros).

Es habitual comparar servicios ofrecidos por la web 1.0 con respecto a los de la web 2.0, por ejemplo:

Tabla I. **Tabla de comparación Web 1.0 vs Web 2.0**

Web 1,0	Web 2,0
Hotmail( Clasico)	Gmail
Doubleclick	AdWords
Geocities	Wordpress
Yahoo!	Google
Enciclopedia Británica	Wikipedia
Messenger	Twitter, Facebook
Akamai	Torrents
---	Google docs
---	Aviary

Fuente: elaboración propia.

### 1.3.3. **Aplicaciones en la nube, *Cloud Computing***

Es otro de los términos fundamentales para definir los servicios ofrecidos por las páginas actuales. Se basa en ofrecer al usuario servicios de modo que el usuario pueda acceder a ellos desde cualquier dispositivo conectado a internet, ofreciendo una altísima disponibilidad del mismo.

Se basa en la programación distribuida de aplicaciones, pero a tal nivel que las aplicaciones se distribuyen incluso por cientos de servidores de partes distintas del planeta y así poder responder a una demanda ingente de peticiones de servicio y además poseer una altísima capacidad de tolerar fallos.

De esta manera, es la tecnología que permite utilizar el software como un servicio (SaaS, Software as a Service), al igual que la web 2.0 (con la que está íntimamente relacionada), al usuario le basta un navegador o una pequeña aplicación (como una App de un dispositivo móvil) para acceder y utilizar el servicio.

La computación en la nube apareció para responder tecnológicamente a los retos de empresas como Google o Facebook que tenían que responder a gran velocidad a las peticiones de millones de usuarios. Hoy en día el concepto permite que se utilice internet como la base de trabajo, sustituyendo así al propio ordenador personal que puede ser un equipo de menor potencia al delegar en la nube el proceso de las tareas y el almacenamiento de la información.

#### **1.4. Servidores de aplicaciones web**

Usualmente se trata de un dispositivo de software que proporciona servicios de aplicación a las computadoras cliente.

##### **1.4.1. Servidor web**

Los servidores web son los encargados de recibir las peticiones referidas a páginas o elementos de la web a través del protocolo http. Normalmente, es un software alojado en un ordenador servidor.

Normalmente es el navegador el que pide al servidor web el recurso que desea el usuario, para finalmente recibir dicho recurso (si fue válida la petición) y traducirle si es necesario a su forma legible por el usuario (es decir la traducción de HTML la hace el navegador).

### **1.4.2. Servidores de aplicaciones web**

Se puede decir que es una ampliación de los anteriores. Es decir son servidores web, pero que tienen capacidad de almacenar y gestionar aplicaciones web. Bajo el entendido que una aplicación web es un servicio al que los usuarios acceden a través de la web.

Este tipo de servidores no solo sirven para atender peticiones http, sino que además son capaces de entender instrucciones de lenguajes avanzados de la web y traducirlas o bien son capaces de acceder a recursos de otros servidores.

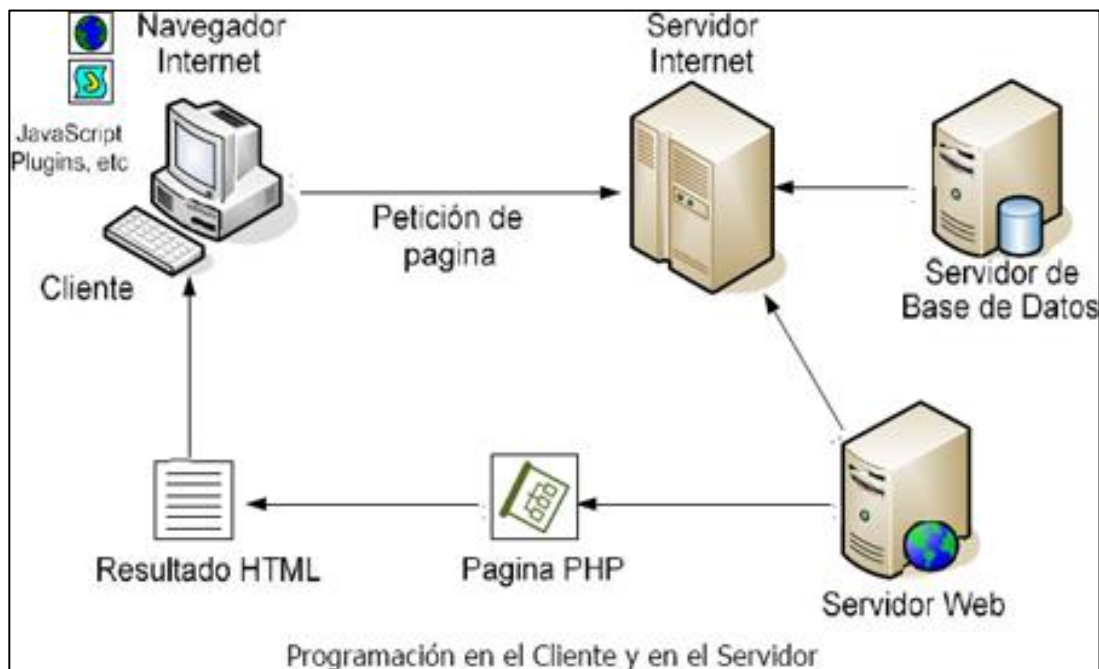
Ese proceso se hace de forma transparente al usuario, es decir el usuario pide el servicio a través, normalmente, de su navegador y el servidor de aplicaciones atiende la petición, e interpreta el código de la aplicación a fin de traducirle y mostrar al usuario el resultado de forma entendible por su navegador (es decir en formato HTML).

A la forma de trabajar de un servidor de aplicaciones, se le conoce normalmente como arquitectura de tres capas (a veces se habla de más capas). Una primera capa es la del navegador que es capaz de traducir código del lado del cliente (HTML, JavaScript, CSS, Flash, entre otros). Para ello esa capa debe disponer de todos los componentes necesarios para hacer esa labor en el ordenador del usuario.

La segunda capa la forma el servidor de aplicaciones en su labor de traducir código en el lado del servidor (JSP, PHP, Ruby onRails, ColdFusion, entre otros) y convertirlo al formato entendible por el navegador.

La tercera capa son todos los servicios a los que accede el servidor de aplicaciones para realizar la tarea encomendada a la aplicación (por ejemplo el acceso a la base de datos).

Figura 11. **Funcionamiento de un servidor de aplicaciones PHP**



Fuente: *aplicaciones web*. mailxmail.com. PHP y MySQL: Consulta: 30 de marzo de 2014.

### 1.4.3. **Servidores de aplicaciones web comerciales**

A continuación se enlistan los servidores de aplicaciones web comerciales, más comunes y simples de utilizar, para su conocimiento y elección.



### **1.4.3.1. Apache**

Este es el servidor web más popular de la actualidad. Abarca el 70 % de todos los servidores web instalados. Se trata de un software de código abierto que utiliza una licencia de tipo Apache License que es una variante de la licencia GPL de Linux. Eso significa que se puede distribuir sin problemas e incluso mejorar.

Dispone de multitud de módulos que convierten a Apache en un servidor capaz de gestionar todo tipo de aplicaciones, lo que también le convierte en el servidor de aplicaciones más popular de la actualidad; por ejemplo, dispone de módulos para:

- Implementar SSL. Protocolo de seguridad en la transferencia de información.
- Enlace con el servidor Tomcat de aplicaciones, para implementar aplicaciones Java de servidor.
- Módulo para Perl.
- Módulo para PHP
- Módulo para Python
- Otros.

### **1.4.3.2. IIS**

Abreviatura de Internet Applications Server es el servidor de aplicaciones de Microsoft que está presente en las versiones profesionales de Windows y en todas las de servidor. Viene con el propio sistema operativo y para instalarle basta con agregar componentes al mismo y elegir las funciones deseadas del servidor IIS.

El servidor incluye un servidor web (tanto http como https), servidor ftp, webDAV y smtp. Además se comporta como servidor de aplicaciones web .NET y admite extensiones para diversos tipos de aplicaciones (incluido PHP), es el segundo servidor web más popular tras Apache (20 % del mercado).

Figura 12. **Vista del sitio por defecto de IIS**



Fuente: *Logo IIS8*. [www.microsoft.com](http://www.microsoft.com). Consulta: 30 de marzo de 2014.

### 1.4.3.3. **Nginx**

A pesar de no ser una alternativa nueva, en la actualidad, se encuentra ganando mucha atención por los profesionales que requieren alternativas ligeras y rápidas al todopoderoso "Apache", y lo mejor de todo, es que es completamente libre.

Se trata de un "servidor web/proxy" completamente inverso, que tiene como principal característica ser sumamente ligero, lo que lleva a su otro gran atractivo, su velocidad, lo que permite servir "aplicaciones web" con una velocidad muy superior a la de sus competidores más directos. Es decir, es un "servidor web de alto rendimiento", ideal para realizar todo tipo de trabajos, ya sean profesionales o aficionados.

También, "nginx" va mucho más allá, ya que puede trabajar como un proxy para correo electrónico o e-mail del tipo IMAP/POP3. Su otra gran fortaleza es que es un software libre o abierto, distribuido bajo licencia BSD simplificada, lo que permite trabajar sin tener que realizar pago alguno por licencia de funcionamiento. Es un servidor web/proxy multiplataforma, listo para trabajar en Linux, Unix, Windows, MAC OS, Solaris y BSD.

#### **1.4.3.4. Apache Tomcat**

Tomcat es un contenedor web basado en el lenguaje Java que actúa como motor de *servlets* y JSPs, así es como se ha convertido en la implementación de referencia para las especificaciones de *servlets* y JSPs.

La clave de este contenedor es que tiene la capacidad de funcionar como servidor HTTP o bien puede conectarse con otro servidor HTTP como el Apache HTTP server o IIS.

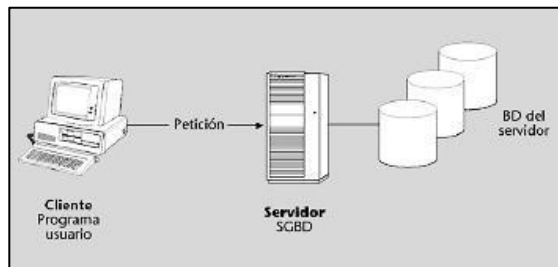
### **1.5. Bases de datos y sistemas de gestión**

Las bases de datos son el método preferido para el almacenamiento estructurado de datos. Desde las grandes aplicaciones multiusuario, hasta los teléfonos móviles y las agendas electrónicas utilizan tecnología de bases de

datos, para asegurar la integridad de los datos y facilitar la labor tanto de usuarios como de los programadores que las desarrollaron.

Desde la realización del primer modelo de datos, pasando por la administración del sistema gestor, hasta llegar al desarrollo de la aplicación, los conceptos y la tecnología asociados son muchos y muy heterogéneos. Sin embargo, es imprescindible conocer los aspectos clave de cada uno de estos temas para tener éxito en cualquier proyecto que implique trabajar con bases de datos.

Figura 13. **Relación de base de datos y su sistema de gestión**



Fuente: CAMPS PARÉ, Rafael. *Bases de datos*. p. 12.

### 1.5.1. **Concepto y origen de las BD y SGBD**

Las aplicaciones informáticas de los años sesenta acostumbraban a darse totalmente por lotes (*batch*) y estaban pensadas para una tarea muy específica relacionada con muy pocas entidades tipo, cada aplicación (una o varias cadenas de programas) utilizaba ficheros de movimientos para actualizar (creando una copia nueva) o para consultar uno o dos ficheros maestros o, excepcionalmente, más de dos. Cada programa trataba como máximo un fichero maestro, que solía estar sobre cinta magnética y en consecuencia, se

trabajaba con acceso secuencial, a medida que se integraban las aplicaciones, se tuvieron que interrelacionar sus ficheros y fue necesario eliminar la redundancia. El nuevo conjunto de ficheros se debía diseñar de modo que estuviesen interrelacionados; al mismo tiempo, las informaciones redundantes (como por ejemplo, el nombre y la dirección de los clientes o el nombre y el precio de los productos), que figuraban en los ficheros de más de una de las aplicaciones, debían estar ahora en un solo lugar.

Estos conjuntos de ficheros interrelacionados, con estructuras complejas y compartidos por varios procesos de forma simultánea (unos *on-line* y otros por lotes), recibieron al principio el nombre de Data Banks, y después, a inicios de los años setenta, el de Data Bases. Aquí se denominarán bases de datos (BD) y sistema de gestión de base de datos (SGDB).

## **1.5.2. Objetivos y servicios de los SGBD**

Los SGBD que actualmente están en el mercado pretenden satisfacer un conjunto de objetivos directamente deducibles.

### **1.5.2.1. Consultas no predefinidas y complejas**

El objetivo fundamental de los SGBD es permitir que se hagan consultas no predefinidas (*ad hoc*) y complejas.

Los usuarios podrán hacer consultas de cualquier tipo y complejidad directamente al SGBD. El SGBD tendrá que responder inmediatamente sin que estas consultas estén preestablecidas; es decir, sin que se tenga que escribir, compilar y ejecutar un programa específico para cada consulta.

El usuario debe formular la consulta con un lenguaje sencillo (que se quede, obviamente, en el nivel lógico), que el sistema debe interpretar directamente. Sin embargo, esto no significa que no se puedan escribir programas con consultas incorporadas (por ejemplo, para procesos repetitivos).

### **1.5.2.2. Flexibilidad e independencia**

La complejidad de las BD y la necesidad de ir las adaptando a la evolución del SI hacen que un objetivo básico de los SGBD sea dar flexibilidad a los cambios.

Interesa obtener la máxima independencia posible entre los datos y los procesos usuarios para que se pueda llevar a cabo todo tipo de cambios tecnológicos.

Los usuarios podrán hacer consultas de cualquier tipo y complejidad directamente al SGBD. El SGBD tendrá que responder inmediatamente sin que estas consultas estén preestablecidas; es decir, sin que se tenga que escribir, compilar y ejecutar un programa específico para cada consulta.

En los ficheros tradicionales, cada vez que se quería hacer una consulta se tenía que escribir un programa a medida, sin que se deban modificar los programas de aplicación ya escritos ni cambiar la forma de escribir las consultas (o actualizaciones) directas.

Para conseguir esta independencia, tanto los usuarios que hacen consultas (o actualizaciones) directas como los profesionales informáticos que escriben programas que las llevan incorporadas, deben poder desconocer las características físicas de la BD con que trabajan. No necesitan saber nada

sobre el soporte físico, ni estar al corriente de qué SO se utiliza, qué índices existen, la compresión o no compresión de datos, entre otros, de este modo, se pueden hacer cambios de tecnología y cambios físicos para mejorar el rendimiento sin afectar a nadie. Este tipo de independencia recibe el nombre de independencia física de los datos.

### **1.5.2.3. Integridad de los datos**

Lo que interesa en el proyecto, es que los SGBD aseguren el mantenimiento de la calidad de los datos en cualquier circunstancia. Se podría perder la corrección o la consistencia de los datos por muchas otras razones: errores de programas, errores de operación humana, avería de disco, transacciones incompletas por corte de alimentación eléctrica, entre otros.

Cuando el SGBD detecte que un programa quiere hacer una operación que va contra las reglas establecidas al definir la BD, no se lo deberá permitir, y le tendrá que devolver un estado de error. Al diseñar una BD para un SI concreto y escribir su esquema, no solo se definirán los datos, sino también las reglas de integridad que se quiere que el SGBD haga cumplir.

Aparte de las reglas de integridad que el diseñador de la BD puede definir y que el SGBD entenderá y hará cumplir, el mismo SGBD tiene reglas de integridad inherentes al modelo de datos que utiliza y que siempre se cumplirán. Son las denominadas reglas de integridad del modelo. Las reglas definibles por parte del usuario son las reglas de integridad del usuario.

El concepto de integridad de los datos va más allá de prevenir que los programas usuarios almacenen datos incorrectos. En casos de errores o

desastres, también se puede perder la integridad de los datos. El SGBD debe dar las herramientas para reconstruir o restaurar los datos estropeados.

#### **1.5.2.4. Concurrency de usuarios**

Uno de los objetivos fundamentales de SGDB es permitir que varios usuarios puedan acceder concurrentemente a la misma DB, cuando los accesos concurrentes son todos de lectura (es decir, cuando la BD solo se consulta), el problema que se produce es simplemente de rendimiento, causado por las limitaciones de los soportes de que se dispone: pocos mecanismos de acceso independientes, movimiento del brazo y del giro del disco demasiado lentos, buffers locales demasiado pequeños, entre otros.

Cuando un usuario o más están actualizando los datos, se pueden producir problemas de interferencia que tengan como consecuencia la obtención de datos erróneos y la pérdida de integridad de la BD.

Para tratar los accesos concurrentes, los SGBD utilizan el concepto de transacción de BD, concepto de especial utilidad para todo aquello que hace referencia a la integridad de los datos, se denomina transacción de base de datos o, simplemente, transacciones un conjunto de operaciones simples que se ejecutan como una unidad. Los SGBD deben conseguir que el conjunto de operaciones de una transacción nunca se ejecute parcialmente. O se ejecuten todas, o no se ejecuta ninguna.



### 1.5.2.5. Seguridad

Actualmente, en el campo de los SGBD, el término seguridad se suele utilizar para hacer referencia a los temas relativos a la confidencialidad, las autorizaciones, los derechos de acceso, entre otros.

Los SGBD permiten definir autorizaciones o derechos de acceso a diferentes niveles: al nivel global de toda la BD, al nivel entidad y al nivel atributo.

Estos mecanismos de seguridad requieren que el usuario se pueda identificar.

Se acostumbra a utilizar códigos de usuarios (y grupos de usuarios) acompañados de contraseñas (*passwords*), pero también se utilizan tarjetas magnéticas, identificación por reconocimiento de la voz, entre otros.

Es de interés almacenar la información con una codificación secreta; es decir, con técnicas de encriptación (como mínimo se deberían encriptar las contraseñas). Muchos de los SGBD actuales tienen prevista la encriptación. Prácticamente todos los SGBD del mercado dan una gran variedad de herramientas para la vigilancia y la administración de la seguridad. Hay algunos que incluso, tienen opciones (con precio separado) para los SI con unas exigencias altísimas, como por ejemplo los militares.

### 1.5.3. Modelos de BD

Una BD es una representación de la realidad (de la parte de la realidad que interesa en el SI). Dicho de otro modo, una BD se puede considerar un modelo de la realidad. El componente fundamental utilizado para modelar en un SGBD relacional son las tablas (denominadas "relaciones" en el mundo teórico). Sin embargo, en otros tipos de SGBD se utilizan otros componentes. El conjunto de componentes o herramientas conceptuales que un SGBD proporciona para modelar recibe el nombre de modelo de BD.

Los cuatro modelos de BD más utilizados en los SI son el modelo relacional, el modelo jerárquico, el modelo en red y el modelo relacional con objetos.

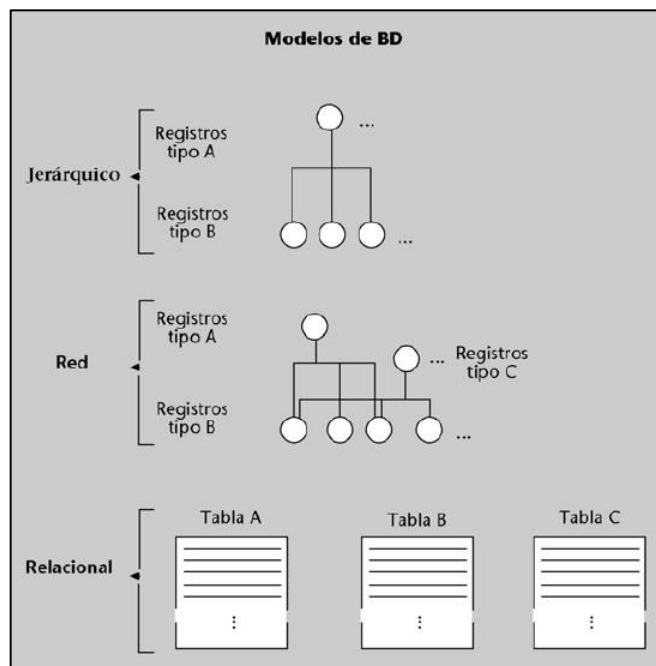
Todo modelo de BD proporciona tres tipos de herramientas:

- Estructuras de datos con las que se puede construir la BD: tablas, árboles, entre otros.
- Diferentes tipos de restricciones (o reglas) de integridad que el SGBD tendrá que hacer cumplir a los datos: dominios, claves, entre otros.
- Una serie de operaciones para trabajar con los datos. Un ejemplo de ello, en el modelo relacional, es la operación `SELECT`, que sirve para seleccionar (o leer) las filas que cumplen alguna condición. Un ejemplo de operación típica del modelo jerárquico y del modelo en red podría ser la que dice si un determinado registro tiene dependencias o no.

Así como en los modelos prerrelacionales (jerárquico y en red), las estructuras de datos constan de dos elementos básicos (los registros y las interrelaciones), en el modelo relacional constan de un solo elemento: la tabla, formada por filas y columnas. Las interrelaciones se deben modelar utilizando las tablas.

Otra diferencia importante entre los modelos prerrelacionales y el modelo relacional es que el modelo relacional se limita al nivel lógico (no hace absolutamente ninguna consideración sobre las representaciones físicas). Es decir, da una independencia física de datos total. Esto es así al hablar del modelo teórico, pero los SGBD del mercado proporcionan una independencia limitada.

Figura 14. Modelos de base de datos



Fuente: CAMPS PARÉ, Rafael. *Bases de datos*. p. 30.

#### 1.5.4. Modelos y usuarios

Para interactuar con el SGBD, el usuario, ya sea un programa de aplicación o un usuario directo, se vale de un lenguaje. Hay muchos lenguajes diferentes, según el tipo de usuarios para los que están pensados y el tipo de cosas que los usuarios deben poder expresar con ellos:

- Usuarios informáticos, expertos que querrán escribir procesos complejos y que necesitarán lenguajes complejos.
- Usuarios finales no informáticos, ocasionales (esporádicos), quienes solo harán consultas. Estos usuarios necesitarán un lenguaje muy sencillo, aunque de un rendimiento bajo en tiempo de respuesta.
- Usuarios finales no informáticos, dedicados o especializados. Son usuarios cotidianos o, incluso, dedicados exclusivamente a trabajar con la BD\*. Estos usuarios necesitarán lenguajes muy eficientes y compactos, aunque no sea fácil aprenderlos. Tal vez serán lenguajes especializados en tipos concretos de tareas.

