



INSTITUTO TECNOLÓGICO SUPERIOR DE MISANTLA

JERARQUÍA DE CLASES PARA
LA CREACIÓN DE UN
FRAMEWORK RESPONSIVO
MEDIANTE ECMAScript 6

MÓDULO: JEFES DE CARRERA DEL SITM

TESIS

QUE PARA OBTENER EL GRADO DE:
Maestra en Sistemas Computacionales

Presenta:

Carmen Juliana Aguilar Fernández

Director:

Dr. Jorge Mario Figueroa García

Misantla, Veracruz Septiembre de 2018



**INSTITUTO TECNOLÓGICO SUPERIOR DE MISANTLA
DIVISIÓN DE ESTUDIOS PROFESIONALES
AUTORIZACIÓN DE IMPRESIÓN DE TRABAJO DE TITULACIÓN MAESTRÍA**

FECHA: 20 de Septiembre de 2018.

ASUNTO: **AUTORIZACIÓN DE IMPRESIÓN
DE TESIS.**

A QUIEN CORRESPONDA:

Por medio de la presente se hace constar que el (la) C:

CARMEN JULIANA AGUILAR FERNÁNDEZ

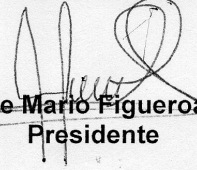
estudiante de la maestría en SISTEMAS COMPUTACIONALES con No. de Control 162T0074 ha cumplido satisfactoriamente con lo estipulado por el Lineamiento de Posgrado para la obtención del grado de Maestría mediante Tesis.

Por tal motivo se **Autoriza** la impresión del **Tema** titulado:

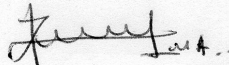
**JERARQUÍA DE CLASES PARA LA CREACIÓN DE UN FRAMEWORK
RESPONSIVO MEDIANTE ECMAScript 6 MÓDULO: JEFES DE CARRERA
DEL SITM**

Dándose un plazo no mayor de un mes de la expedición de la presente a la solicitud del examen para la obtención del grado de maestría.

ATENTAMENTE


DR. Jorge Mario Figueroa García
Presidente




M.I.A. Roberto Ángel Meléndez Armenta
Secretario


M.S.C. Galdino Martínez Flores
Vocal

Archivo.

Agradecimientos

El proyecto SITM es un proyecto con una magnitud superior a un software tradicional, por ello y gracias a la perspectiva del Dr. Jorge Mario Figueroa García, es que este proyecto tomó la forma que actualmente tiene y en la cual mi compañero Maestro en Sistemas Computacionales: Francisco Adolfo Aguilar Gómez y yo, culminamos satisfactoriamente.

Agradezco al Consejo de Ciencia y Tecnología (CONACYT) por su apoyo para la elaboración de este proyecto de investigación.

Número de Becario:

60733

Número de CVU:

786738

Número de apoyo:

447745

Dedicatoria

Gracias Dios por darme la vida, la salud y la fuerza necesaria para culminar esta meta personal.

La realización de esta Tesis, sin lugar de dudas está dedicada a más de una persona, que todos siempre me han apoyado y motivado para seguir adelante. Gracias a mi familia, mis papás y mis hermanas por su apoyo incondicional, lo mejor que tengo en la vida.

Madrina Clara, Gaby y Danielito su risas y compañía me llenan de mucha alegría y fortaleza
Madrina Chela, gracias porque aun estando lejos físicamente de la familia siempre estás al pendiente de nosotros.

Tía Mica, gracias por escucharme siempre que lo necesite.

Tia Irma, gracias por tus ánimos y palabras de aliento.

Mis compañeros de clase y maestría: Gracias por los momentos compartidos, los mejores cómplices que me pude haber encontrado.

Resumen

El Sistema Integral del Tecnológico de Misantla (SITM) es un ERP (Enterprise Resource Planning) que se está desarrollando para integrar y automatizar los procesos docentes y administrativos primordiales sobre una plataforma web para el Instituto Tecnológico Superior de Misantla. En su desarrollo se usan los principios de Programación Orientada a Objetos (POO) con la finalidad de implementar la reutilización de componentes de software en los códigos que generan la interfaz y en la transferencia de información entre los clientes y el servidor. Se trabaja bajo el análisis y diseño del modelo en espiral, debido a la adaptación gradual que este proporciona y como patrón de diseño se tomó como referencia el Modelo Vista Controlador.