El lenguaje SQL, que es el más utilizado en las BD relacionales, tiene verbos (instrucciones) de tres tipos diferentes:

- Verbos del tipo DML: por ejemplo, SELECT para hacer consultas, e INSERT, UPDATE y DELETE para hacer el mantenimiento de los datos.
- Verbos del tipo DDL: por ejemplo, CREATE TABLE para definir las tablas, sus columnas y las restricciones.
- Además, SQL tiene verbos de control del entorno, como por ejemplo COMMIT y ROLLBACK para delimitar transacciones.

En cuanto a los aspectos DML, se puede diferenciar dos tipos de lenguajes:

- Lenguajes muy declarativos (o implícitos), con los que se especifica qué se quiere hacer sin explicar cómo se debe hacer.
- Lenguajes más explícitos o procedimentales, que exigen conocer más cuestiones del funcionamiento del SGBD para detallar paso a paso cómo se deben realizar las operaciones (lo que se denomina "navegar" por la BD).

Como es obvio, los aspectos DDL (descripciones de los datos) son siempre declarativos por su propia naturaleza.

### **1.5.5. Tipos de datos**

En una base de datos, cada columna, variable local, expresión y parámetro tiene un tipo de datos relacionado. Un tipo de datos es un atributo que especifica el tipo de datos que el objeto puede contener: datos de enteros, datos de caracteres, datos de moneda, datos de fecha y hora, cadenas binarias, entre otros.

Las categorías en las que se dividen los datos son las siguientes:

- Cadenas de caracteres (*char, nchar, varchar, nvarchar,*)
- Numéricos exactos (*int, smallint, tinyint, long, ...*)
- Numéricos aproximados (*float, double, single, ...*)
- Fecha y hora (*date, datetime, time, ...*)
- Otros tipos (datos de imagen, textos largos, Unicode, binarios, ...)

## 1.6. *Framework web*

El concepto *framework* se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones web. Se puede encontrar *frameworks* para el desarrollo de aplicaciones médicas, de visión por computadora, para el desarrollo de juegos, y para cualquier ámbito que pueda ocurrirse.

En general, el término *framework*, se refiere a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un *framework* se puede considerar como una aplicación genérica incompleta y configurable a la que se puede añadirle las últimas piezas para construir una aplicación concreta.

Los objetivos principales que persigue un *framework* son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Un *framework web*, por tanto, se puede definir como un conjunto de componentes (por ejemplo, clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web.

## 2. TECNOLOGÍAS APLICABLES EN EL PROYECTO

Debido a la vasta diversidad de tecnologías, plataformas y dispositivos, desde los cuales hoy en día es posible acceder a los sitios web, es necesario desarrollar un entorno web totalmente adaptable, como flexible, para que la experiencia de los usuarios sea completamente agradable, usable e intuitivo.

Para mantenerse a la vanguardia de las plataformas, han sido desarrolladas nuevas tecnologías para ambientes web, desde móviles hasta de sistemas operativos de código abierto, entre otros.

### 2.1. Diseño web adaptativo

También llamado *responsive Web design (RWD)*, es un conjunto de técnicas que defiende el diseñador web Ethan Marcotte. Sitios diseñados con este enfoque adaptan sus diseños de acuerdo al entorno del navegador del usuario, en gran parte por hacer algunas cosas ingeniosas con CSS.

Dependiendo del valor actual de ciertos navegadores, como tamaño de ventana, la orientación del dispositivo, o relación aspecto, se puede aplicar diferentes CSS en diferentes circunstancias. Al repensar la forma de hacer los diseños de página, se puede hacer antes un contenedor único para todas las columnas y diseños de a cuadrícula que fluyan con más naturalidad, a través de la gran variedad de tamaños de ventana del navegador o dispositivos

El RWD se fija también mucho en el usuario y la interactividad de este con la página. Trata de aligerar las cargas lo más posible mediante la eliminación de

cosas que antes se hacían con imágenes como por ejemplo, los menús, los cuales se usaban con puro código html y css. Y el trato de imágenes explícitas se realiza de una forma que no pesen demasiado, es decir, sugiere guardarlas para web, ya sea desde Photoshop y otro programa similar.

Figura 15. **Ejemplo de un sitio web adaptativo**



Fuente: *Responsive VS Adaptive*. [www.inflexioninteractive.com](http://www.inflexioninteractive.com). Consulta: 10 de marzo de 2014.

Existen 3 técnicas básicas para la construcción de un sitio web basado en RWD:

### 2.1.1. **Template de rejilla fluida**

Para esta técnica es necesario diseñar la estructura HTML, luego la codificación CSS, la cual hará el trabajo necesario de sitio adaptativo. Las principales propiedades de RWD son el min-width y max-width, estas dos son la base para generar estructuras fluidas.



El otro punto importante es prescindir de las medidas absolutas y usar las relativas, es decir, evitar los pixeles para establecer altos y anchos, y en su lugar utilizar porcentajes, esto se puede observar en la figura 16.

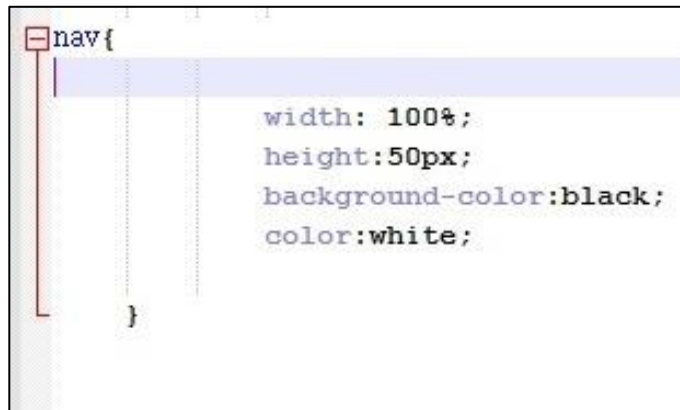
Figura 16. **Ejemplo de hoja de estilo con pixeles limitados**

```
body{  
  
    margin: 0px auto;  
    max-width: 1200px;  
    min-width: 320px;  
  
}
```

Fuente: elaboración propia.

Se necesita que el Nav mantenga un ancho del 100 % en todo momento, este se encuentra en el *body*, el cual tiene los límites de 320 a 1 200 pixeles, con esto siempre se tendrá el Nav al 100 % de lo que mida el *body*. Si la ventana se hace grande o pequeña, el nav hará lo mismo. También se puede fijar la altura a 50 px, un fondo negro y las letras blancas:

Figura 17. **Ejemplo de hoja de estilo con altura fija**

A screenshot of a CSS code editor showing a rule for a navigation bar. The code is: 

```
nav{  
width: 100%;  
height: 50px;  
background-color: black;  
color: white;  
}
```

 The code is displayed in a light blue background with a white border. A vertical red line is on the left side of the code block, and a horizontal blue bar highlights the first line of the code.

Fuente: elaboración propia.

De esta forma se puede hacer con cada uno de los estilos de la hoja para lograr que el sitio sea totalmente adaptable a cada uno de los navegadores, dispositivos, entre otras.

### 2.1.2. **Imágenes fluidas**

Las imágenes juegan un papel importante a la hora de hacer sitios web adaptativos. Una imagen no debe exceder a su tamaño original debido a que se pixelaría y se vería con cierto grado de distorsión. Caso contrario, las imágenes si pueden ser más pequeñas de su tamaño original. Debe verse siempre de una forma coherente y con sentido de posición.

Lo que se debe hacer con las imágenes es indicarles que ocupen el 100 % de su contenedor, tomando en cuenta que no excedan el límite de su tamaño. Si por ejemplo el estilo está a la resolución 1 200 px, tomará el 100 % del objeto que en ese momento es de poco menos de 300 px, ya que la clase tiene un porcentaje de *padding* en cada lado. A resoluciones más chicas, el 100 %

será más pequeño, por lo tanto el estilo irá haciéndose chico y como no importa que la imagen se haga más pequeña, no afectará su visibilidad.

Figura 18. **Parte del CSS que fija imágenes al 100 % del contenedor**

```
img {  
    width: 100%;  
}
```

Fuente: elaboración propia.

Ahora bien, esta página se verá como se muestra en los navegadores de computadoras de escritorio.

Pero si se observa en un dispositivo móvil no se verá con los cambios, siempre se verá la máxima resolución y no tomará en cuenta los *media queries*.

Para que la página funcione correctamente, hay que agregar una meta en el *head*.

Figura 19. **Código en *head* para funciones las imágenes fluidas**

```
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width,initial-scale=1.0">  
  <title>Prueba de Web Adaptativo</title>  
  <link rel="stylesheet" href="estilo.css">  
</head>
```

Fuente: elaboración propia.

### 2.1.3. Media types y media queries

Los *media types* permiten indicar una serie de estilos que se aplican según el tipo del medio, en algunas páginas, al ver el código fuente, se encuentra algo como se indica en la figura 20.

Figura 20. **Head con media types**

```
<link rel="stylesheet" type="text/css" media="all" href="style.css">  
<link rel="stylesheet" type="text/css" media="print" href="print.css">
```

Fuente: elaboración propia.

En la figura 20, se incluye una hoja de estilos llamada style.css siempre (media="all"), pero existe una hoja de estilos llamada print.css que solo se ejecutará cuando se esté imprimiendo el documento (media="print").

Los *media types* más comunes son:

- All: para todo (valor por defecto)
- Print: en la vista previa de impresión y a la hora de imprimir
- Screen: para las pantallas de ordenador
- Tv: para televisores

Hablando de *media queries* los cuales son propios de CSS3, utilizando un *media type* con una serie de expresiones relacionadas con las características del dispositivo desde el cual se accede a la web. Las *media queries* dan como

resultado verdadero o falso, Si el resultado es verdadero, se leen las reglas CSS de su interior, si es falso no, como lo mostrado en la figura 21.

Figura 21. **Ejemplo de CSS con media quertie**

```
body {  
    background: green ;  
}  
  
@media (max-width: 600px) {  
    body {  
        background: red ;  
    }  
}
```

Fuente: elaboración propia.

Esto va a hacer que el fondo de la web sea verde, pero cuando la pantalla sea menor a 600 px el fondo será rojo.

Al incluir la hoja de estilos en el HTML de la siguiente forma:

Figura 22. **Inclusión de la hoja de estilos en el HTML**

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <link rel="stylesheet" type="text/css" media="all" href="style.css">  
  </head>  
  <body>  
  </body>  
</html>
```

Fuente: elaboración propia.

En resumen, las *media queries* se componen de un *media type* y una o más expresiones, implicando características del medio, la cual se resuelve como verdadera o falsa. El resultado de la consulta es verdadero si el tipo del medio especificado en el *media query* concuerda con el tipo de dispositivo que está siendo mostrado y todas las expresiones en el *media query* son verdaderas.

## **2.2. Arquitectura MVC (Modelo-Vista-Controlador)**

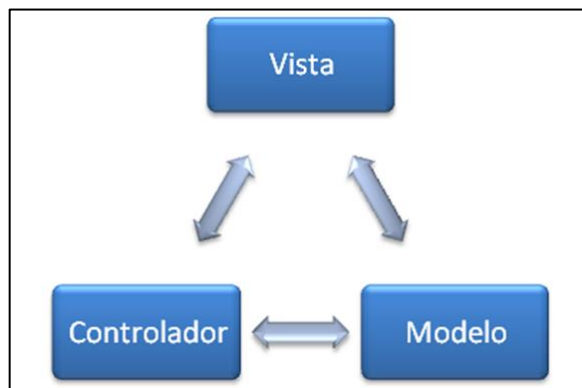
Las tecnologías vistas se enmarcan dentro del desarrollo de aplicaciones web orientado a objetos y por lo tanto, deberían responder al patrón de diseño Modelo-Vista-Controlador (MVC). Este patrón propone dividir la aplicación en tres partes diferenciadas:

- El modelo o la lógica de negocio de la aplicación. De entre todos los elementos tecnológicos que han ido apareciendo alrededor del modelo web básico, el papel del modelo queda reservado a los "servidores de aplicación", gestionados mediante un marco de trabajo middleware y basando los desarrollo en el modelo de "componentes software distribuidos" al que estos marcos proporcionan servicios.
- La vista. Es responsable de gestionar el nivel de presentación, proporcionando una interfaz de usuario altamente desacoplada, tanto de la lógica de negocio como, sobre todo, del cliente. En la práctica, resulta muy aconsejable poder aislar al máximo este apartado del resto de la aplicación, de forma que se puede dedicar personal especializado en aspectos como el diseño gráfico y de interfaces de usuario más que en el de aplicaciones.
- El controlador. Es el componente que se encarga de manejar la interacción entre la vista y el modelo, y tiene definido el flujo de la

aplicación. Actúa como catalizador necesario para desarrollar las dos partes anteriores. De su elección depende principalmente que se puedan diferenciar nítidamente los distintos roles que deben establecerse para el desarrollo de aplicaciones web.

La lógica de negocio o "modelo" deberá estar codificada en forma de componentes software, que la encapsulen y sin embargo, la manera en la que se reparten los otros dos papeles es algo abierto a discusión.

Figura 23. **Diagrama de flujo MVC**



Fuente: mind42.com. Modelo Vista Controlador MVC.

En aplicaciones a gran escala, cambia con mayor frecuencia la lógica, por ello un diseño incorrecto en el cual se fusionen la interfaz de usuario con el mundo conllevaría a que fuera necesario modificar varios elementos del negocio en el caso en que se presentase una necesidad de transformación, conllevando a mayores gastos y riesgos y así se requiere que se desacople la vista, logrando con ello mayor reusabilidad.

### 2.2.1. Patrón MVC en PHP

Para que el concepto de Modelo-Vista-Controlador quede más claro, se irá analizando con ejemplos concretos en php.

Respecto al modelo, no se tiene demasiada novedad. El modelo, no es más que el conjunto de clases que ya se conocen, incluyendo entre ellas, la capa de abstracción de la base de datos.

En esta etapa, se pueden ampliar las referencias de PHP, incorporando el concepto de *interfaces*, las cuales son objetos que permiten crear código con el cual especificar que métodos deben ser implementados por una clase, sin tener que definir que estos métodos son manipulados.

Las interfaces son definidas utilizando la palabra clave *interface* y se implementa utilizando la palabra clave *implements*.

Figura 24. **Generación e implementación de la interface**