El sistema está dividido en un subsistema BackEnd (ya concluido, probado e implantado en otros sistemas), que lleva a cabo una representación orientada a objetos de las bases de datos y sus elementos, logrando con ello una gestión arbitrada de un cluster de base de datos heterogéneas de forma relativamente simple. El segundo subsistema es un FrontEnd compuesto por clientes web asíncronos y adaptativos que extienden una o más clases ECMAScript6 que sirven de plantilla y dotan a sus extensiones de las capacidades necesarias para interactuar con el BackEnd.

Se obtuvo un modelo de comunicación con el servidor muy simple, transferencia de códigos y datos reducida al mínimo, una eficiente renderización de los componentes de interfaz, y de forma inherente se ha elevado la seguridad del sistema.

Actualmente ambos subsistemas se encuentran operando satisfactoriamente, por el lado de la FrontEnd, no se han encontrado vulnerabilidades de inyección de código SQL, además la plantilla opera según la modificación realizada al Modelo Vista-Controlador habiéndose probado en diferentes navegadores y funcionando también de manera adecuada el diseño responsivo en diferentes dispositivos Móviles.

Abstract

The Integral System of the Technological of Misantla (SITM) is an ERP (Enterprise Resource Planning) that is being developed to integrate and automate the primary educational and administrative processes on a web platform for the Higher Technological Institute of Misantla. In its development the principles of Object Oriented Programming (OOP) are used with the purpose of implementing the reuse of software components in the codes that generate the interface and in the transfer of information between the clients and the server. Work is carried out under the analysis and design of the spiral model, due to the gradual adaptation it provides and as a design pattern the Controlled View Model was taken as reference.

The system is divided into a BackEnd subsystem (already concluded, tested and implemented in other systems), which carries out an object-oriented representation of the databases and their elements, thus achieving an arbitrated management of a base cluster of data. heterogeneous data in a relatively simple way. The second subsystem is a FrontEnd composed of asynchronous and adaptive web clients that extend one or more ECMAScript6 classes that serve as templates and endow their extensions with the necessary capabilities to interact with the BackEnd.

A communication model with the very simple server was obtained, code and data transfer was reduced to a minimum, efficient rendering of the interface components, and inherently the security of the system has been increased.

Currently both subsystems are operating satisfactorily, on the FrontEnd side, no SQL code injection vulnerabilities have been found, and the template operates according to the modification made to the Vista-Controller model, having been tested in different browsers and also working properly Responsive design in different Mobile devices.

Índice general

| | |
|---|-----------|
| Autorización de impresión | III |
| Agradecimientos | III |
| Dedicatoria | IV |
| Resumen | V |
| Abstract | VI |
| Índice de figuras | IX |
| Índice de tablas | XI |
| 1. Generalidades | 12 |
| 1.1. Introducción | 12 |
| 1.2. Planteamiento del Problema | 12 |
| 1.2.1. Justificación | 13 |
| 1.3. Objetivos | 14 |
| 1.3.1. Objetivo general | 14 |
| 1.3.2. Objetivos específicos | 14 |
| 1.4. Hipótesis | 14 |
| 1.5. Alcances y limitaciones | 14 |
| 1.5.1. Alcances | 14 |
| 1.5.2. Limitaciones | 15 |
| 1.6. Estructura de la Tesis | 15 |
| 2. Marco teórico | 16 |
| 2.1. Patrón de Fase | 16 |
| 2.1.1. Espiral | 16 |
| 2.2. FrontEnd | 17 |
| 2.2.1. MVC | 17 |
| 2.2.2. Navegador | 19 |