```
/* Generamos la interface:
interface Postre {
public function set_ingredientes();
}
/*Implementamos la interface:
class Bizcochuelo implements Postre {
var $ingredientes = array();
public function set_ingredientes() {
$this->ingredientes = array('harina'=>'2 tazas', 'leche'=>'1 taza');
}
}
```

Fuente: elaboración propia.



No obstante, es necesario esclarecer que una interface no puede tener el mismo nombre que una clase ya que PHP las interpretaría como una clase más. Crear una interface con el mismo nombre que una clase, sería interpretado por PHP como una “re-declaración de clase”), así también diferentes interfaces no pueden tener nombres de métodos idénticos si serán implementadas por la misma clase.

La capa de la vista también puede aprovechar la separación de código, las páginas web suelen contener elementos que se muestran de forma idéntica a lo largo de toda la aplicación: cabecera de la página, el *layout* genérico, el pie de página y la navegación global. Normalmente solo cambia el interior de la página. Por este motivo, la vista se separa en un *layout* y en una plantilla.

Normalmente, el *layout* es global en toda la aplicación o al menos en un grupo de páginas. El *layout* es global en todas las aplicaciones o al menos en un grupo de páginas. La plantilla solo se encarga de visualizar las variables definidas en el controlador. Para que estos componentes interaccionen entre sí correctamente, es necesario añadir cierto código. Siguiendo estos principios, la parte de la vista del *script* inicial se puede separar en 3 partes: la plantillas de vista, la lógica de vista, *layout* de la vista.

La capa del controlador en la vida real suele tener mucho trabajo. Una parte importante de su trabajo es común a todos los controladores de la aplicación. Entre las tareas comunes se encuentran el manejo de las peticiones del usuario, el manejo de la seguridad, cargar la configuración de la aplicación y otras tareas similares. Por ese motivo, el controlador normalmente se divide en un controlador frontal, que es único para cada aplicación, y las acciones que incluyen el código específico del controlador de cada página.

Una de las ventajas de utilizar un controlador frontal es que ofrece un punto de entrada único para toda la aplicación. Así, en caso de que sea necesario impedir el acceso a la aplicación, solamente es necesario editar el *script* correspondiente al controlador frontal. Si la aplicación no dispone de controlador frontal. Si la aplicación no dispone de controlador frontal, se debe modificar cada uno de los controladores.

### **2.2.2. Modelo MVC en ASP.NET**

Las aplicaciones web utilizan una variante del modelo MVC conocida como MVC pasivo, porque las vistas solo se actualizan cuando se realiza alguna acción a través del controlador. Esto es, aunque el estado del modelo se vea modificado, la vista no se actualizará automáticamente. El navegador del usuario muestra la vista y responde a las acciones del usuario pero no detecta los cambios que se puedan producir en el servidor. Dichos cambios pueden producirse cuando varios usuarios utilizan concurrentemente una aplicación o existen otros actores que pueden modificar los datos con los que trabaja esta aplicación. Las peculiaridades de las interfaces web hacen que, cuando es necesario reflejar en el navegador del usuario los cambios que se producen en el servidor, y se tenga que recurrir a estrategias de refrescamiento de los mismos.

Retomando en qué consistían las páginas ASP.NET, se encontró que existen dos estilos para la confección de páginas ASP.NET:

- Se puede incluir todo en un único `.aspx` o separar los controles de la interfaz del código de la aplicación. Si se emplea un único fichero `.aspx`, dicho fichero implementa los otros roles diferenciados por el modelo MVC: modelo, vista y controlador aparecen todos revueltos. Entre otros

inconvenientes, esto ocasiona que un error introducido en el código al modificar el aspecto visual de una página no se detectará hasta que la página se ejecute de nuevo.

- Si se emplean ficheros diferentes, se separan físicamente la interfaz y la lógica de la aplicación. El fichero .aspx contendrá la presentación de la página (la vista según el modelo MVC) mientras que el fichero.aspx.cs combinará el modelo con el controlador. El código común a distintas páginas se podrá reutilizar con comodidad y un cambio que afecte a la funcionalidad común, y no requerirá que haya que ir modificando cada una de las páginas. Además, se podrá modificar el aspecto visual de la página sin temor a introducir errores en el código de la aplicación y se podrá utilizar todas las facilidades que ofrezca el entorno de desarrollo para desarrollar ese código (al contener el fichero .aspx.cs una clase convencional escrita en C#).

Cuando se crea una aplicación MVC con ASP.NET debe considerarse cuidadosamente, si desea implementar una aplicación web mediante el marco de ASP.NET MVC o el modelo de formularios Web Forms de ASP.NET. El marco de MVC no reemplaza el modelo de formularios Web Forms; puede usar cualquiera de los dos marcos para las aplicaciones basadas en formularios Web Forms, estas seguirán funcionando exactamente igual que siempre.

Antes de decidir usar el marco de MVC o el modelo de formularios Web Forms para un sitio web concreto, sopesa las ventajas de cada método.

El marco de ASP.NET MVC ofrece las ventajas siguientes:

- Facilita la administración de la complejidad, al dividir una aplicación en el modelo, la vista y el controlador.

- No usa el estado de vista ni formularios basados en servidor. Esto hace que el marco MVC sea el idóneo para los desarrolladores que deseen un control completo sobre el comportamiento de una aplicación.
- Usa un modelo de controlador frontal que procesa las solicitudes de la aplicación web a través de un controlador único. Esto permite diseñar una aplicación que admite una infraestructura de enrutamiento avanzada.
- Proporciona una mayor compatibilidad con el desarrollo basado en pruebas (TDD).
- Funciona bien para las aplicaciones web en las que trabajan equipos grandes de desarrolladores y para los diseñadores web que necesitan un alto grado de control sobre el comportamiento de la aplicación.

El marco de ASP.NET MVC ofrece las características siguientes:

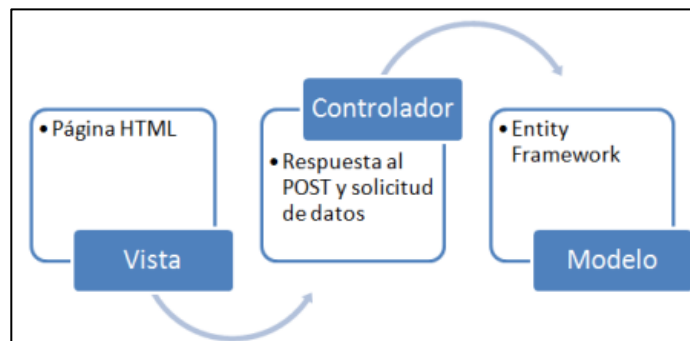
- Separación de tareas de aplicación (lógica de entrada, lógica de negocios y lógica de la interfaz de usuario), facilidad para pruebas y desarrollo basado en pruebas (TDD). Todos los contratos principales del marco de MVC se basan en interfaz y se pueden probar mediante objetos ficticios, que son objetos simulados que imitan el comportamiento de objetos reales en la aplicación. Puede hacer una prueba unitaria de la aplicación sin tener que ejecutar los controladores en un proceso de ASP.NET, lo cual hace que las pruebas unitarias sean rápidas y flexibles.
- Un marco extensible y conectable. Los componentes del marco de ASP.NET MVC están diseñados para que se puedan reemplazar o personalizar con facilidad. Puede conectar su propio motor de vista, directiva de enrutamiento de URL, serialización de parámetros de

método y acción, y otros componentes. El marco de ASP.NET MVC también admite el uso de los modelos de contenedor Inyección de dependencia (DI) e Inversión de control (IOC). DI permite insertar objetos en una clase, en lugar de depender de que la clase cree el propio objeto. IOC especifica que si un objeto requiere otro objeto, el primer objeto debe obtener el segundo objeto de un origen externo como un archivo de configuración.

- Amplia compatibilidad para el enrutamiento de ASP.NET, un eficaz componente de asignación de direcciones URL que le permite compilar aplicaciones que tienen direcciones URL comprensibles y que admiten búsquedas. Las direcciones URL no tienen que incluir las extensiones de los nombres de archivo y están diseñadas para admitir patrones de nombres de direcciones URL que funcionan bien para la optimización del motor de búsqueda (SEO) y el direccionamiento de transferencia de estado representacional (REST, Representational State Transfer).
- Compatibilidad para usar el marcado en archivos de marcado de páginas de ASP.NET existentes (archivos .aspx), de controles de usuario (archivos .ascx) y de páginas maestras (archivos .master) como plantillas de vista. Puede usar las características de ASP.NET existentes con el marco de ASP.NET MVC, tales como páginas maestras anidadas, expresiones en línea (<%= %>), controles de servidor declarativos, plantillas, enlace de datos, localización, entre otros.
- Compatibilidad con las características de ASP.NET existentes. ASP.NET MVC le permite usar características, tales como la autenticación de formularios y la autenticación de Windows, la autorización para URL, la pertenencia y los roles, el *caching* de resultados y datos, la

administración de estados de sesión y perfil, el seguimiento de estado, el sistema de configuración y la arquitectura de proveedor.

Figura 25. **Modelo Vista Controlador Entity Framework**



Fuente: *Arquitectura MVC*. [www.compilando.es](http://www.compilando.es). Consulta: 10 de marzo de 2014.

### 2.3. **ADO.NET Entity Framework**

Entity Framework es un marco de trabajo diseñado para la plataforma .NET que permite superponer varias capas de abstracción sobre un almacén relacional, con el fin de hacer posible un desarrollo de manera conceptual y de reducir a una mínima expresión el desajuste de impedancias, causado por las diferencias entre los modelos de programación relacional y orientado a objetos.

Es importante destacar que Entity Framework es una parte integral de ADO.NET a partir de .NET Framework 3.5 service Pack 1. Más exactamente Entity Framework incluye un nuevo proveedor de ADO.NET, llamado Entity Client, que habilita el acceso a los modelos conceptuales.

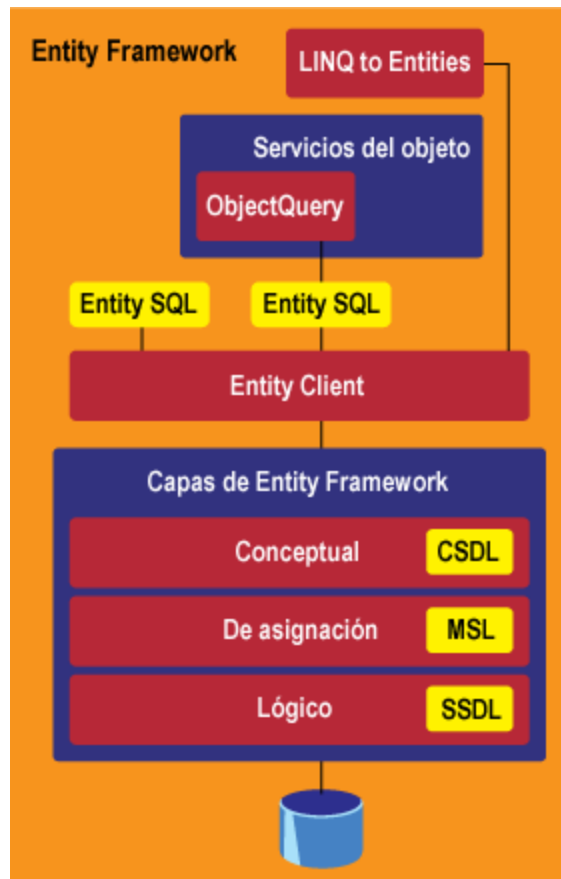
Como todo marco de trabajo, Entity Framework incluye dos componentes fundamentales:

- Recursos para el entorno de desarrollo y en particular un asistente para el diseño visual de modelos de entidades dentro de Visual Studio, así como la generación de código a partir de los mismos.
- Una biblioteca, en la cual los tipos que componen ADO.NET Entity Framework se implementan físicamente en el ensamblado System.Data.Entity.

### **2.3.1. Arquitectura y componentes**

ADO.NET Entity Framework se apoya en seis elementos fundamentales construidos encima de todas las bases de ADO.NET, tal y como se puede ver en la figura 26.

Figura 26. Componentes principales de ADO.NET Entity Framework



Fuente: *ADO.NET*. Entity. Framework.blogs.msdn.com. Consulta: 12 de marzo de 2014.

Esta es la estructura de capas de extracción en la cual el *entity framework* está basado, de arriba hacia abajo, hasta llegar a la base de datos puramente.

### 2.3.2. Proveedores específicos

Una de las características más atractivas de ADO.NET EF es su “gnosticismo” con relación a la base de datos contra la que se trabaja. En verdad, EF es una implementación de un Data Mapper entre las entidades



definidas en un modelo conceptual y el esquema físico de su base de datos subyacente. Por defecto, en la actual versión de esta tecnología se tiene un proveedor específico para Sql Server, en todas sus versiones 200, 2005,2008 y 2008R2, y también para SLQ Compact Edition.

La posibilidad de crear aplicaciones multibase de datos es cada vez más un requisito indispensable para fabricantes de software, también conocidos por sus siglas ISV (Independent Software Vendors), en departamentos de informática de grandes corporaciones donde la selección del motor de la base de datos podría variar entre cambios organizativos y de directores. Lógicamente, el hecho de que las aplicaciones puedan soportar múltiples motores relacionales es un valor añadido para aquellas empresas que quieren ubicar su producto en el mercado.

### **2.3.3. Modelo de entidades**

El primero de los elementos fundamentales en la arquitectura de ADO.NET Entity Framework es el modelo de entidades, normalmente conocido como EDM. Este diseñador permite definir los conjuntos de entidades y relaciones entre las mismas de los modelos conceptuales, así como especificar la forma en la cual los tipos se mapearán a la estructura de la fuente de almacenamiento relaciona subyacente.

### **2.4. Motor de vistas “Razor”**

Desde su versión ASP.NET MVC ha tenido el concepto de motor de vistas (View Engine). En ASP.NET MVC las vistas realizan tareas solo de presentación. No contienen ningún tipo de lógica de negocio y no acceden a datos. Básicamente se limitan a mostrar datos y a solicitar datos nuevos al

usuario. Los Web Controls no existen en ASP.NET MVC. No hay *drag and drop*, no se configuran propiedades. Las vistas son básicamente HTML. Y lo que no es HTML, son pequeñas porciones de código de servidor destinadas a terminar generando HTML para mostrar información.

## 2.5. Jquery

El jQuery es una librería de JavaScript y está diseñado para cambiar la manera en que se escribe código JavaScript.

Las aplicaciones en internet son cada vez más complejas, ya que incorporan efectos visuales, *drag and drop*, autocompletar, animaciones, entre otros. El desarrollar todos estos conceptos desde cero puede resultar complicado sobre todo si se tiene que presentar la solución con muy poco tiempo, en este tipo de situaciones el empleo de librerías como el jQuery facilitan el desarrollo de la aplicación. Otra ventaja paralela es la despreocupación por saber cuando codificar en la compatibilidad de navegadores, ya que la librería resolverá esto.

jQuery ha logrado algo aún más impresionante, ha animado a un gran número de diseñadores web a comenzar a escribir su propio código.

Es un archivo relativamente pequeño (84 kb sin comprimir) que contiene docenas de funciones (métodos) de JavaScript diseñadas para simplificar el trabajo con JavaScript. Simplemente se liga la página web al archivo y se usan los métodos de jQuery para agregar interactividad al sitio web.

JavaScript es un lenguaje muy potente, pero hay dos obstáculos importantes para su uso:

- Fue desarrollado originalmente por Netscape, el archirrival de Internet Explorer durante la guerra de los navegadores en los años 90. Aunque Internet Explorer adoptó rápidamente el lenguaje (su versión oficial se llama JScript), hay diferencias fundamentales en ciertas áreas, como el manejo de eventos. Hacer *scripts* que funcionen en todos los navegadores generalmente implica rescribir mucho código.
- La interactividad consiste en agregar y quitar elementos de una página o afectar los estilos que tengan. Los métodos oficiales para hacer esto, usando el Documento de Modelo de Objetos del consorcio World Wide Web (W3C DOM), son bastante engorrosos.

Las librerías de JavaScript destinados a resolver estos dos problemas comenzaron a surgir alrededor de 2005. Los primeros en ganar un uso generalizado fueron Prototype y Script.aculo.us. Otras librerías conocidas son Dojo Toolkit, Moo Tools, Yahoo! UI Library y el Adobe Spry framework. Todos tienen los objetivos básicos de jQuery: proporcionar a todos los navegadores de herramientas para añadir interactividad a sus páginas. Así que, ¿por qué elegir jQuery sobre los demás?

- jQuery funciona en todos los navegadores actuales incluyendo Internet Explorer 6+, Firefox 2+, Safari 3+, Chrome, y Opera 9+.
- Es de código libre licenciado a la vez con MIT License y GNU General Public License
- Se estima que tres de cada cuatro sitios web con JavaScript han adoptado jQuery.
- Empresas líderes usan jQuery como Amazon.com, Bank of America, BBC y Twitter.



### **3. EL SISTEMA ANTIGUO Y EL NUEVO**

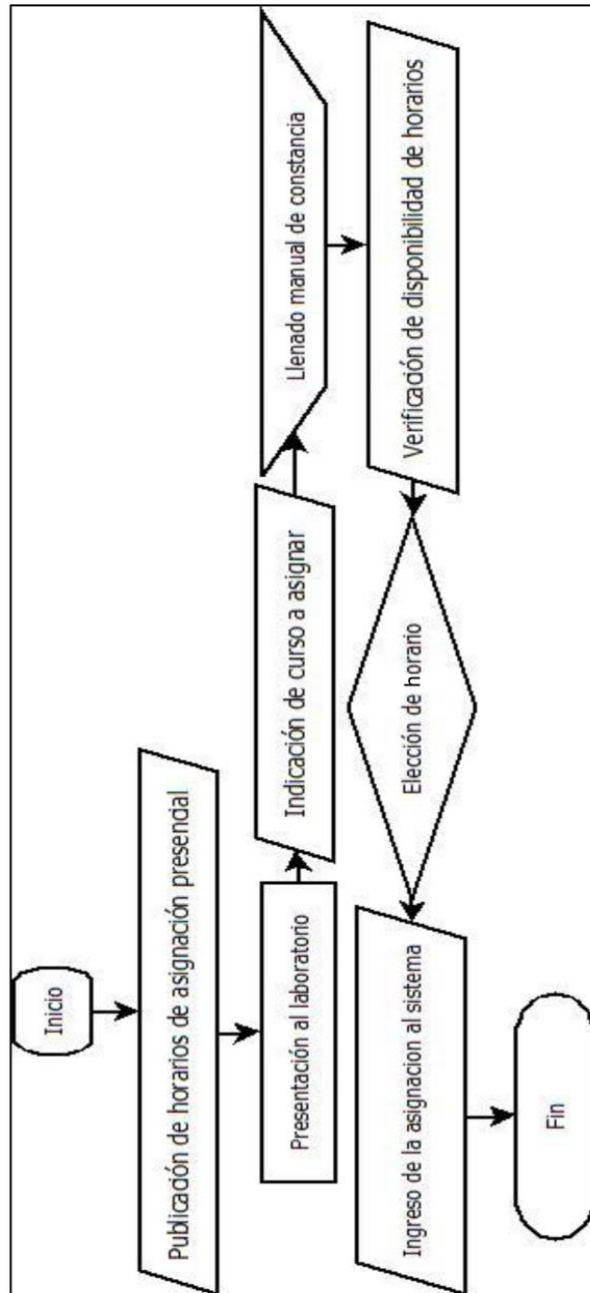
#### **3.1. Antecedentes**

Todos los estudiantes de la Escuela de Ingeniería Mecánica Eléctrica, de la Universidad de San Carlos de Guatemala, a partir del 6to semestre han tenido la necesidad de ser parte del proceso de asignación de laboratorios cada semestre, ya que es parte esencial de la formación profesional, y apegado al flujo y sistema existentes, se realiza el proceso de asignación y revalidación de laboratorios como correspondía.

A continuación se detallará cada una de las partes del proceso de asignación con el sistema antiguo, el flujo del mismo está representado por la figura 27.

En este proceso se depende de varios actores, los cuales deben coincidir, en el momento de la asignación, es decir, todos los integrantes del curso, y el auxiliar asignado. Sumado a esto, el proceso se realiza de forma manual, en varias de sus fases, aparte del hecho de que se realiza de forma presencial, en horarios no seleccionados por el alumno, lo cual llega a volverse complejo, ya que algunas veces el horario de asignación es en horario laboral, y un buen porcentaje de los estudiantes tiene compromisos laborales.

Figura 27. Diagrama del proceso de asignación de laboratorios



Fuente: elaboración propia.

### **3.1.1. Descripción del proceso**

El proceso inicia cuando son publicadas las fechas y horas en las cuales pueden realizar la asignación presencial en el Laboratorio de Electrónica.

Luego de publicados los días y horas de asignación, los alumnos en grupos con un máximo de 3 integrantes, deben presentarse al laboratorio, en los horarios publicados, y esperar su turno. Una vez presente el grupo completo frente al auxiliar, este entregará una boleta en papel, la cual debe ser llenada por los alumnos con sus nombres, números de carné, y nombre del curso a asignarse. Habiendo llenado esta información, el auxiliar procede a indicar los días y horas libres para poder asignarse el curso, y así los alumnos en consenso, eligen el horario que más les convenga, estos horarios están sujetos a la disponibilidad de máquinas, tarjetas, y equipo en general.

Habiendo elegido el horario, este es indicado al auxiliar, para que sea ingresado en el sistema. Seguido de esto, el auxiliar firma y sella la constancia de asignación, y el alumno se retira.

### **3.1.2. El sistema de asignaciones**

El sistema que se utiliza en la actualidad es un sistema de escritorio, el cual necesita de una instalación, y una base de datos local, como se explicó en los primeros capítulos un sistema consta básicamente de una colección de datos (base de datos) y una aplicación (web o de escritorio), en este caso en específico se indicará la forma en que fue construida la aplicación, y su funcionalidad.

### 3.1.2.1. El equipo

El sistema que es utilizado para asignar los cursos no era manejado por un servidor, sino más bien, instalado en un equipo *stand-alone*, en el cual la base de datos era instalada, configurada, alimentada y consultada localmente por la aplicación, por lo cual, no era necesario tener un servidor que manejara la aplicación, ni ser veía la necesidad de que tuviera disponibilidad 24 horas.

Todo esto permitía que el sistema fuera utilizado en cualquier equipo las siguientes especificaciones mínimas, ver tabla II.

Tabla II. **Especificaciones mínimas del equipo que aloja el sistema**

<b>Característica</b>	<b>Valor</b>
Sistema Operativo	XP o Superior
Memoria RAM	1 GB o Superior
Procesador	Pentium IV o superior
Disco Duro	200 GB o superior

Fuente: elaboración propia.

El paquete de instalación se encarga de ubicar cada uno de los componentes necesarios para que la aplicación pueda funcionar en cualquier equipo, así como instalar la base de datos y su motor.

### 3.1.2.2. La base de datos de asignaciones

El desarrollador del sistema que se utiliza actualmente eligió en su momento como motor de bases de datos Firebird, en su versión 1.5.3, el cual es un motor de bases de datos relacionales el cual ofrece todos los estándares



ANSI SQL , el cual correó en Linux, Windows y una variedad de plataformas UNIX.

Las tablas de esta base de datos, contiene la información de los alumnos, de los cursos, de las asignaciones, y de una forma relacional, manejaban las transacciones de asignación para la aplicación.

A continuación se detallará las tablas que estaban contenidas en este esquema:

- T0\_CLAVES: contiene la información de seguridad para acceder a la aplicación, por medio de esta y de las claves de acceso almacenadas aquí, la aplicación manejaba el *roll* con el que se accede.
- T0\_ESTUDIANTES: maneja toda la información personal del estudiante, incluyendo la ruta en la que está almacenada su foto, para ser utilizada por la aplicación.
- T3\_RUBROS: maneja cada una de las subcategorías en las que se divide el curso, por ejemplo, tarjetas, asistencia, nota mínima de aprobación, entre otros.
- T0\_CURSOS: en ella están almacenados cada uno de los cursos disponibles, así como la cantidad de tarjetas asignadas al mismo, los períodos por semana, entre otros.
- T1\_GRUPOS: únicamente contiene el número de grupo y el prefijo asignado según el curso, así como el código del mismo.
- T0\_HORARIOS: contiene la información de los intervalos de tiempo en los cuales pueden ser asignados los cursos, especificando si es fin de semana, el tiempo del intervalo, entre otros.
- T0\_DIAS: es un catálogo el cual se utiliza para asignar un código a cada día de la semana a nivel de base de datos.

- T2\_ASIGNACIONES: relaciona un correlativo de asignación, al alumno por su carnet, y la nota final del laboratorio.
- T1\_REV\_AUTORIZADAS: define cada una de las revalidaciones que han sido validadas a solicitud del estudiante, incluyendo su vigencia, la fecha en la que el curso fue aprobado, el auxiliar del laboratorio y la razón de la revalidación.
- T0\_PREFIJOS: aquí se definen los prefijos de cada uno de los grupos y los cursos que serán utilizados en el momento de la asignación.
- T3\_NOTAS: relaciona a cada alumno con la nota asignada en un rubro específico del curso.
- T1\_RESUMENOTAS: contiene la sumatoria de todos los rubros por alumno por curso.
- T2\_HORARIOSXGRUPO: en este punto, se relacionan los grupos con un horario en específico, lo cual también reserva un lugar para cada grupo.

La relación entre cada una de las tablas puede ser observada en el diagrama del anexo 2, en el cual se ve detalladamente cada uno de los campos el tipo de dato del mismo.

### **3.1.2.3. La aplicación de asignaciones**

Una vez teniendo el equipo que va a ser utilizado, la aplicación puede ser instalada en el mismo para ser utilizada (puede consultarse el manual de instalación de la aplicación antigua en el apéndice 1). La instalación desempaqueta los archivos por defecto en la carpeta de programas con el nombre "Asignaciones", la base de datos queda ubicada en una subcarpeta llamada *Database* con la Carpeta de "Asignaciones" como nivel superior.

Una vez instalada la aplicación se puede acceder a ella ubicando el acceso directo en el escritorio que muestra un circuito integrado como ícono y con nombre "Asignaciones", utilizando el usuario administrador el cual es único y la clave debe ser compartida por todos los usuarios que deseen ingresar a la aplicación con privilegios, la ventana de acceso se puede observar en la figura 28.

Figura 28. **Ventana de acceso al sistema antiguo**



Fuente: elaboración propia.

Una vez dentro de la aplicación, se observa el menú principal de la misma, en la cual se cuenta con 3 submenús claves, los cuales son los siguientes.

### **3.1.2.3.1. Submenú de configuración**

Utilizado para configurar la información concerniente a la aplicación y a personalización de la misma, para ser operada en cada uno de los casos.

Básicamente este submenú se utiliza para llenar las tablas catálogo, las cuales tiene la información que será utilizada después en cada asignación, las opciones que contiene este menú son las de configurar los "Cursos",

"Semestres, Rubros y Horarios", así como también poder cambiar las contraseñas de los usuarios de sistema.

Figura 29. **Opciones del submenú de configuración**



Fuente: elaboración propia.

En la opción de "Cursos" se puede agregar, eliminar y modificar cada uno de los cursos que se impartirán en el laboratorio, así como el código al cual están asignados, las tarjetas que están disponibles para el curso, la cantidad de estudiantes permitidos por grupo, y los períodos a la semana necesarios para el curso.

Figura 30. Administración de cursos

Código	Nombre del curso	Tarjetas disponibles	Estudiantes por grupo	Períodos semanales
*	Nuevo Curso			1
0232	Electronica 1	3	3	2
0234	Electronica 4	3	3	2
0235	Robotica	3	3	2
0236	Sistemas de Control 1	3	3	1
0240	Electronica 2	2	3	2
0242	Comunicaciones 1	2	3	2
0244	Comunicaciones 2	2	3	2
0245	Comunicaciones 3	3	3	1
0246	Electronica 3	3	3	2
0248	Electronica 5	2	3	2
0249	Electronica 6	2	3	1
0250	Comunicaciones 4	1	3	1
0330	Chatologia	2	3	2
0888	pics	1	14	1

Fuente: elaboración propia.

En la opción de "Semestres" se puede administrar el año en el que está cursando el semestre, si es el primero o segundo del año, la cantidad de estaciones disponibles, así como la fecha de finalización del mismo, entre otras opciones.

Figura 31. Administración de semestres

Año	Semestre	Prefijo	Activo
2006	01	L	N
2006	02	M	N
2007	01	N	N
2008	01	P	N
2008	02	Q	N
2015	01	R	S
2999	01	z	N

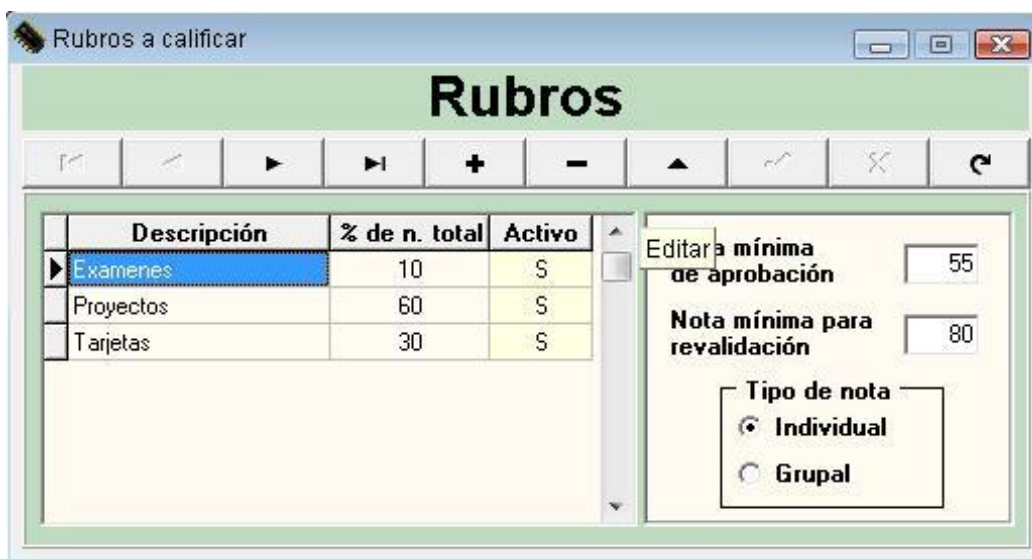
  

Estaciones disponibles	01	Fecha de vencimiento	30/06/2015
Nota mínima de aprobación	61	Fecha de vencimiento de rubros	30/06/2015

Fuente: elaboración propia.

Cuando se utiliza la opción de "Rubros" es importante destacar que en esta parte se detallarán las categorías en las cuales se va a desglosar el curso (laboratorio), estas pueden ser administradas desde la opción antes mencionada.

Figura 32. Administración de rubros

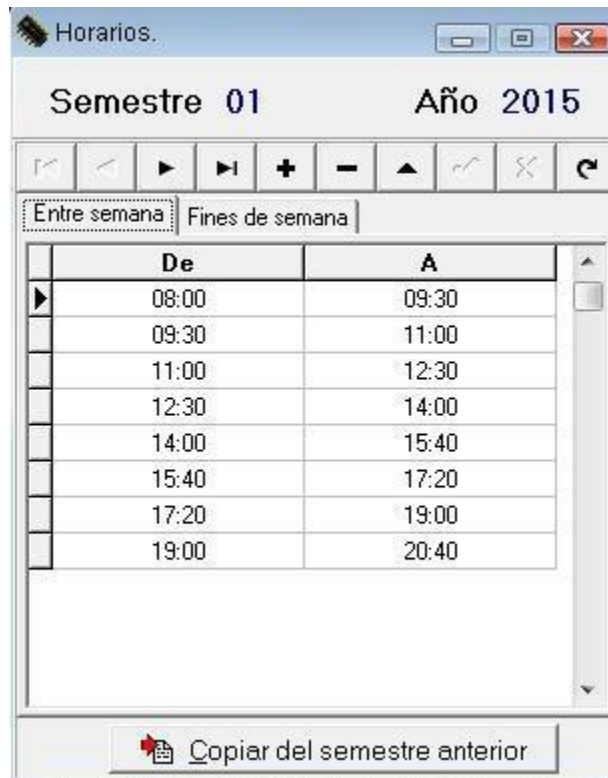


Fuente: elaboración propia.

Más adelante estos rubros serán adicionados para formar una nota total de laboratorio, lo cual permite saber a detalle, el porcentaje de la nota total que está siendo aplicado a cada categoría.

En la opción de "Horarios", se puede crear el desglose de fracciones de tiempo en los cuales será dividido el día de laboratorio, ya bien sea de días dentro de la semana o en fin de semana.

Figura 33. Administración de horarios



Fuente: elaboración propia.

En la opción de cambiar contraseñas, se realizan modificaciones a las claves de acceso de cada uno de los roles del sistema, tanto administrador (acceso total), como para ingresar asignaciones, reportaría, o modificaciones.

Al cambiar únicamente una de las contraseñas, a pesar de estar en el mismo formulario, las demás contraseñas permanecen intactas.

Figura 34. **Cambio de contraseñas**

The image shows a Windows-style dialog box titled "Contraseñas." with standard minimize, maximize, and close buttons in the top right corner. The dialog is divided into four distinct sections, each with a title bar and a light gray background. Each section contains two text input fields: "Contraseña" (Password) and "Confirmar contraseña" (Confirm password). The "Contraseña" fields are filled with asterisks, while the "Confirmar contraseña" fields are filled with dots. To the right of each section are two buttons: "Cambiar" (Change) with a document icon and "Cancelar" (Cancel) with a circular arrow icon. At the bottom center of the dialog is a "Cerrar" (Close) button with a window icon.

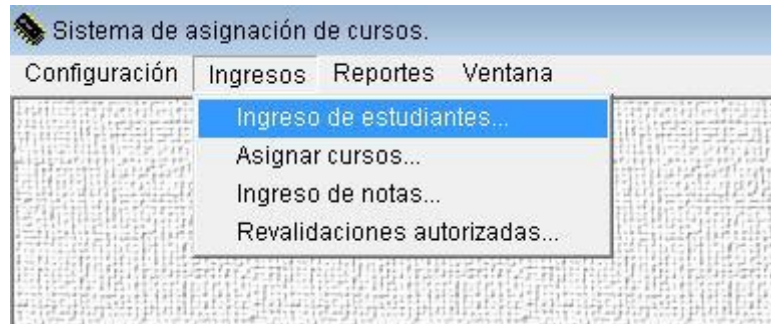
Fuente: elaboración propia.

### 3.1.2.3.2. **Submenú de ingresos**

Este submenú permite al auxiliar ingresar toda la información concerniente al alumno, asignaciones y notas, así como las revalidaciones de los cursos, este submenú, es el más utilizado ya que aquí se realiza el ingreso de la información que es cambiante durante el semestre.



Figura 35. **Submenú de ingresos**



Fuente: elaboración propia.

En la opción de "Ingreso de estudiantes" es donde se da de alta a cada uno de los estudiantes en el sistema, esta opción es utilizada cuando un estudiante es nuevo en el laboratorio y toda su información no está registrada en la base de datos.

Los datos a ingresar son identificativos de cada alumno, por ejemplo, el carnet del alumno, nombre completo, una dirección de correo electrónico, la dirección de casa y un número telefónico de referencia, el carnet del usuario es el identificador primario en la tabla que almacenará toda esta información, ya que este es único y personal, también se puede importar una fotografía que ayudará al auxiliar a identificar al alumno en la aplicación.

Figura 36. Ingreso de estudiantes

Estudiantes.

**Carnet** 000014768

**Nombres** Jose Victor

**Apellidos** Quiroa Rios

Correo electrónico jrquiroa43@hotmail.com

Dirección 5 calle A 11-49 zona 11

Teléfono 24716986

Importar imagen

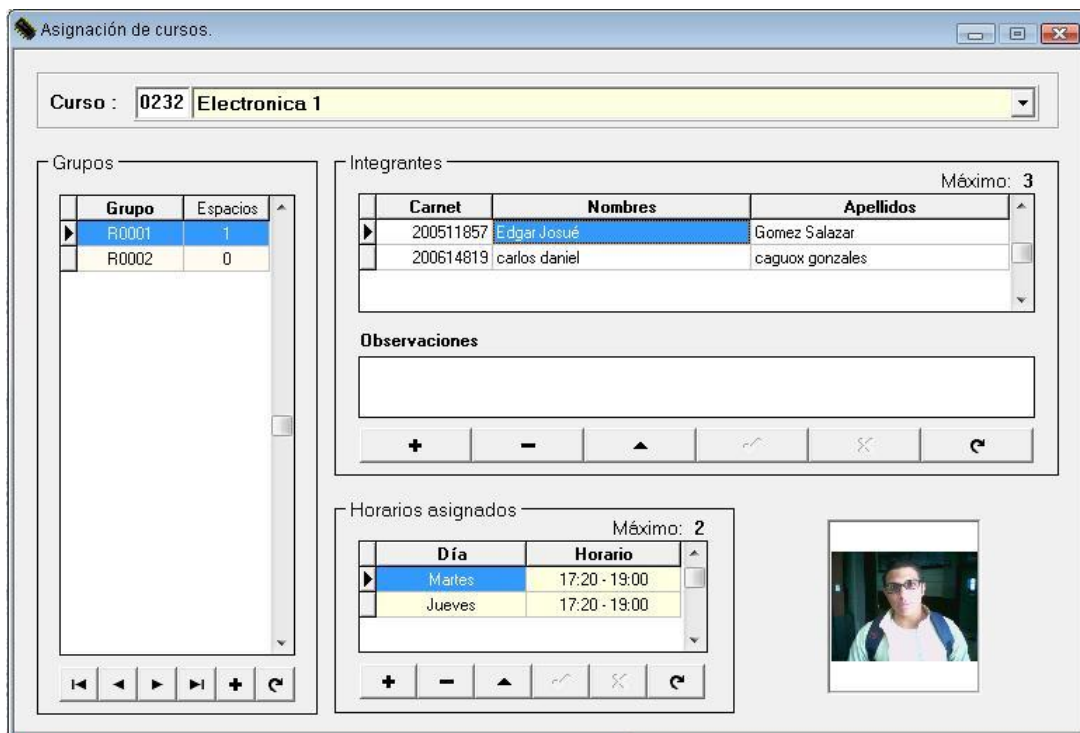
Estado: examinando

Fuente: elaboración propia.

En la opción de "Asignación de cursos", es donde el auxiliar puede ingresar y asignar a un grupo a los horarios del curso en cuestión.

A nivel de base de datos, esta ventana es la que relaciona la tabla que contiene a los grupos y la tabla que contiene los horarios, pudiendo asignar varios horarios al mismo grupo, con la salvedad del máximo de horarios definido en el submenú de configuración, anteriormente mencionado.

Figura 37. Asignación de cursos



Fuente: elaboración propia.

Eligiendo la opción de "Ingreso de notas por curso", el auxiliar puede realizar el ingreso de la nota de cada alumno en los rubros individuales, y la nota del grupo en los rubros grupales, esto que sumado será la nota del alumno en el laboratorio, al cual en la aplicación se llamará "Curso".

Todo lo mencionado anteriormente puede ser realizado seleccionando la opción de Ingreso de notas por curso, y luego seleccionando el curso en la lista de cursos que se muestra, luego de esto, en la ventana se mostrarán todos los alumnos asignados al curso, y así las calificaciones por rubro.

Figura 38. Ingreso de notas por curso

The screenshot shows a software interface for entering grades. At the top, there is a window title 'Ingreso de notas por curso.' and a course selection dropdown menu set to '0232 Electronica 1'. To the right, it indicates 'Estudiantes asignados en este curso' with a count of 5. Below this, there are two tabs: 'Rubros individuales' and 'Rubros grupales'. The main area is divided into two panels: 'Estudiantes asignados' and 'Calificaciones'. The 'Estudiantes asignados' panel contains a table with the following data:

Carnet	Nombre	Apellido	Grupo
200511857	Edgar Josué	Gomez Salazar	R0001
200611167	Jose Francisco	Sánchez Sánchez	R0002
200611513	julio Roberto	Xocoy Cruz	R0002
200611522	Henry Willyam	Camo Cojom	R0002
200614819	carlos daniel	caguox gonzales	R0001

The 'Calificaciones' panel contains a table with the following data:

Rubro	Nota
Exámenes	

Fuente: elaboración propia.

Por último esta la opción de "Revalidaciones autorizadas" la cual, permite realizar la revalidación de cursos los cuales han sido aprobados en semestres anteriores y han sido autorizados a revalidación según los criterios utilizados en el laboratorio.

Esto se realiza seleccionando el curso en el listado de la ventana, para luego presionar el botón + (más) el cual permitirá agregar la nota de revalidación del curso. Nótese que el sistema no realiza la verificación automáticamente, sino que el auxiliar debe buscar la nota de aprobación del alumno y colocarla en las revalidaciones.

Figura 39. Ingreso de revalidaciones autorizada

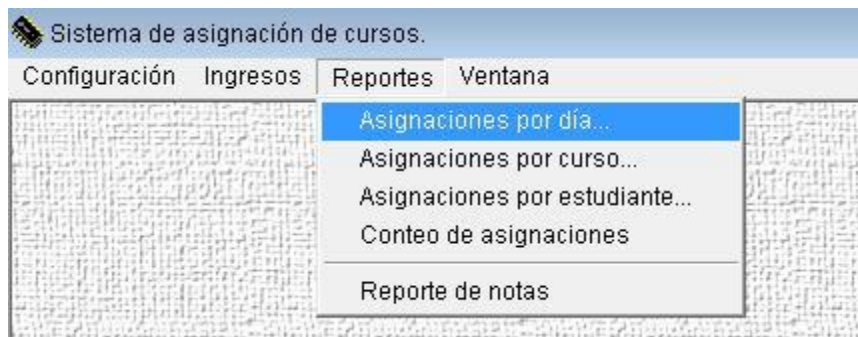
The screenshot shows a software window titled "Ingreso de revalidaciones autorizadas." with a standard Windows-style title bar. Below the title bar is a dropdown menu for "Curso" with "0232 Electronica 1" selected. Below this are two tabs: "Vigentes" (selected) and "Vencidas". The main area contains a table with the following columns: "Carnet", "Nombre", "Nota", "Fecha rev.", "Vig. desde", and "Vig. hasta". The first row of the table is highlighted in blue and contains double slashes (//) in the "Fecha rev.", "Vig. desde", and "Vig. hasta" columns. Below the table is a form with several input fields: "Carnet" (empty), "Nombre" (empty), "Nota de aprobación" (empty), "Fecha de revalidación" (//), "Vigente desde" (//) and "hasta" (//), and "Auxiliar que revalida" (empty). There is also a "Razonamiento" label followed by a large text area. At the bottom of the window is a toolbar with icons for adding (+), deleting (-), undo (↶), redo (↷), and refresh (↻).

Fuente: elaboración propia.

### 3.1.2.3.3. Submenú de reportes

Por medio de este submenú se puede acceder a toda la reportaría que puede brindar la aplicación, por medio de esa opción de menú, se puede desplegar la información de asignaciones por alumno, por días, por cada curso, notas, así como métricas de las asignaciones.

Figura 40. Submenú de reportes

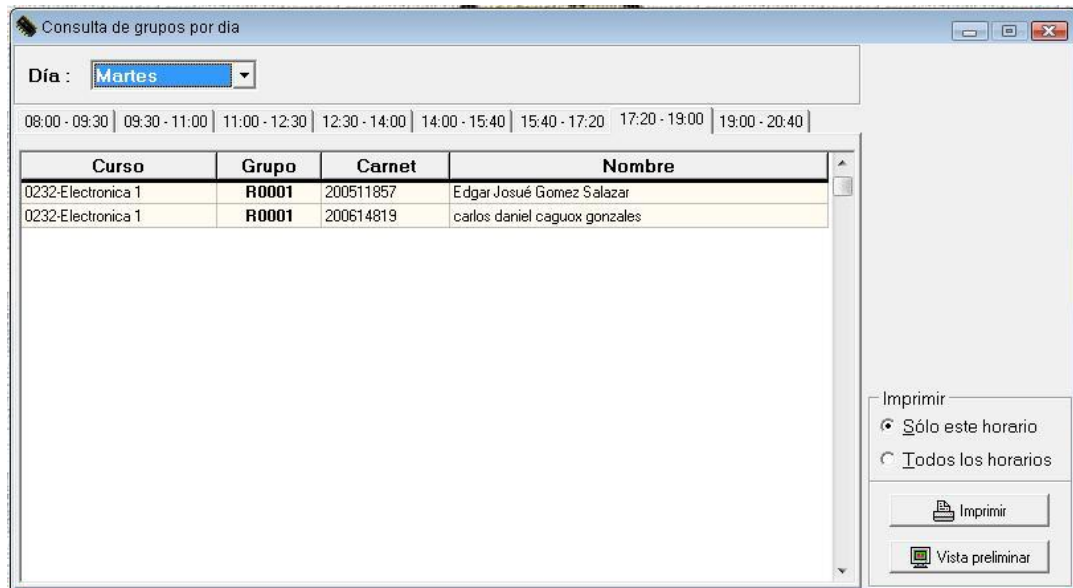


Fuente: elaboración propia.

Cuando se necesita ver los grupos asignados por cada día, se selecciona la opción de "Asignaciones por día", dentro de la ventana, se selecciona el día que se quiere desplegar de la lista, y saldrán varias pestañas con los diferentes horarios creados para este día, se selecciona la pestaña del horario de interés y se desplegarán los alumnos asignados en este día, a este horario, mostrando la información del curso asignado, el grupo al que pertenecen, el carnet y el nombre completo del alumno.

En esta ventana también es posible imprimir el reporte y tener una vista preliminar antes de la impresión, ya sea únicamente del horario seleccionado o de todos los horarios del semestre.

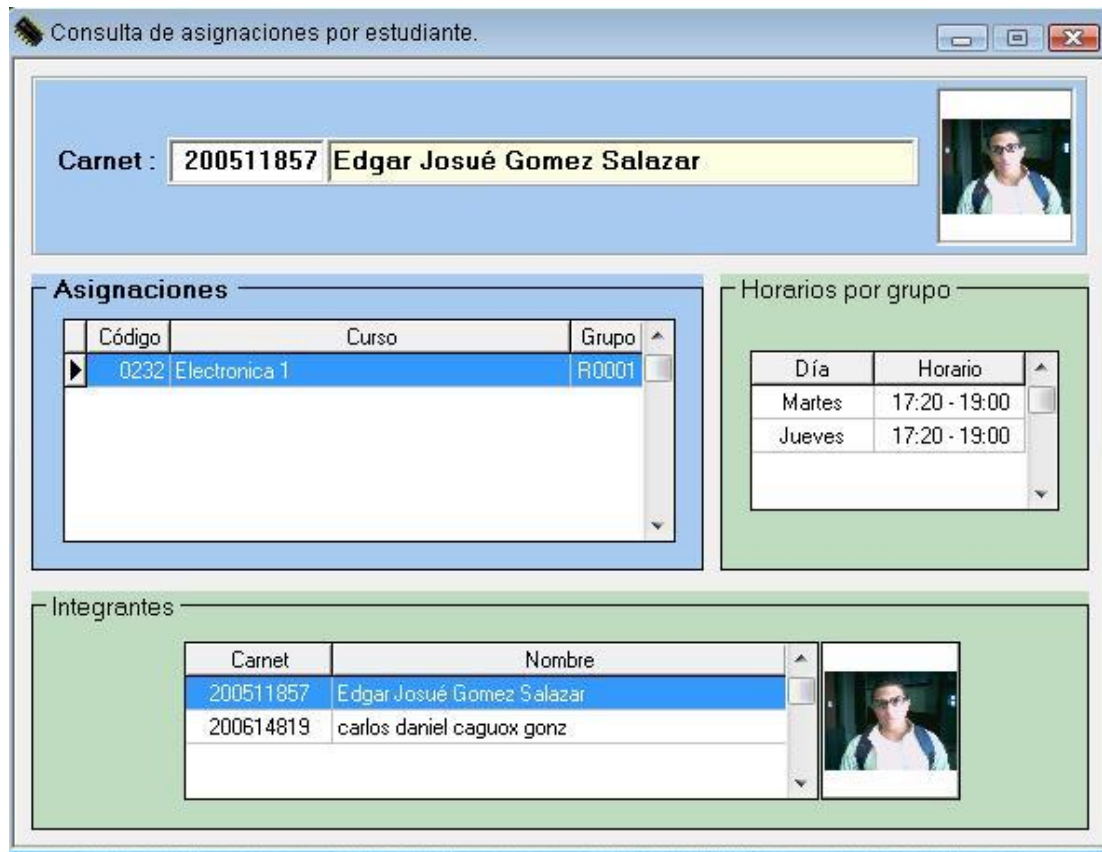
Figura 41. Asignaciones por día



Fuente: elaboración propia.

Quando se quiere conocer, los cursos asignados de cada estudiante, su grupo, y cuales horarios, se utilizará la opción de "Asignaciones por estudiante", la cual permitirá hacer una búsqueda por el número de carnet del alumno, esto desplegará el nombre del alumno seleccionado y toda la información concerniente a las asignaciones del alumno, conteniendo cada curso asignado, el grupo por medio del cual se asignó por curso, los días y horarios de cada curso asignado, y los integrantes del grupo, así como la fotografía del alumno.

Figura 42. Asignaciones por alumno



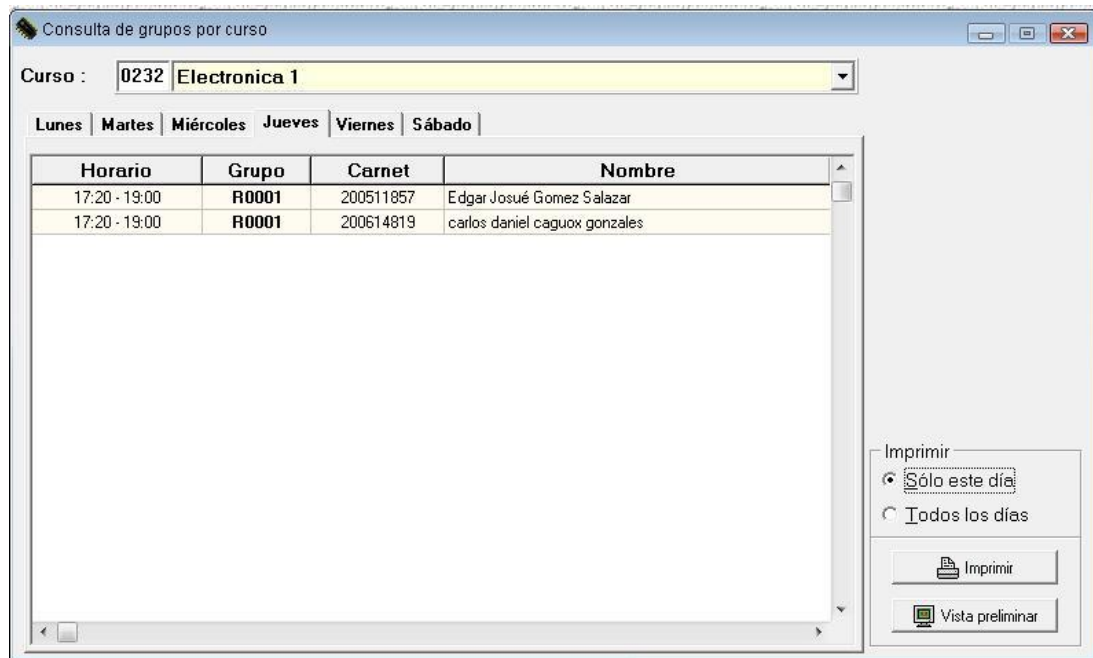
Fuente: elaboración propia.

También se puede consultar los grupos que tiene asignado cada curso, por medio de la opción de "asignaciones por curso", la cual será consultada seleccionando de la lista el nombre del curso creado, entonces se mostrará el código del curso, y las pestañas de los días que se imparten los mismos, en cada día se despliegan los horarios asignados, el código del grupo, el carnet del alumno y el nombre.

Este reporte puede ser impreso, seleccionando únicamente un día, o todos los días, así como utilizar la vista previa antes de enviar la impresión.



Figura 43. Grupos por curso

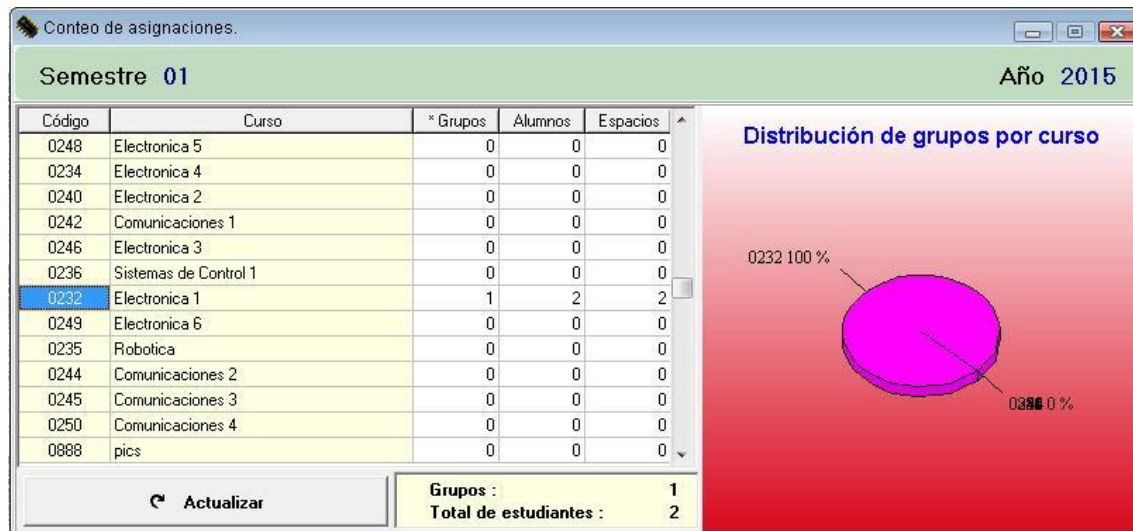


Fuente: elaboración propia.

Quando se requiere desplegar las métricas del semestre, se debe seleccionar la opción de "Conteo de asignaciones", el cual mostrará cuantos grupos, alumnos espacios, están siendo asignados por cada uno de los cursos del semestre.

Este reporte también muestra una gráfica de tipo *pie* en la cual se muestra la distribución de los grupos por cada curso, en la parte derecha de la ventana, dividiendo en colores cada curso, y mostrando porcentualmente la asignación de alumnos por curso.

Figura 44. **Conteo de asignaciones**



Fuente: elaboración propia.

Por último esta la opción de visualizar un reporte de "Notas por curso" el cual, al seleccionar el curso de la lista, muestra cada uno de los alumnos, número de carnet, nombre, el tipo de nota, y la nota por cada uno de los rubros.

Este reporte también permite filtrar la consulta por medio del tipo de nota a mostrar, como los que han sido por revalidación únicamente o todos.

También puede ser observado el historial de notas del alumno, lo cual muestra el rendimiento en el curso del alumno en semestres anteriores si ya ha sido cursado el mismo más de una vez.

## **3.2. El nuevo sistema**

Debido a los antecedentes mencionados, surge la necesidad de crear un sistema mucho más eficiente, que pueda ser accedido desde cualquier lugar, que pueda ser respaldado, y así también que cuente con la última tecnología, para que este se adapte a los nuevos dispositivos.

Así también se vió la necesidad de crear facilidades para el alumno y el auxiliar, por medio de factores que permitirán mayor eficiencia en el manejo de las asignaciones, las notas de cursos y los horarios.

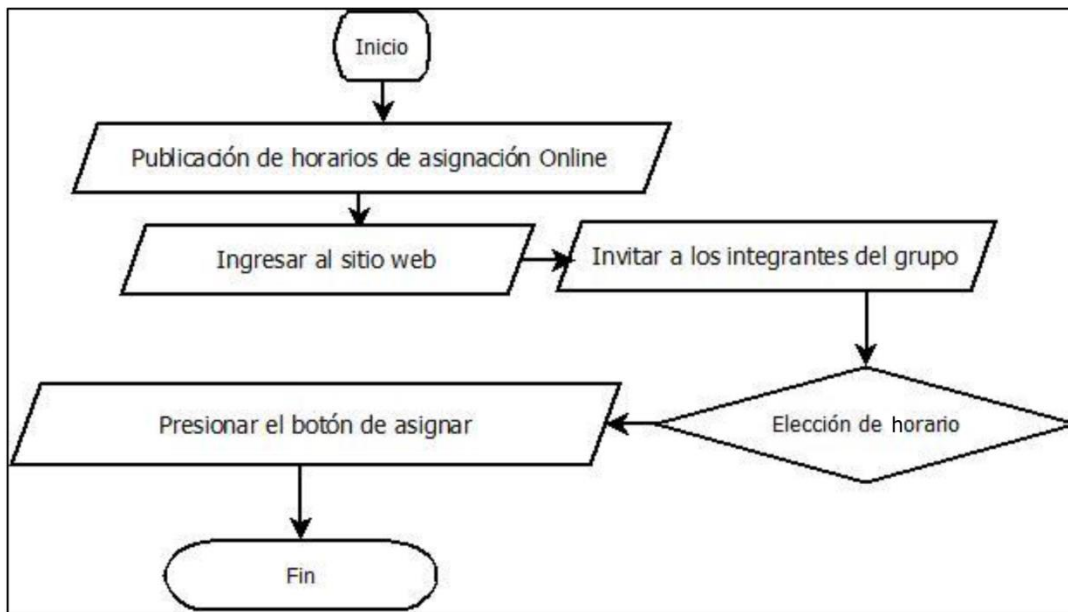
### **3.2.1. Descripción del nuevo proceso**

Debido a las necesidades mencionadas anteriormente, era necesario antes de desarrollar un nuevo sistema, diseñar un nuevo proceso, el cual evitará la intervención de tantos actores en el mismo, ya que a mayor cantidad de actores, el proceso se veía entorpecido.

Se diseñó un nuevo proceso en el cual, los alumnos no tuvieran la necesidad de hacer acto de presencia en el laboratorio para asignarse, ni tampoco era necesario concordar con sus compañeros en el mismo momento en el mismo lugar; un proceso en el cual los únicos actores son los alumnos, y las decisiones y validaciones las realice el sistema automáticamente. Es un proceso en el cual, a cualquier hora pueda ser asignado un curso (dentro de las fechas publicadas) sin la necesidad de tener a un auxiliar sentado frente a un escritorio anotando manualmente los grupos y las asignaciones.

Este proceso está descrito en la figura 45, por medio de la cual, se toma el proceso con un sistema que tiene flexibilidad y puede ser accedido desde cualquier parte.

Figura 45. **Diagrama del nuevo proceso de asignaciones**



Fuente: elaboración propia.

Como se observa, el número de pasos se convirtió de 9 a 7 sumado a esto, cada paso del nuevo proceso es más sencillo, lleva menos tiempo y no es susceptible al error humano. También se redujo el número de actores de 4 a 3, dejando únicamente como actores a los alumnos que pueden utilizar ellos mismos el sistema.

### **3.2.2. El nuevo sistema de asignación**

Una vez definido el nuevo proceso, era necesario decidir por el tipo de sistema a utilizar para poder cumplir con las partes del proceso y brindar las ventajas por la cual fue elegido.

Debido a su alta disponibilidad, confiabilidad, usabilidad, flexibilidad, la facilidad de trabajo colaborativo, a distancia y la seguridad de la información, era indispensable elegir un nuevo sistema, más no un cliente-servidor sino en un ambiente web, para que los alumnos, el auxiliar, en resumen todos los usuarios, puedan acceder a ella, desde su casa, trabajos, o cualquier lugar, brindando así una ventaja abismal para todos sobre el sistema anterior.

Se ha elegido utilizar una plataforma basada en .NET como lenguaje de desarrollo, basados en su facilidad de desarrollo, la adaptabilidad con las nuevas tecnologías y la cantidad de controles comprobados, los cuales brindan mayor facilidad a la hora de desarrollar.

Así también, se ha elegido SQL Express 2012 como motor de bases de datos, debido a su compatibilidad con los sistemas .NET, debido a que esta versión es de licencia gratuita y a que utiliza el estándar SQL.

El migrar a un nuevo sistema conlleva a que sea necesario utilizar nuevos equipos, modificar la estructura de la base de datos y diseñar un ambiente web, que permitan que el sistema, sea lo que promete ser, un sistema robusto.

### 3.2.2.1. El equipo

Ya que se ha elegido desarrollar una aplicación en un ambiente web, es necesario tener un servidor que responda a las peticiones de los usuarios, que en este nuevo proceso pueden llegar a ser muchas al mismo tiempo, por lo cual, debe tener cierta capacidad de procesamiento, y debe tener la capacidad de almacenar una buena cantidad de datos en la memoria aleatoria, para poder así responder a todas las peticiones, y evitar el encolamiento de las solicitudes.

Basados en la cantidad de usuarios que pueden estar utilizando dicha aplicación al mismo tiempo.

Tomando la recomendación que hace Microsoft en su sitio web, para montar un sitio web, con sistema operativo Windows, y que responda a las necesidades de este aplicativo es necesario como mínimo contar con las siguientes características:

Tabla III. **Especificaciones mínimas de un servidor web**

<b>Característica</b>	<b>Valor</b>
Sistema Operativo	Windows server 2008 o superior
Memoria RAM	1 GB o Superior
Procesador	2 procesadores Cual core IntelXeon a 2.8 gigahertz
Disco duro	60 GB o superior

Fuente: elaboración propia.

El disco duro, se ha elegido, basado en que este servidor alojará la aplicación y la base de datos del sistema, la cual puede llegar a crecer a medida que transcurren los semestres, entonces proyectando este servidor a

largo plazo, es necesario tener como mínimo esta capacidad de almacenamiento.

### **3.2.2.2. La nueva base de datos**

Luego de haber elegido el tipo de sistema a utilizarse, era de suma necesidad readecuar la base de datos para que fuera manejable por el lenguaje de desarrollo, y así también al Framework que se utilizará en este caso. El primer cambio de fondo es modificar los nombres de las tablas al idioma inglés, este cambio radical está ligado a que al utilizar Entity Framework , y al ser este el manejador de los datos, el mismo inserta nuevas tablas en el contexto, todas estas están ligadas al manejo de la seguridad, los eventos, roles, personalización de usuarios, rutas, módulos, entre otras. Estas nuevas tablas son insertadas en Inglés, por lo cual, las tablas antiguas fueron traducidas, para mantener un estándar del idioma de las mismas.

A continuación se detallará cada una de las tablas, desde las que vienen arrastradas del sistema anterior, hasta las que serán insertadas en el nuevo sistema:

- Student: maneja toda la información personal del estudiante, incluyendo la ruta en la que está almacenada su foto, para poder ser utilizada por la aplicación, al igual que en la base de datos anterior.
- Entry: anteriormente llamada la tabla de "rubros", maneja cada una de las subcategorías en las que se divide el curso, por ejemplo, tarjetas, asistencia, nota mínima de aprobación, entre otros.
- Course: almacena cada uno de los cursos disponibles, así como la cantidad de tarjetas asignadas al mismo, los periodos por semana, al igual que en la tabla "Cursos" en la base de datos anterior.

- Group: es muy básica, únicamente contiene el número de grupo y el prefijo asignado según el curso, así como el código del mismo, al igual que en la tabla "Grupos" en la base de datos anterior.
- Schedule: contiene la información de los intervalos de tiempo en los cuales pueden ser asignados los cursos, especificando si es fin de semana, el tiempo del intervalo, entre otros. Contiene la misma información que la tabla "Horarios", de la base de datos anterior.
- Day: es una tabla catálogo la cual se utiliza para asignar un código a cada día de la semana a nivel de base de datos, al igual que anteriormente en la tabla "Días".
- Assignment: relaciona un correlativo de asignación, al alumno por su carnet, y la nota final del laboratorio. Esta tabla era anteriormente llamada "Asignaciones".
- Authorized: define cada una de las revalidaciones que han sido validadas a solicitud del el estudiante, incluyendo su vigencia, la fecha en la que el curso fue aprobado, el auxiliar del laboratorio y la razón de la revalidación. En la base de datos anterior, esta tabla era llamada "Rev\_Autorizadas".
- Prefix: define los prefijos de cada uno de los grupos y los cursos que serán utilizados en el momento de la asignación. Antes se le llamaba a esta tabla "Prefijos".
- Grade: relaciona a cada alumno con la nota asignada en un rubro específico del curso, esta tabla anteriormente era llamada Notas.
- SummaryNote: contiene la sumatoria de todos los rubros por alumno por curso, anteriormente llamada ResumeNota.
- ScheduleXGroup: en este punto, se relacionan los grupos con un horario en específico, lo cual también reserva un lugar para cada grupo, esta tabla puede encontrarse en el diagrama anterior como HorariosXGrupo.



- AspNet\_WebEvent\_Events, Sysdiagrams, AspNet\_SchemaVersions, Constant, AspNet\_Profile, AspNet\_Users, AspNet\_Roles, AspNet\_UsersInRoles, AspNet\_PersonalizationPerUser, AspNet\_Membership, AspNet\_PersonalizationAllUsers, AspNet\_Applications, AspNet\_PathsMenuXRole, Menu, Module son las tablas que introduce el Entity Framework de ASP, por medio de las cuales maneja, los roles, permisos, menús, y personalización de los usuarios, así como los esquemas, las constantes, módulos, y eventos.
- Message: esta tabla es nueva, debido a que el sistema, está diseñado para que tenga una notificación de "Nuevo mensaje directo", según las solicitudes realizadas por otros alumnos, o así bien mensajes del auxiliar hacia el alumno.

### **3.2.2.3. La nueva aplicación de asignaciones**

La nueva aplicación de asignaciones fue diseñada para que funcione en un ambiente web, basado en ASP.NET, utilizando las últimas tecnologías orientadas al Entity Framework, e integrando tecnologías de adaptabilidad como el Responsive Web Design.

Todos los cambios mencionados anteriormente, han sido orientados a beneficiar al alumnado que debe cursar los laboratorios, y a los auxiliares y maestros, para que fácilmente puedan gestionar la información concerniente a los cursos, para así agilizar el proceso de asignación y facilitar la gestión de notas y todos los procesos relacionados a dicha gestión.

En aras de cumplir el objetivo mencionado, se decidió convertir la aplicación de escritorio en una aplicación de tipo web, la cual reside en un servidor de aplicaciones web, y puede ser accedido por cualquier navegador

web que forme parte de la red, en el caso de manejarlo en una red local; si esta aplicación es publicada por medio de una dirección ip pública, en el algún sitio que preste servicios de *host* en la nube, podrá ser accedida desde cualquier parte del mundo por los alumnos, profesores y auxiliares. Sumado a esto se buscó darle al sitio web una tónica de red social, en la cual los alumnos puedan invitar a otros alumnos a sus grupos, recibir notificaciones en forma de mensajes, verificar su calendario de laboratorios, entre otros. Todo esto debido a la influencia que tiene las redes sociales en la vida estudiantil, haciendo que sea fácil la familiarización de los usuarios al nuevo sistema.

Al entrar al sitio web desde un navegador o dispositivo móvil, se encontrará con la página de inicio en la cual se podrá crear una nueva cuenta o ingresar con nuestras credenciales

Figura 46. **Página de inicio sitio web de asignaciones**



Fuente: elaboración propia.

Al entrar al sitio por primera vez es necesario registrarse y crear una nueva cuenta con la cual se podrá acceder posteriormente; en el registro del sitio web se solicitaran varios campos: "El Nombre de usuario o Carné" el cual será utilizado como identificador de usuario, el nombre de usuario será utilizado por los auxiliares y profesores, y el número de carné por los alumnos.

A continuación será necesario ingresar los "nombres y apellidos", así como la "dirección de residencia" y una "dirección de correo electrónico" de referencia, será necesario incluir un número "telefónico de contacto", subir una foto identificativa para el perfil, y elegir una contraseña para el acceso.

Figura 47. **Registro de nuevos usuarios**

The image shows a web registration form for 'Laboratorio de Electrónica'. The page has a dark header with the site name and navigation links for 'Registrarse' and 'Iniciar Sesión'. Below the header, there is a 'Inicio' link and a main heading 'Registrarse. Crear nueva cuenta.'. The form contains several input fields: 'Nombre de Usuario / Carné', 'Nombres', 'Apellidos', 'Dirección', 'Dirección de correo', 'Telefono', 'Foto' (with an 'Examinar...' button and a message 'No se ha seleccionado ningún'), and 'Contraseña'. A 'Confirmar contraseña' field is also present. A 'Registrar' button is located at the bottom left of the form area. The footer of the page contains the copyright notice '© 2015 - ELAB'.

Fuente: elaboración propia.

### 3.2.2.3.1. Acceso como usuario restringido

Habiendo creado un usuario, se debe ingresar al sistema por medio del "Inicio de Sesión", para lo cual se utilizará el usuario y contraseña creados, y se elige si se desea que el navegador mantenga logueada la sesión, lo cual no es recomendable hacerlo si se esta ingresando en un equipo de uso público.

En esta ventana también dará la opción de crear una cuenta si no se tiene, o si no se ha ingresado por medio de la página principal del sitio.

Habiendo ingresado al sitio se tendrá un menú horizontal las opciones "Inicio, Asignación de cursos, Ver notas, Calendario", las cuales se observan como en la figura 48.

Figura 48. Menú de estudiante



Fuente: elaboración propia.

Se tiene el menú de asignación de cursos en el cual cada alumno coordinador de grupo puede asignar al grupo completo al curso en el horario seleccionado dependiendo de la disponibilidad. Dentro de la opción de asignación se tendrán dos opciones: "Crear asignación y Buscar asignación", para crear una asignación basta con hacer clic en el botón de *crear* y será desplegado el formulario para crearla, ahí se elige el curso y se presiona el

botón de crear grupo, esto asignará el número de grupo y mostrará el número de períodos semanales necesarios para el curso, a continuación en el apartado de integrantes, se ingresa el número de carnet de otro de los integrantes y será desplegado el nombre del mismo y su fotografía, después de esto se debe presionar el botón de "Agregar" y estos será agregado al grupo, al mismo tiempo será enviada una notificación al otro alumno para que acepte la invitación al grupo, esta notificación la puede encontrar en la parte superior del perfil en el área de mensajes.

Luego de haber agregado a todos los integrantes se eligen los horarios para los períodos necesarios, cabe mencionar que la asignación será exitosa hasta el momento que todos los estudiantes hayan aceptado dicha asignación.

Figura 49. Asignación de alumnos

**Opciones**

Volver a Índice

---

**Crear nueva Asignación**

Curso: 234 - Electronica 4      Grupo #: 3

Espacios: 3      Periodos: 2

---

**Integrantes**

Camé

Nombre del Estudiante

**Agregar**

Carnet	Nombres	Apellidos	Coordinador	Invitación	Acciones
200112553	Marco	Gomez	<input type="checkbox"/>	Pendiente	
200511557	Edgar Josue	Gomez Salazar	<input checked="" type="checkbox"/>	Aceptada	

---

**Horarios**

Día: Elija uno      Horario: Elija uno

**Agregar Horario**

Día	Hora	Acciones
Mañes	08:00:00 - 09:30:00	
Jueves	15:42:00 - 17:20:00	

Fuente: elaboración propia.

En el menú "Ver notas" los alumnos podrán verificar las notas de sus cursos asignados con forme estas estén siendo ingresadas al sistema por el auxiliar, para esto únicamente es necesario que el alumno elija el año y el semestre que quiere consultar par que se muestren todas las notas de los cursos de dicho semestre.

Figura 50. Ver notas por semestre



The screenshot shows the 'Laboratorio de Electrónica' web application. The top navigation menu includes 'Inicio', 'Asignación de Cursos', 'Ver Notas', and 'Calendario'. The main content area is titled 'Ver Notas' and features two dropdown menus for 'Año' (set to 2015) and 'Semestre' (set to 1). Below these is a table with the following data:

Nombre del Curso	Exámenes	Proyectos	Tarjetas	Nota	Revalidar
Electronica 1	70	80	80	79	

At the bottom of the interface, there is a copyright notice: © 2015 - ELAB.

Fuente: elaboración propia.

Aquí mismo se podrá solicitar la revalidación de un curso de semestres anteriores para que sea válida la nota en el presente semestre.

Esta forma de ver las notas, aparte de dar privacidad al alumno, brinda notas previas de cada uno de los rubros del curso, para que el alumno pueda ver el desglose de su nota, y también pueda saberlas conforme la vaya ingresando.

La opción de calendario es otro de los valores agregados de este sistema, ya que le brinda al alumno visibilidad de los cursos asignados y en que horario han sido asignados los mismos, para que el alumno tenga pleno control del

momento en que debe presentarse al laboratorio. El alumno también puede verificar quienes son los miembros de su grupo dando clic sobre el horario asignado que quiera consultar.

Figura 51. Calendario de laboratorios



Fuente: elaboración propia.

En la parte superior derecha del perfil, se podrá observar el nombre de usuario y mensajes, así como la opción de cerrar sesión, presionando sobre el nombre se podrá cambiar la contraseña antigua por una nueva, en la parte de mensajes se podrán ver las solicitudes de formar parte de un grupo que se han enviado y en la parte de cerrar sesión.

Figura 52. **Cambio de contraseña**

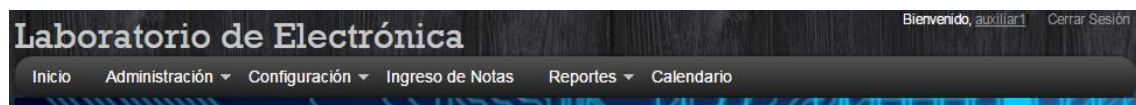
The screenshot shows the 'Laboratorio de Electrónica' website interface. At the top right, it says 'Bienvenido, Edgar Josue Mensajes (0) Cerrar Sesión'. Below this is a navigation bar with 'Inicio', 'Asignacion de Cursos', 'Ver Notas', and 'Calendario'. The main content area is titled 'Administracion de la cuenta.' and shows 'Estás conectado como 200511857.' Under the heading 'Cambio de contraseña', there are three input fields: 'Contraseña actual', 'Nueva contraseña', and 'Confirmar nueva contraseña'. A 'Cambio de contraseña' button is located at the bottom of the form.

Fuente: elaboración propia.

### 3.2.2.3.2. Acceso como auxiliar

Entrando al sistema con el perfil de auxiliar se tiene acceso a las opciones de menú "Inicio, administración, configuración, Ingreso de Notas, reportes y Calendario".

Figura 53. **Menú de auxiliar**



Fuente: elaboración propia.



En la opción de administración se coordina a los usuarios, crearlos o eliminarlos del sistema, así como buscar algún alumno para ver sus datos y su perfil.

Figura 54. **Administración de usuarios**

The screenshot shows a web interface titled "Lista de Usuarios". At the top, there are two buttons: "Crear Usuario" and "Buscar Usuario". Below these is a table with the following data:

Usuario	Nombres	Apellidos	Telefono	Email	Fecha	
Auxiliar1	Auxiliar1			auxiliar1@ingenieria-usac.edu.gt	17/04/2015 9:12:13 p. m.	 
200511857	Edgar Josue	Gomez Salazar	40220286	jossuche@hotmail.com	17/04/2015 9:09:40 p. m.	 
admin	admin			admin@elab.com	10/01/2015 2:46:44 a. m.	 
200112583	Marco	Gomez	22213003	tkvinicio@yahoo.com.mx	18/04/2015 3:33:25 p. m.	 
900082321	Oscar	Vicente	45124512	vicenteyan@gmail.com	17/04/2015 9:12:10 p. m.	 

Showing 1 - 5 of 5

Fuente: elaboración propia.

La opción de configuración permite administrar todo lo concerniente a los cursos, los semestres, los rubros de los cursos y los horarios activos por curso, todo esto para adecuar cada semestre a la disponibilidad de cursos, auxiliares, lugares de trabajo, entre otros.

Figura 55. Menú de configuración



Fuente: elaboración propia.

La primera configuración disponible es la de los cursos, por medio de esta misma se podrá listar los cursos creados así como la edición de los mismos, esto con el fin de poder dar mantenimiento al listado de cursos que aparecen disponibles para asignación de los alumnos

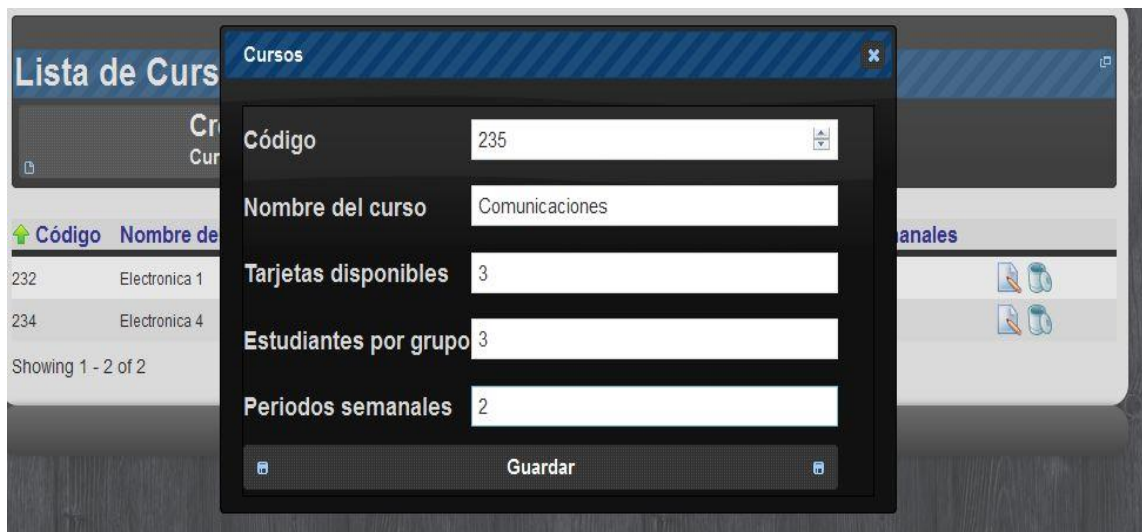
Figura 56. Lista de cursos

The image shows a web page titled 'Lista de Cursos'. It has two buttons: 'Crear Curso' and 'Buscar Curso'. Below is a table with columns for 'Código', 'Nombre del curso', 'Tarjetas disponibles', 'Estudiantes por grupo', and 'Periodos semanales'. There are two rows of data. At the bottom, it says 'Showing 1 - 2 of 2'.

Fuente: elaboración propia.

Cuando un curso nuevo será agregado a la disponibilidad de los semestres, es necesario crearlo, esto únicamente los usuarios administradores pueden realizarlo, esta tareas es bastante simple y se lleva a cabo presionando en el botón "Crear", por medio del cual se desplegará un formulario en el cual se ingresará la información correspondiente al nuevo curso.

Figura 57. **Crear curso**



The image shows a web application interface with a modal window titled "Cursos" for creating a new course. The modal contains the following fields and values:

Field	Value
Código	235
Nombre del curso	Comunicaciones
Tarjetas disponibles	3
Estudiantes por grupo	3
Periodos semanales	2

At the bottom of the modal is a "Guardar" (Save) button. In the background, a table titled "Lista de Cursos" is partially visible, showing columns for "Código" and "Nombre de" (likely "Nombre del curso"). The table lists two rows: "232 Electronica 1" and "234 Electronica 4".

Fuente: elaboración propia.

Las siguiente opción del menú de configuraciones es la administración de semestres, por medio de la cual, se dará de alta un nuevo semestre según corresponda, y así poder habilitar la asignación de los alumnos al mismo, los semestres disponibles son listados y pueden ser administrados desde la lista.

Figura 58. **Lista de semestres**



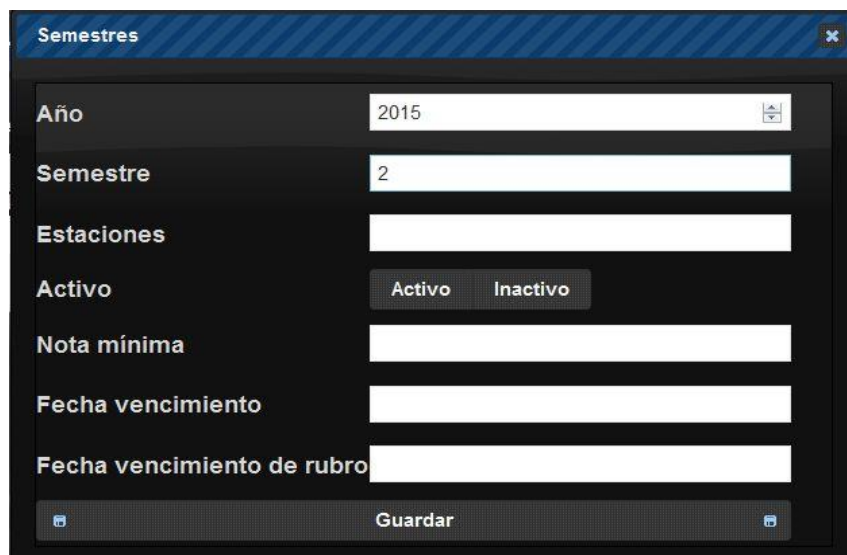
The screenshot shows a web interface titled "Lista de Semestres". At the top, there are two buttons: "Crear Semestre" and "Buscar Semestre". Below these is a table with the following columns: "Prefijo", "Año", "Semestre", "Activo", "Estaciones", "Nota mínima", "Fecha vencimiento", and "Fecha vencimiento de rubro". The table contains one row of data: "10", "2015", "1", a checked checkbox, "12", "61", "30/06/2015", and "30/06/2015". At the bottom left, it says "Showing 1 - 1 of 1".

Prefijo	Año	Semestre	Activo	Estaciones	Nota mínima	Fecha vencimiento	Fecha vencimiento de rubro
10	2015	1	<input checked="" type="checkbox"/>	12	61	30/06/2015	30/06/2015

Fuente: elaboración propia.

Para crear un nuevo semestre, basta con presionar el botón de "Crear" y esto desplegará un formulario para ingresar los datos del nuevo semestre, de la siguiente forma:

Figura 59. **Crear semestre**



The screenshot shows a form titled "Semestres" with the following fields and controls:

- Año:** A dropdown menu with "2015" selected.
- Semestre:** A text input field containing "2".
- Estaciones:** An empty text input field.
- Activo:** Two radio buttons labeled "Activo" and "Inactivo".
- Nota mínima:** An empty text input field.
- Fecha vencimiento:** An empty text input field.
- Fecha vencimiento de rubro:** An empty text input field.

At the bottom of the form is a "Guardar" button.

Fuente: elaboración propia.

Para que los cursos tengan el detalle y la división de las notas, es necesario crear rubros y luego administrarlos en caso sea necesario realizar modificaciones, para esto se utiliza la opción de "Rubros", en la cual se tendrá acceso a todos los rubros creados, ya sea para listarlos o editarlos, y así también acceso a crear nuevos rubros para los cursos:

Figura 60. Lista de rubros

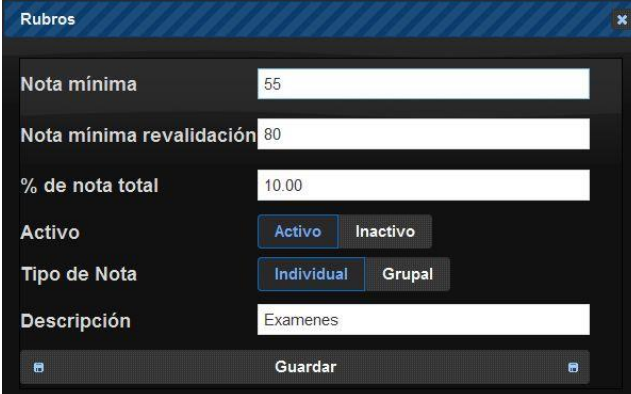


Id	Descripción	Nota mínima	Nota mínima revalidación	% de nota total	Activo	Tipo de Nota
9	Exámenes	55	80	10.00	<input checked="" type="checkbox"/>	Individual
10	Proyectos	70	80	60.00	<input checked="" type="checkbox"/>	Grupal
11	Tarjetas	70	80	30.00	<input checked="" type="checkbox"/>	Grupal

Showing 1 - 3 of 3

Fuente: elaboración propia.

Figura 61. Crear rubros



Rubros

Nota mínima: 55

Nota mínima revalidación: 80

% de nota total: 10.00

Activo:  Activo  Inactivo

Tipo de Nota:  Individual  Grupal





























Descripción: Exámenes

Guardar

Fuente: elaboración propia.

Por último, en la configuración es importante gestionar los horarios que se van a utilizar los cursos ya sea durante la semana o en el fin de esta, estos horarios son gestionados en la opción de "Horarios", en la cual se podran ver todos los horarios disponibles, editarlos, y así también crear nuevos horarios para los cursos.

Figura 62. Lista de horarios

Lista de Horarios				
Crear Horario		Buscar Horario		Copiar Horarios anteriores
Id	Tipo de Horario	Hora Inicio	Hora Fin	
32	Entre Semana	08:00:00	09:30:00	 
33	Entre Semana	09:30:00	11:00:00	 
34	Entre Semana	11:00:00	12:30:00	 
35	Entre Semana	12:30:00	14:00:00	 
36	Entre Semana	14:00:00	15:40:00	 
37	Entre Semana	15:40:00	17:20:00	 
38	Entre Semana	17:20:00	19:00:00	 
39	Entre Semana	19:00:00	20:40:00	 
40	Fin de Semana	07:00:00	08:40:00	 
41	Fin de Semana	08:40:00	10:20:00	 
42	Fin de Semana	10:20:00	12:00:00	 
43	Fin de Semana	12:00:00	13:40:00	 
44	Fin de Semana	13:40:00	15:20:00	 
45	Fin de Semana	15:20:00	17:00:00	 

Showing 1 - 14 of 14

Fuente: elaboración propia.

Figura 63. **Crear horario**

Fuente: elaboración propia.

La siguiente opción es la de "Ingreso de Notas", la cual permite al auxiliar ingresar los rubros de cada grupo o alumno dentro del sistema para que este pueda ser visualizado por el mismo y tomado en cuenta en la nota total de laboratorio, para lo cual debe elegir el curso, dar clic en el grupo e ingresar la nota que corresponda en cada uno de los casos, grupal o individual.

Figura 64. **Ingreso de notas**

Estudiantes asignados				Calificaciones	
Carnet	Nombres	Apellidos	Grupo	Rubro	Nota
200511857	Edgar Josue	Gomez Salazar	1	Exámenes	70
900082321	Oscar	Vicente	1		
200112583	Marco	Gomez	2		

Fuente: elaboración propia.

Luego está la opción de "Reportes" en la cual el auxiliar podrá consultar las asignaciones por día, por curso, por estudiante, y finalmente el reporte de notas en el cual se elige el curso y este desplegará un detalle de las notas de los alumnos.

Figura 65. **Reporte de notas**

The screenshot displays the 'Reporte de Notas' interface. At the top, there is a header 'Reporte de Notas'. Below it, a 'Curso' dropdown menu is set to '232 - Electronica 1'. To the right of the dropdown is a 'Recalcular Notas' button. Below the course selection, there are two main sections:

**Estudiantes**

Carne	Nombre	Tipo	Nota
200112583	Marco Gomez	Normal	77
200511857	Edgar Josue Gomez Salazar	Normal	79
900082321	Oscar Vicente	Normal	78

**Notas por Rubro**

Rubro	Nota
Exámenes	80
Proyectos	80
Tarjetas	70

Below these tables, there is a section for 'Semestres Anteriores' with a table structure showing 'Semestre', 'Rubro', and 'Nota' columns.

Fuente: elaboración propia.

La última opción del menú de auxiliar es la de "calendario", la cual es muy parecida a la que tiene los alumnos con la diferencia de que el auxiliar elegirá el curso que quiere ver en el calendario y se le mostrarán los grupos por día y hora asignada.

### 3.2.2.3.3. Acceso como administrador

Habiendo ingresado las credenciales, el sistema permitirá acceder al sitio web, y será mostrado el menú al cual se tiene acceso, en este caso el usuario "admin" tiene el rol de administrador por lo cual tiene acceso a todas las funcionalidades.



La primera de las funcionalidades como administrador es la de administrar los roles del sistema, esta pestaña le permite al administrador crear nuevos roles, administrar los existentes, y asignar un rol diferente a los usuarios que sea necesario

Figura 66. Lista de usuarios por role



Usuario	Nombres	Apellidos	Telefono	Email	Fecha
aux1	aux1			aux1@elab.com	17/02/2015 03:01:54 p.m.
900082322	Gabby	Monroy	45454545	vicent_hy@hotmail.com	15/02/2015 10:02:37 p.m.
200511857	Edgar	Salazar	40220286	jossuche@hotmail.com	15/01/2015 01:13:43 a.m.
admin	admin			admin@elab.com	10/01/2015 02:46:44 a.m.
900082321	Oscar	Vicente	56450076	vicenteyan@gmail.com	25/09/2013 03:32:13 a.m.
900108032	Angel	Vicente	44444444	avicente@gmail.com	11/01/2015 03:58:32 a.m.

Showing 1 - 6 of 6

Fuente: elaboración propia.

Puede realizarse también una búsqueda de roles dentro de la aplicación por medio del botón de "Buscar Role".

Figura 67. Buscar un rol



Fuente: elaboración propia.

Los usuarios pueden ser administrados de igual forma, su información puede ser editada, así como puede realizarse la búsqueda de usuarios por medio de un criterio.

Figura 68. **Buscar usuario**



Fuente: elaboración propia.

## 4. IMPLEMENTACIÓN FINAL

Habiendo realizado el desarrollo del proyecto, es necesario esclarecer la forma de implementarlo, por lo cual en este último capítulo, se explicará cada uno de los puntos necesarios de infraestructura para hacer funcionar la aplicación, así como las configuraciones necesarias para posteriores implementaciones, todo esto con el afán de facilitar las mismas después de entregado dicho proyecto.

### 4.1. Cumplimiento de requisitos de sistema

A continuación se describe el cumplimiento de requisitos de sistema con su respectivo ejemplo.

#### 4.1.1. Cumplimiento de requisitos de hardware

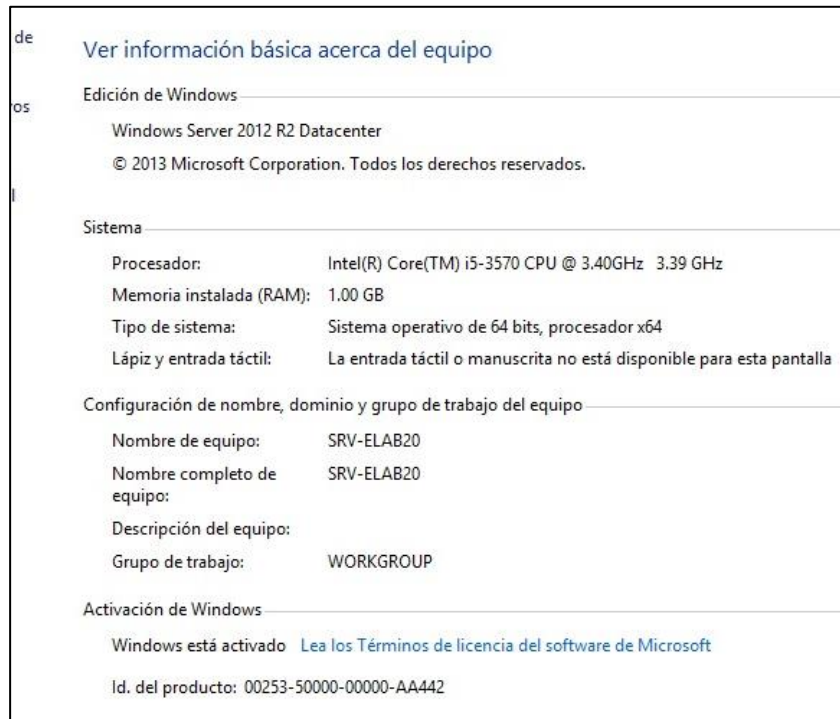
Derivado de la obligatoriedad de cumplir con los requisitos mínimos de hardware anteriormente mencionados, se procederá a verificar que el servidor en el que se va a instalar el servicio web, cumpla con los mismos:

Tabla IV. **Requisitos de hardware**

<b>Característica</b>	<b>Valor</b>
Memoria RAM	1 GB
Procesador	2 procesadores Core I5 a 3.4 gigahertz
Disco duro	60 GB

Fuente: elaboración propia.

Figura 69. **Cumplimiento de requisitos de ram y procesador**



Fuente: elaboración propia.

Figura 70. **Cumplimiento de requisitos de disco duro**



Fuente: elaboración propia.

Habiendo verificado que el hardware cumple con los requisitos mínimos, se procede a verificar el cumplimiento de los requisitos de software.

#### 4.1.2. Cumplimiento de requisitos de software

El servidor debe cumplir así también con varios requisitos de software los cuales garantizarán la estabilidad sistema, así como la compatibilidad de los componentes de la aplicación, estos requisitos se observan en la tabla V.

Tabla V. **Requisitos de software**

Sistema Operativo Windows Server 2008 o superior
Servicio WEB IIS 7 o superior
Microsoft SQL server 2008 o superior
Microsoft SQL Management 2008 o superior

Fuente: elaboración propia.

El sistema operativo es Windows Server 2012 R2 Datacenter, por lo cual el primero de los requisitos, que corresponde a sistema operativo, está cumplido.

Figura 71. **Sistema operativo**

[Ver información básica acerca del equipo](#)

Edición de Windows \_\_\_\_\_  
Windows Server 2012 R2 Datacenter  
© 2013 Microsoft Corporation. Todos los derechos reservados.

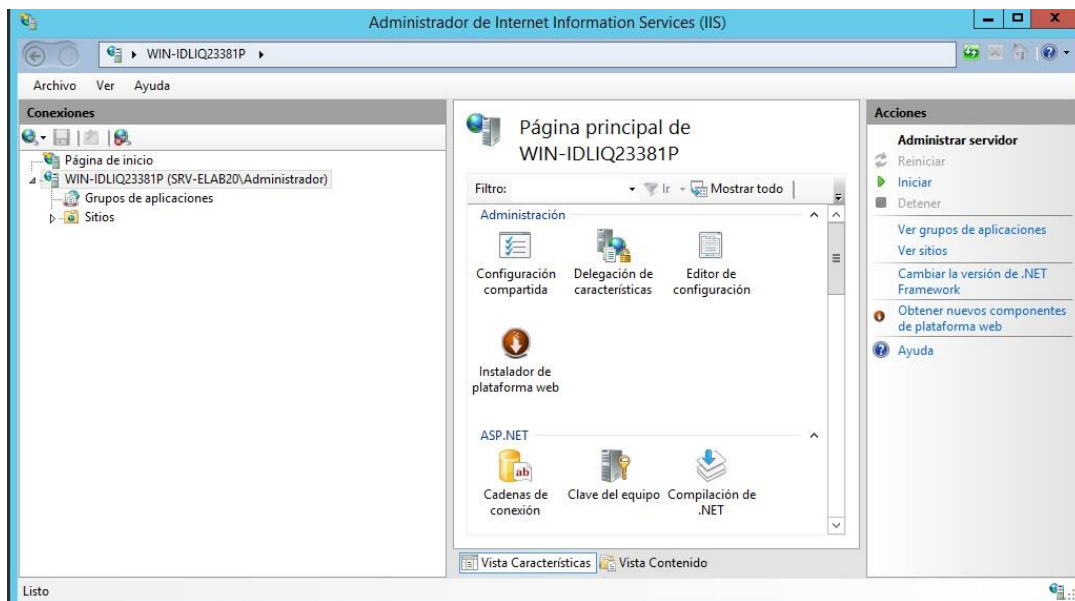
Sistema \_\_\_\_\_  
Procesador: Intel(R) Core(TM) i5-3570 CPU @ 3.

Fuente: elaboración propia.

Luego es necesario verificar que el servicio IIS esté instalado, el cual será el motor de respuesta a las peticiones web, y el cual alojará la aplicación.

Al verificar dentro del servidor, el Administrador de Internet Information Service se encuentra instalado en el mismo y el servicio detenido, ya que no tiene ninguna aplicación corriendo en este.

Figura 72. Verificación de IIS



Fuente: elaboración propia.

Luego de esto, es importante verificar que SQL server está instalado en el equipo, por lo cual se verifica que exista el servicio en el servidor, y así también el SQL Server Management Studio, el cual será necesario más adelante para realizar la configuración de la base de datos.

Figura 73. **Verificación de componentes de base de datos**



Fuente: elaboración propia.

Habiendo verificado que todo el software que se necesita, esté instalado en el servidor, el paso siguiente es su configuración, con todas las características que hará funcionar correctamente la aplicación.

## **4.2. Configuración del servidor**

La configuración del servidor está dividida en 3 etapas detalladas a continuación.

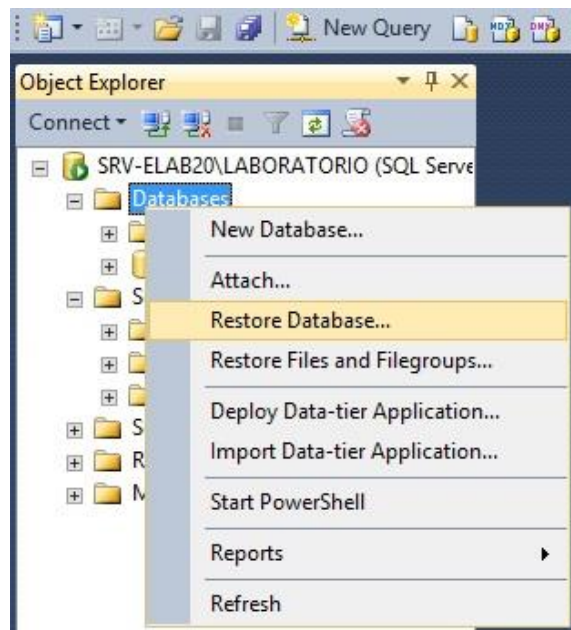
### **4.2.1. Configuración de base de datos**

Para que la base de datos sea funcional para la aplicación, es necesario restaurar un respaldo de la misma, y que así quede adjunta en el servidor,

sumado a esto, es necesario realizar algunas configuraciones para que la aplicación se comunique con ella de forma exitosa.

El primer paso es ingresar a la base de datos por medio del SQL Server Management Studio, ya dentro de la aplicación se debe dirigir a la sección del explorador de objetos y dando clic derecho sobre la carpeta de las bases de datos del sistema, se selecciona la opción "Restaurar Base de Datos",

Figura 74. **Explorador de objetos**

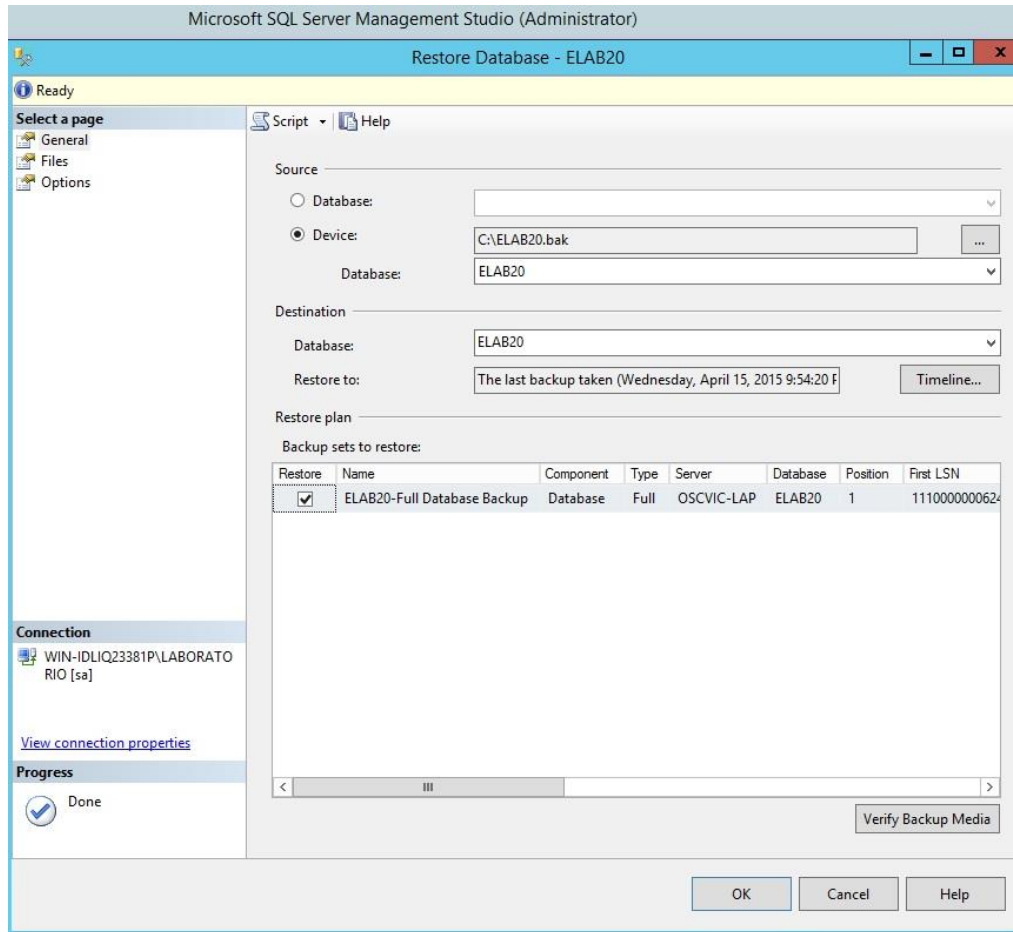


Fuente: elaboración propia.

Dentro de esta ventana, se debe seleccionar dispositivo, y desde donde se quiere montar la base de datos, así como también sobre qué base de datos se quiere restaurar y se presiona "ok", para iniciar la restauración.



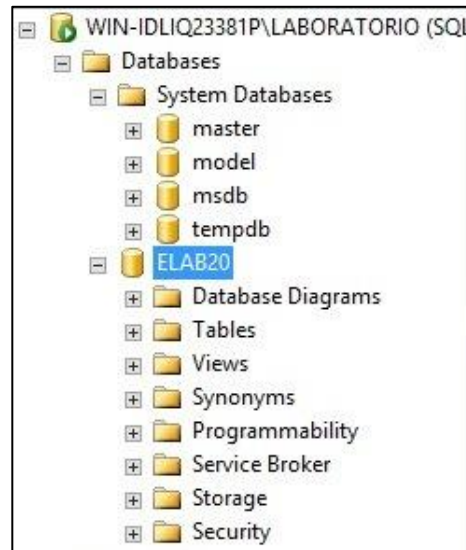
Figura 75. Restaurar base de datos



Fuente: elaboración propia.

Luego de esto, la base de datos con todas las tablas y procedimientos almacenados estará dentro del servidor.

Figura 76. **Base de datos ya restaurada**



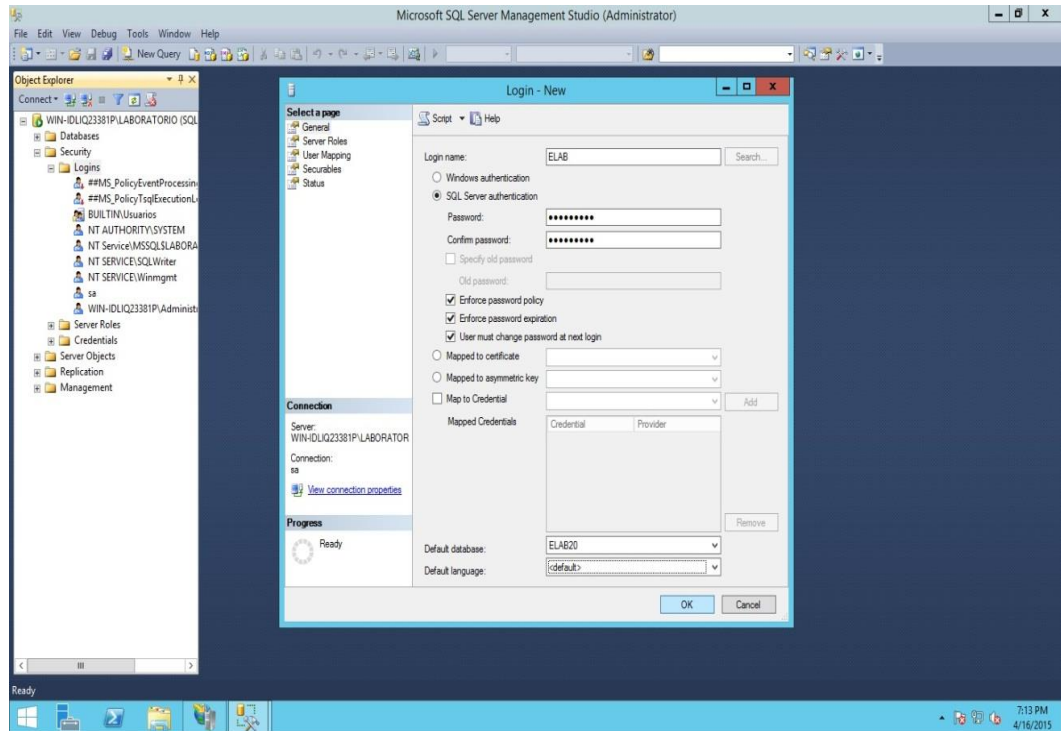
Fuente: elaboración propia.

Luego de haber restaurado la base de datos es necesario establecer los permisos sobre la misma para que la aplicación pueda conectarse exitosamente.

El primero de los permisos es en la configuración de la base de datos, en el explorador de objetos, se expandió la base de datos ELAB20, en el apartado de seguridad, se debe buscar la parte de "Usuarios para luego buscar el usuario ELAB y eliminarlo, esto se debe a que la restauración de la base de datos lo ubica el usuario en la base de datos directamente y no le brinda permisos sobre el servidor completo.

Seguido de esto dirigirse a la parte de seguridad del servidor, al apartado de acceso, dentro de este se crea un nuevo usuario llamado ELAB, colocando la misma contraseña que se colocó en la aplicación.

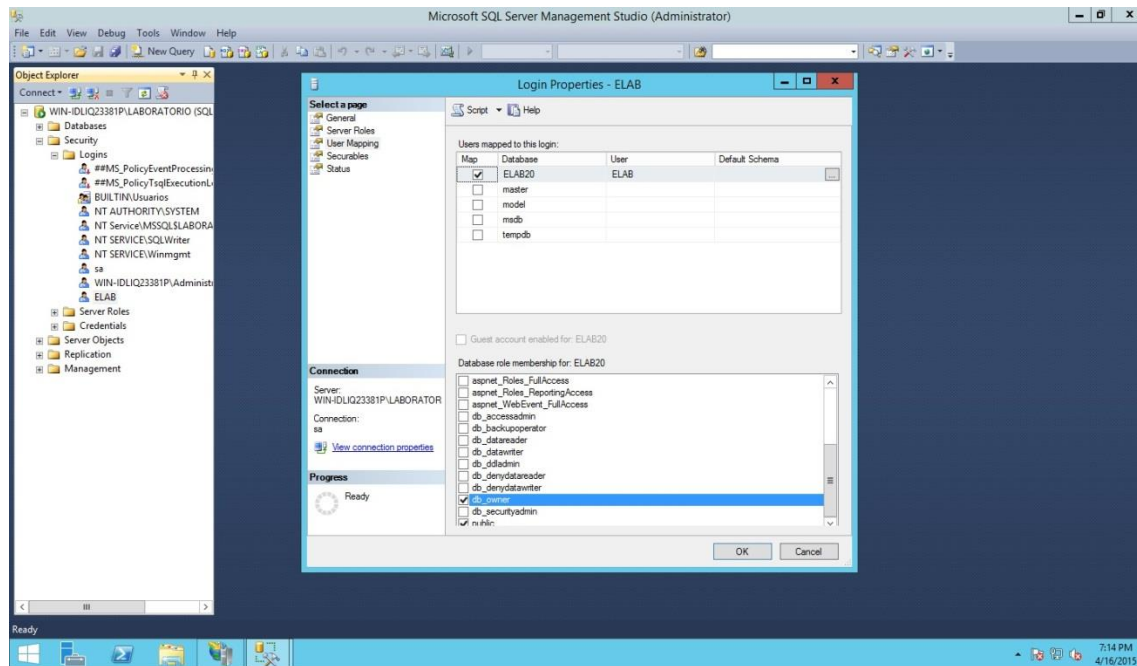
Figura 77. Crear nuevo usuario



Fuente: elaboración propia.

Para luego dirigirse al apartado de "User Mapping", en este apartado se brindarán los permisos del usuario a la base de datos de ELAB20, en la parte inferior de la pantalla, casi al final de los permisos disponibles, se seleccionan los permisos de *db\_owner* y *Public*, con eso se logra que el usuario ELAB20 sea propietario de la base de datos y pueda realizar todas las operaciones necesarias.

Figura 78. Permisos sobre la base de datos



Fuente: elaboración propia.

Por último se debe presionar "ok", y salir de la administración del usuario, para así salir al explorador de objetos nuevamente.

Luego de esto, procede desconectarse de la base de datos, para luego intentar ingresar al servidor pero ahora con las credenciales del nuevo usuario que se creó, esto para verificar que las mismas sean válidas, y tengan permitido acceder a la base de datos de ELAB20.

Figura 79. **Acceso como ELAB**



Fuente: elaboración propia.

Después de haber confirmado el acceso, ha finalizado la configuración a nivel de base de datos.

#### **4.2.2. Configuración de IIS**

Al llegar a esta parte, es bueno haberse asegurado que el IIS esté instalado en el equipo, en esta parte se muestra las características de IIS que se debe configurar para que el sitio funcione correctamente, y no se tenga ningún error al momento de que los usuarios accedan al mismo.

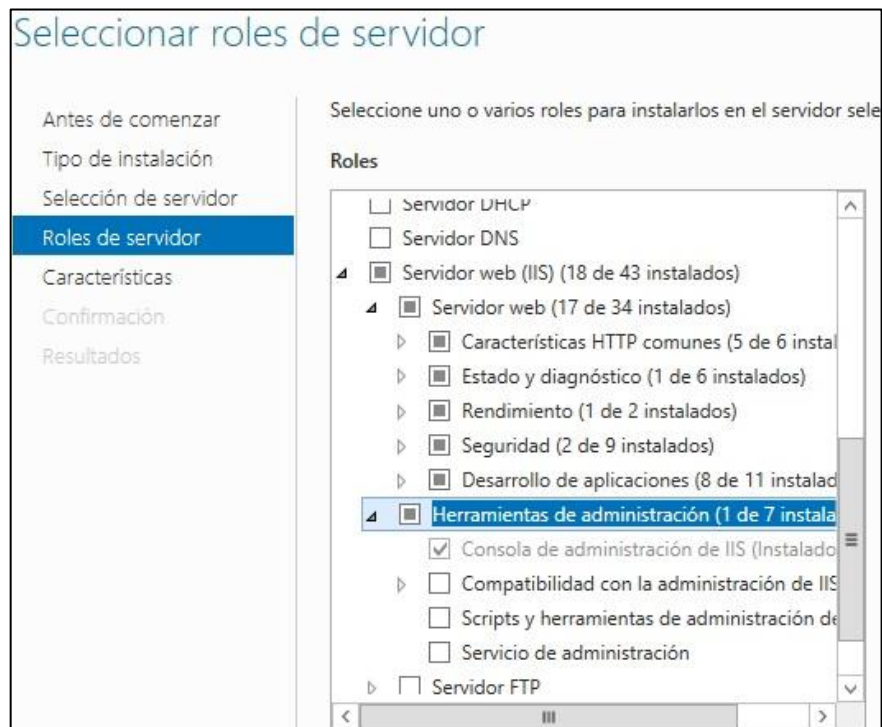
Para verificar esto, dirigirse al "wizard" para activar o desactivar características de Windows, en el apartado de Roles de servidor, se podrá

observar un rol llamado *Servidor web (IIS)*, el cual se debe expandir, y activar los siguientes subroles:

- Características HTTP comunes
  - Contenido estático
  - Documento predeterminado
  - Errores HTTP
  - Examen de directorios
  - Redireccionamiento HTTP
  - Estado y diagnostico
  - Registro HTTP
  
- Rendimiento
  - Compresión de contenido estático
  
- Seguridad
  - Filtrado de solicitudes
  - Autorización para URL
  
- Desarrollo de aplicaciones
  - ASP
  - ASP.NET 3.5
  - ASP.NET 4.5
  - CGI
  - Extensibilidad de .NET 3.5
  - Extensibilidad de .NET 4.5
  - Extensiones ISAPI
  - Filtros ISAPI

- Herramientas de administración
  - Consola de administración de IIS

Figura 80. Roles servidor



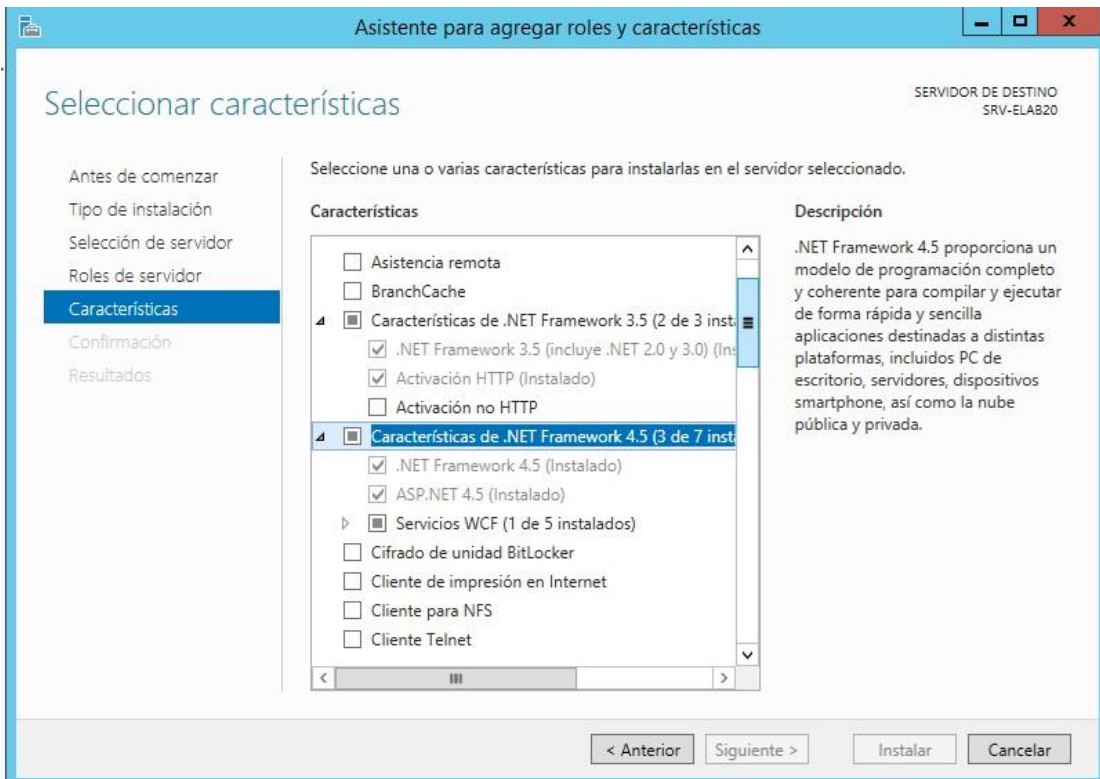
Fuente: elaboración propia.

Al presionar "Siguiente" se observan las características de Windows, y en este caso, se debe centrar en las características del *framework*:

- Características de .NET Framework 3.5
  - .NET framework 3.5
  - Activación de HTTP

- Características de .NET Framework 4.5
  - NET framework 4.5
  - ASP.NET 4.5
  - Servicios WFC
    - Uso compartido de puertos TCP

Figura 81. **Características de servidor**



Fuente: elaboración propia.

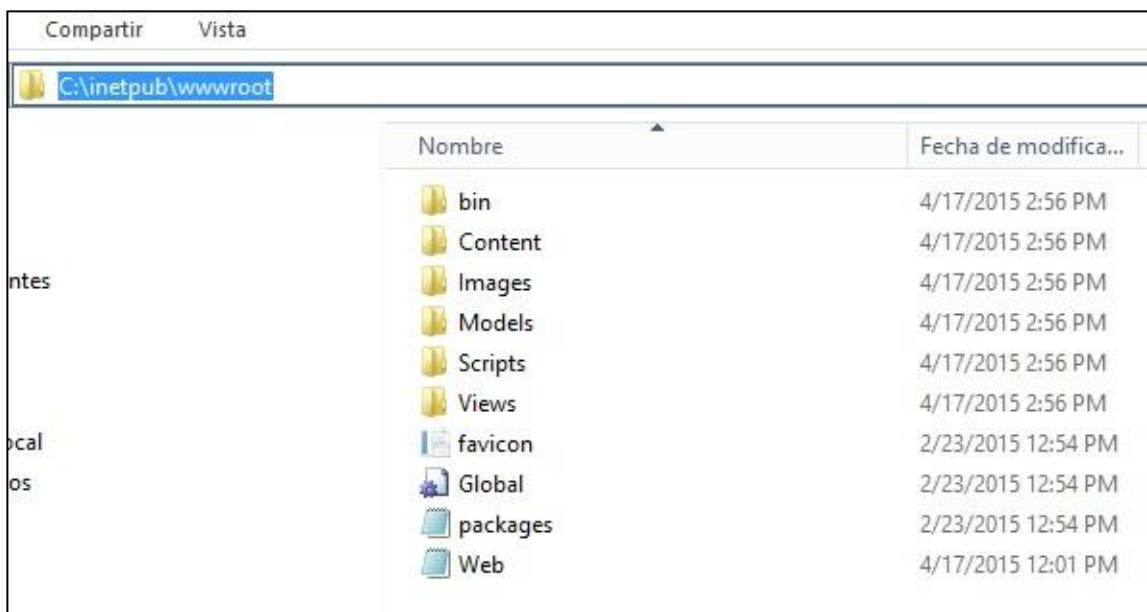
De no estar alguna de estas características seleccionadas, será necesario instalarlas únicamente presionando el botón de "Instalar".



Al desarrollar el sitio web con MVC, no se tendrá un formulario como el típico *Index.aspx*, sino que MVC utiliza un ruteo interno por medio de *dll* para moverse entre vistas.

Estos *dll* son generados en el momento de compilar el proyecto y deben ser copiados en la dirección *C:\inetpub\wwwroot* para que el IIS los reconozca y los despliegue.

Figura 82. Directorio **wwwroot**

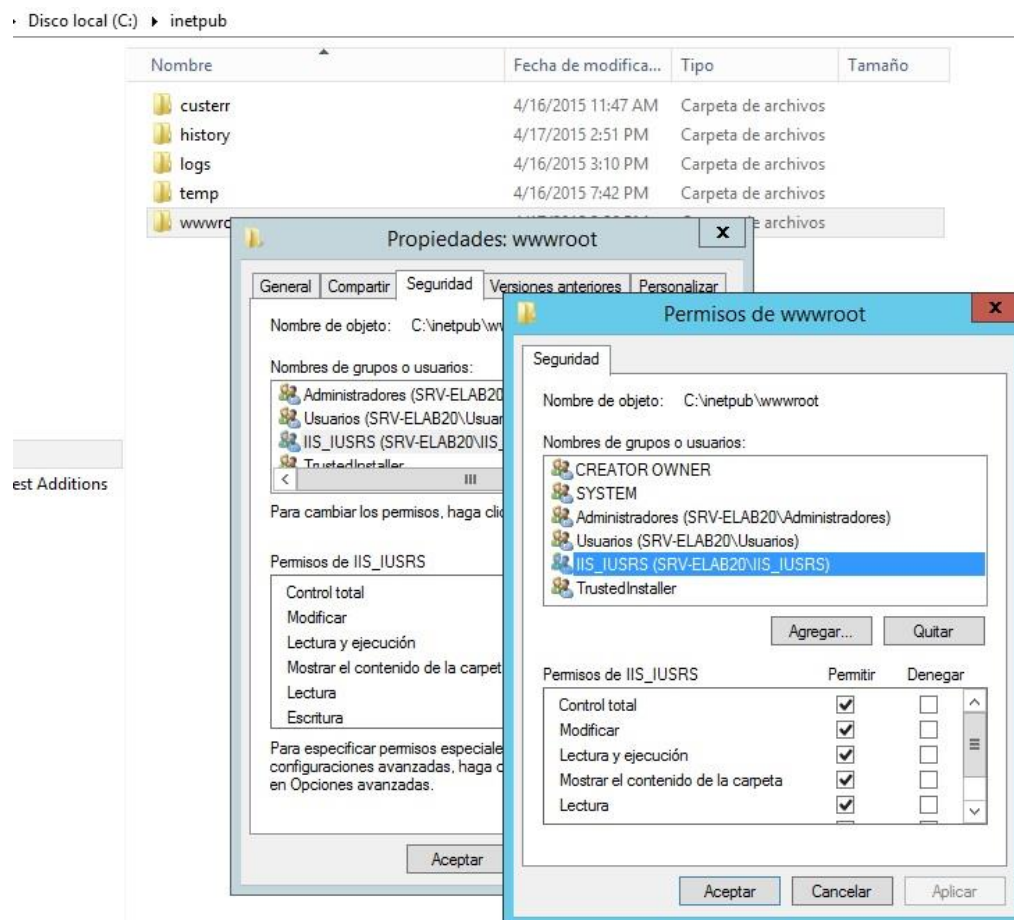


Fuente: elaboración propia.

Para que el servicio de IIS pueda acceder a esta carpeta, es necesario darle permisos al grupo de usuarios del servicio y que este grupo tenga control total sobre dicho directorio.

El grupo que utiliza IIS para manejar esta seguridad es el que tiene el nombre de IIS\_IUSRS, para este se seleccionará la casilla de "Control total", la cual seleccionará todas las demás por defecto.

Figura 83. Permisos de IIS



Fuente: elaboración propia.

Luego de esto, se reiniciará el servicio de IIS, para que los cambios cobren efecto.

Habiendo realizado esto, el sitio web podrá ser visitado localmente en un navegador web por medio de la dirección `http://localhost`.

Figura 84. Vista local del sitio



Fuente: elaboración propia.

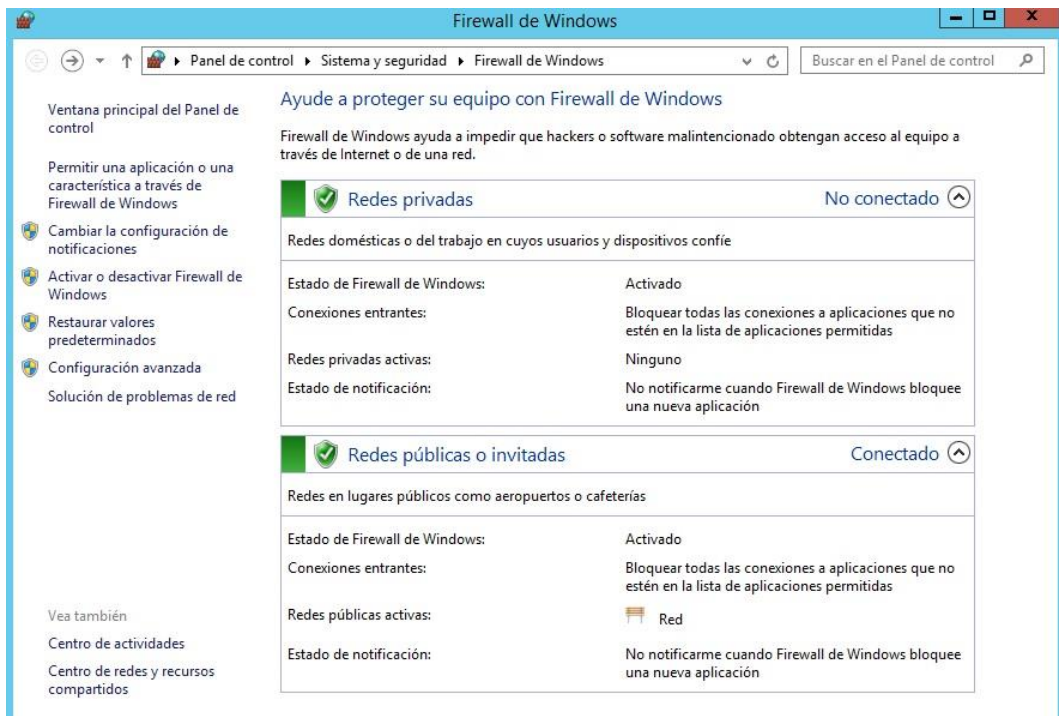
#### 4.2.3. Configuración de *firewall* local

Habiendo comprobado que la aplicación esté funcionando de manera local, es importante realizar algunos procedimientos para que el sitio pueda ser visible desde el exterior.

Esto es posible gracias a una ip pública configurada en la interfaz del equipo, pero eso puede ser peligroso, por lo tanto se debe verificar el estado del *firewall* de sistema operativo, para únicamente permitir las solicitudes que estén dirigidas al puerto 80.

Como primer paso abrir la administración del *firewall* en el servidor y verificar que se encuentre en estado Activado.

Figura 85. Estado de *firewall*



Fuente: elaboración propia.

De no estar activado, el *firewall* se podrá activar en la opción "Activar o Desactivar Firewall de Windows".

Luego de haberse asegurado de que el "*firewall*" está activado, es necesario verificar que el puerto 80 por medio del cual llegarán las peticiones de HTTP esté abierto. Para esto debe dirigirse en el panel izquierdo a la opción "Configuración Avanzada", esto abrirá una ventana llamada "Firewall de Windows con seguridad avanzada". Dentro de esta ventana, en el panel

izquierdo seleccionar la opción "Reglas de Entrada", y verificar que únicamente la regla "Servicios de Word Wide Web (entrada de tráfico HTTP)" este habilitada.

Figura 86. **Regla de tráfico HTTP**

<input type="radio"/>	Registros y alertas de rendimiento (DCOM de entrada)	135
<input type="radio"/>	Registros y alertas de rendimiento (DCOM de entrada)	135
<input type="radio"/>	Registros y alertas de rendimiento (TCP de entrada)	Cualquiera
<input type="radio"/>	Registros y alertas de rendimiento (TCP de entrada)	Cualquiera
<input type="radio"/>	Servicio de administración de claves (TCP de entrada)	1688
<input type="radio"/>	Servicio de Net Logon (NP de entrada)	445
<input type="radio"/>	Servicio de Net Logon Authz (RPC)	Puertos diná...
<input type="radio"/>	Servicio iSCSI (TCP de entrada)	Cualquiera
<input checked="" type="radio"/>	Servicios de World Wide Web (Entrada de tráfico HTTP)	80
<input type="radio"/>	Servicios de World Wide Web (Entrada de tráfico HTTPS)	443
<input type="radio"/>	Supervisión de máquina virtual (DCOM de entrada)	135
<input type="radio"/>	Supervisión de máquina virtual (RPC)	Puertos diná...
<input type="radio"/>	Supervisión de máquina virtual (Sesión NB de entrada)	139
<input type="radio"/>	Supervisión de máquina virtual (Solicitud de eco - ICMPv4 de entrada)	Cualquiera
<input type="radio"/>	Supervisión de máquina virtual (Solicitud de eco - ICMPv6 de entrada)	Cualquiera
<input type="radio"/>	Uso compartido de archivos e impresoras a través de SMBDirect (iWARP d...	5445

Fuente: elaboración propia.

Habiendo realizado esto, se podrá ingresar al sitio web desde la dirección ip pública que se le asignó al servidor, permitiendo así que el sitio sea visitado desde cualquier parte del mundo, cumpliendo así uno de los objetivos primordiales del proyecto.

Figura 87. **Sitio web público**



Fuente: elaboración propia.

### 4.3. Pruebas al sistema

Habiendo implementado la aplicación en el servidor, es necesario conocer el rendimiento de la aplicación y así también las vulnerabilidades relacionadas con el acceso a la misma.

#### 4.3.1. Pruebas de rendimiento

Para realizar las pruebas de rendimiento, se utilizará una herramienta en línea para examinar el sitio, esta herramienta es llamada “Web Page Performance test”, la cual puede ser ejecutada desde el sitio web <http://www.webpagetest.org/>. Esta prueba consiste en un primer intento de cargar la página y luego cuando esta ya está guardada en caché en su segunda carga, aquí se observan los resultados de dicha prueba:

Tabla VI. **Prueba de rendimiento**

		Primera Carga	Carga recurrente
	Tiempo de Carga	4,250s	0,916s
	Primer Byte	0,316s	0,318s
	Inicio del Render	4,188s	0,990s
	Velocidad de Índice	4 207	1 000
	Elementos	61	61
Documento completo	Tiempo	4,250s	0,916s
	Peticiones	26	1
	Bytes de entrada	1,171 KB	4 KB
Completamente cargado	Tiempo	4,406	0,916s
	Peticiones	26	1
	Bytes de entrada	1,202 KB	86 KB

Fuente: elaboración propia.

Este tiempo fue medido publicando el sitio en un servidor con ancho de banda de 1Mbps, a pesar de esto el tiempo de carga del sitio no es elevado y mejora rotundamente en la carga recurrente (con el sitio en caché), también se puede observar en la carga recurrente que únicamente debe realizar una petición de las 26 peticiones iniciales.

En los anexos 4 y 5 se puede observar la forma de carga del sitio en la línea del tiempo para tener una idea más clara de la forma en la que van cargando los componentes.

### 4.3.2. Escaneo de puertos

En busca de la seguridad del sitio, es necesario verificar que los cambios aplicados en el *firewall* del servidor que lo contiene estén funcionando de manera correcta, para esto, es necesario realizar un escaneo de puertos para verificar las vulnerabilidades del mismo y así asegurarse de cubrirlas para evitar que los ataques a los que pueda ser objeto logren su objetivo.

Para realizar esta prueba desde el exterior, se utilizará una herramienta pública alojada en el sitio web <http://mxtoolbox.com> el cual es reconocido por su fiabilidad en análisis de seguridad y de resolución de nombres.

En el sitio web, se debe buscar el apartado de *Port Scanner* y colocar la dirección ip del sitio de interés, lo cual desplegará los resultados.

La tabla VII describe el estado del puerto y el número. Como se puede observar el sitio web únicamente tiene el puerto 80 (HTTP) abierto, el cual es el único puerto necesario para que la aplicación funcione sin exponer el sistema a ataques externos.

Cabe mencionar que este análisis de puertos realiza una búsqueda entre los puertos que son comúnmente atacados y que representan una vulnerabilidad para el sistema, muestra cuales están abiertos y cuales cerrados para que posteriormente, se analice la necesidad de estos y se mitigue el riesgo.



Tabla VII. Escaneo de puertos del sitio

<b>Estado</b>	<b>Número de puerto</b>	<b>Nombre</b>
Cerrado	21	FTP
Cerrado	22	SSH
Cerrado	23	TELNET
Cerrado	25	SMTP
Cerrado	53	DNS
Abierto	80	HTTP
Cerrado	100	POP3
Cerrado	111	PORT MAPPER
Cerrado	135	MICROSOFT RPC
Cerrado	139	NETBIOS
Cerrado	143	IMAP
Cerrado	389	LDAP
Cerrado	443	HTTPS
Cerrado	445	SMB
Cerrado	587	MSA-OUTLOOK
Cerrado	1025	IIS,NFS,RFS
Cerrado	1352	LOTUS NOTES
Cerrado	1433	SQL SERVER
Cerrado	1723	POINT TO PONTOTUNNELING
Cerrado	3306	MYSQL
Cerrado	3389	REMOTE DESKTOP
Cerrado	5060	SIP
Cerrado	5900	VNC
Cerrado	6001	X WINDOW SERVER
Cerrado	8080	WEBCACHE

Fuente: elaboración propia.



## CONCLUSIONES

1. Es importante presentar los conceptos claves que permiten comprender la naturaleza de un proyecto, ya que esta misma no es de dominio del público en general.
2. Debido a los avances en las tecnologías de desarrollo web y la gran variedad que se presenta actualmente, es de suma importancia que previo al desarrollo de un sistema, sea definido cuales tecnologías son aplicables en el caso de ese sistema en concreto.
3. Para darle validez a la implementación de un nuevo proceso o sistema, es necesario realizar la comparación de lo que se está dejando atrás, contra lo que se está proponiendo, en este caso, el sistema antiguo comparado al nuevo sistema.
4. En pro de beneficiar la administración de un sistema, es de suma importancia documentar el proceso de implementación del mismo, evitando así los vacíos de configuración que puedan presentarse posteriormente.
5. Para demostrar la fiabilidad y el buen rendimiento de un sistema, es necesario someterlo a las pruebas pertinentes para cada caso, asegurando así la estabilidad del mismo.



## RECOMENDACIONES

1. Los administradores del sistema pueden tomar en consideración un plan de mantenimiento para la base de datos y la aplicación, esto en función de conservar el rendimiento de las mismas.
2. A los administradores del sistema programar respaldos mensuales a la base de datos, en función de mantener la integridad de la información y así evitar pérdidas que puedan ser generadas por alguna falla del sistema operativo o del disco duro.
3. Es un punto a considerar la instalación de actualizaciones y parches que requiera el sistema operativo, esto en pro de mantener la estabilidad y seguridad del sistema y de la información.
4. Puede llegar a ser de utilidad, que este sistema pueda ser accedido desde la página de la Facultad de Ingeniería, para que sea publicada de forma integrada y brindando un fácil acceso a todos los usuarios.
5. Al realizar una actualización o migración de sistema, es recomendable conocer los flujos de proceso a los que se apega el sistema anterior, para así rediseñar los flujos haciéndolos más eficientes, generando una plataforma más sólida sobre la cual se desarrollara el nuevo sistema.



## BIBLIOGRAFÍA

1. ÁLVAREZ, Miguel Angel. *Que es JSP* [en línea]. julio de 2002. <[http://www. Desarrollo web.com/ articulos/ 831.php](http://www.Desarrollo web.com/ articulos/ 831.php)>
2. CALDAS SÁNCHEZ, Rolando. *Media queries como base del responsive design* [en línea]. <<http://rolandocaldas.com/html5/media-queries-y-responsive-design>, > [Consulta: 13 de enero de 2015].
3. CAMPS PARÉ, Rafael; CASILLAS SANTILLÁN, Luis Alberto; COSTA COSTA, Dolors, *et al. Software libre, bases de datos*. Fundació per a la Universitat Oberta de Catalunya. Barcelona: Eureka Media SL, 2005. 460 p. ISBN: 84-9788-269-5.
4. *Diseño básico de páginas web en HTML*. Málaga: Vértice, 2008. 162 p. ISBN: 978-84-92533-11-4.
5. GARCÍA GARCÍA, Ángel Luis. *Introduccion a Python*. España: Autoedición, 2011. 88 p.
6. GUTIÉRREZ, Javier J. *¿Qué es un framework web* [en línea]. <[http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf)> [Consulta: 8 de enero de 2014].
7. GRUPO RUBY. *Programación en Rubry*. [en línea]. Wiki Libros, Septiembre 2009 <[http://es.wikibooks.org/wiki/Programaci %C3 %B3n\\_en\\_Ruby](http://es.wikibooks.org/wiki/Programaci%C3%B3n_en_Ruby)>. [Consulta: 4 de febrero de 2014].

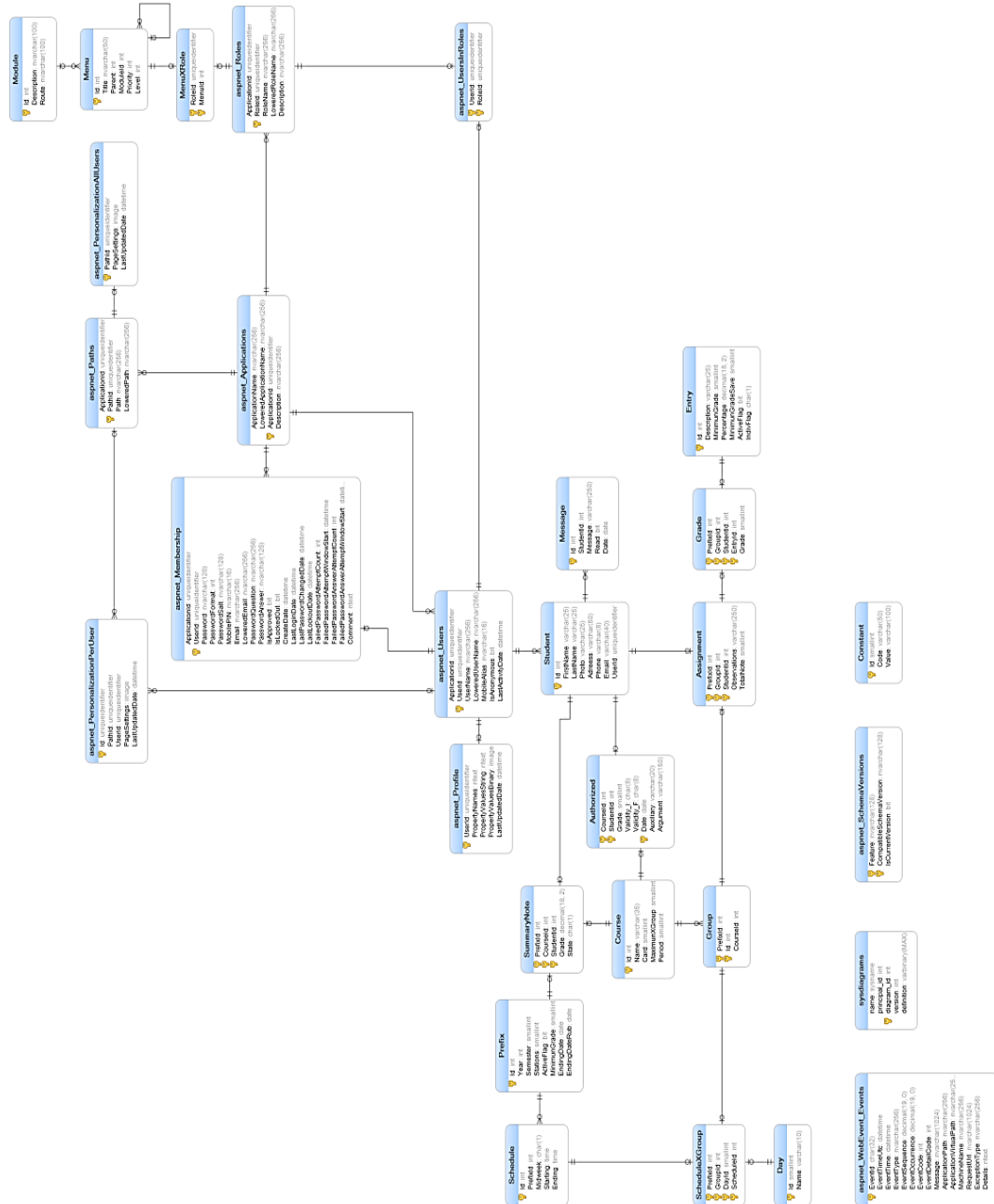
8. HERNÁNDEZ RODRÍGUEZ, Jesús Ignacio. *Análisis y desarrollo web*. México: 2014. 346 p.
9. LUJAN MORA, Sergio. *Programación de aplicaciones web: historia, principios básicos y clientes web*. España: Editorial Club Universitario, 2002. 349 p. ISBN: 84-8454-206-8.
10. MACIÁ PÉREZ, Francisco; MORA GIMENO, F. J; GIL MARTÍNEZ-ABARCA, J. A., et al. *Administración de servicios de internet: de la teoría a la práctica*. Alicante: Publicaciones Universidad de Alicante, 2008. 350 p. ISBN: 978-84-7908-989-4.
11. MARTINEZ, Rafael; FERNÁNDEZ, Víctor; BOSHELL, Leonardo, et al. *Manual de PHP*. México: Grupo de documentación de PHP, 2002. 112 p.
12. SÁNCHEZ, Jorge. *Java2. Basado en el lenguaje Java definido por Sun* [en línea] 2004 <Http://java.sun.com> [Consulta: 4 de enero de 2015]
13. SANCHEZ ASENJO, Jorge. *Servidores de aplicaciones WEB*. [en línea] Creativecommons, 2011. <http://issuu.com/daniel\_villa\_nueva\_montoya/docs/servidores\_de\_aplicaciones\_web\_-\_jo> [Consulta: 10 de enero de 2015].





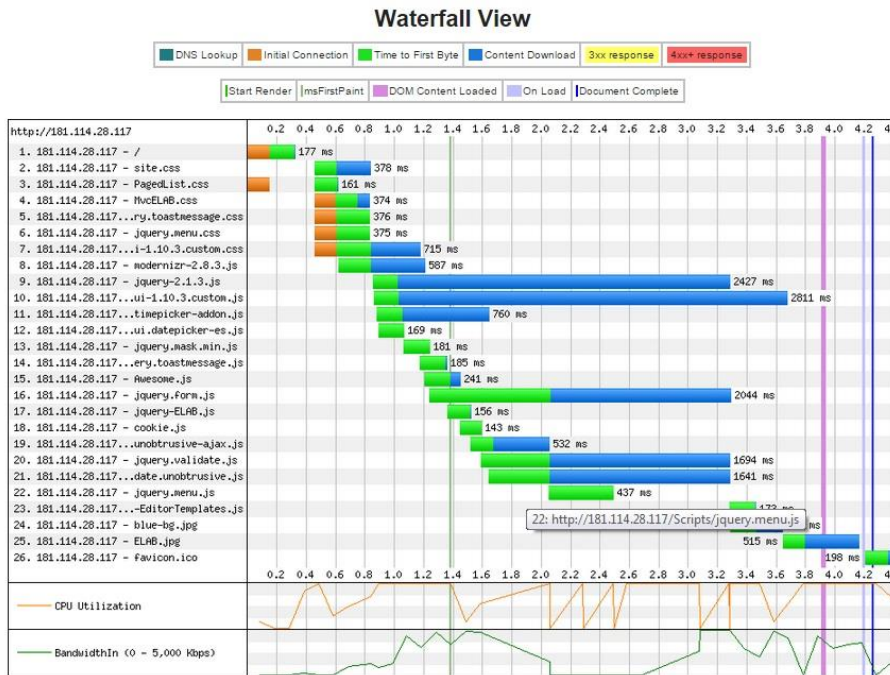


### Anexo 3 Diagrama entidad relación en el nuevo sistema



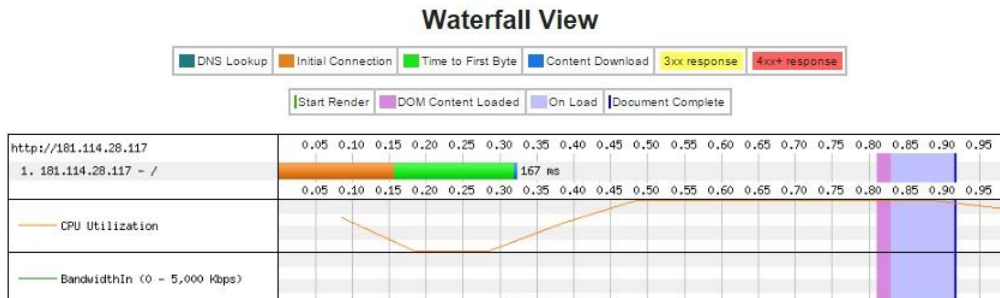
Fuente: Diagrama entidad del nuevo sistema.

## Anexo 4 Vista de cascada en la primera carga del sitio web



Fuente: Vista de cascada en la primera carga del sitio web.

## Anexo 5 Vista de cascada en la carga recurrente del sitio web



Fuente: Vista de cascada en la carga recurrente del sitio web.