| | | |
|-----------|---|-----------|
| 2.3. | Interfaz de Comunicación | 20 |
| 2.3.1. | JSON | 20 |
| 2.4. | BackEnd | 20 |
| 2.4.1. | Base de Datos | 20 |
| 3. | Estado del arte | 23 |
| 4. | Desarrollo de la solución | 26 |
| 4.1. | Herramientas | 26 |
| 4.2. | El enfoque del sistema | 26 |
| 4.3. | Diseño del FrontEnd | 26 |
| 4.3.1. | Vista | 27 |
| 4.3.2. | Controlador | 29 |
| 4.3.3. | Interfaz de comunicación | 29 |
| 4.4. | Diseño del BackEnd | 29 |
| 4.4.1. | Clases para describir el modelo de datos | 29 |
| 4.4.2. | Intercomunicación con los clientes web | 31 |
| 4.5. | Diagramas de Flujo | 31 |
| 4.6. | Diagramas de Clases | 31 |
| 4.7. | Diagrama de Secuencia | 32 |
| 5. | Análisis de resultados | 41 |
| 5.1. | Pruebas de Verificación y Validación | 41 |
| 5.1.1. | Líneas de Código: Librería mensajes.js | 41 |
| 5.1.2. | Análisis de Seguridad Web | 42 |
| 5.2. | Compatibilidad entre navegadores | 42 |
| 5.3. | Diseño Responsivo en diferentes tamaños de Pantalla | 42 |
| 5.3.1. | Validación de la Arquitectura | 43 |
| 6. | Conclusiones y Trabajos futuros | 47 |
| 6.1. | Conclusiones | 47 |
| 6.2. | Trabajo futuro | 47 |
| | Referencias | 48 |
| | Anexos | 50 |
| 6.3. | Encuesta sobre Navegadores más utilizados | 50 |
| 6.4. | Código HTML | 50 |

Índice de figuras

| | |
|--|----|
| 2.1. Variante del modelo en espiral | 17 |
| 2.2. Modelo Vista Controlador tradicional. | 18 |
| 4.1. Arquitectura MVC adaptada al FrontEnd Fuente: Propia del Proyecto . . | 27 |
| 4.2. Imagen muestra la manera en la que se consume cada una de las clases para implementar tabla, registro y campo. | 28 |
| 4.3. Imagen muestra la manera en la que se consume cada una de las clases para implementar bloques, materias y materia. | 28 |
| 4.4. Modelo de factorización Down-Top aplicado en la implementación del BackEnd | 30 |
| 4.5. Comunicación de los diferentes cliente con el Servidor mediante peticiones Ajax. | 32 |
| 4.6. Diagrama de Flujo de carga de la Pantalla para el módulo de Jefes de Carrera. | 33 |
| 4.7. Diagrama de Flujo de la Opción Carga Académica. | 34 |
| 4.8. Diagrama de Clases FrontEnd del módulo Jefes de carrera. | 35 |
| 4.9. Diagrama de Clases de los descriptores del Backend. | 36 |
| 4.10. Diagrama de Secuencia para la función de Carga Académica. | 37 |
| 4.11. Diagrama de Secuencia para la opción de Especialidades dentro de la función de Configuración | 38 |
| 4.12. Diagrama de Secuencia para la opción de Administrar Grupos la función de Configuración | 39 |
| 4.13. Diagrama de Secuencia para la opción de Permisos de Inscripción dentro de la función de Configuración | 40 |
| 5.1. Imagen del módulo Jefes de Carrera con el navegador Chrome. | 43 |
| 5.2. Imagen del módulo Jefes de Carrera con el navegador Edge | 44 |
| 5.3. Imagen del módulo Jefes de Carrera con el navegador Firefox. | 44 |
| 5.4. Imagen del módulo Jefes de Carrera, opción carga académica. | 45 |
| 5.5. Imagen del módulo Jefes de Carrera, opción configuración. | 45 |
| 5.6. Cronograma de actividades módulo: Adeudos. | 46 |
| 6.1. Resultado de JavaScript Orientado a Objetos. | 54 |
| 6.2. Plan de Estudios Ing. Ambiental. | 55 |
| 6.3. Plan de Estudios Ing. Bioquímica. | 56 |
| 6.4. Plan de Estudios Ing. Civil. | 57 |
| 6.5. Plan de Estudios Ing. Electromecánica. | 58 |

6.6. Plan de Estudios Ing. en Gestión Empresarial. 59
6.7. Plan de Estudios Ing. Industrial. 60
6.8. Plan de Estudios Ing. Petrolera. 61
6.9. Plan de Estudios Ing. en Sistemas Computacionales. 62
6.10. Plan de Estudios Ing. en Tecnologías de la Información y Comunicaciones. . 63

Índice de tablas

| | |
|---|----|
| 5.1. Comparativa de la librería mensajes, alertify y bootstrap.dialog en Javascript | 41 |
| 5.2. Comparativa entre entre SITM e implementaciones del SII | 42 |
| 6.1. Uso de exploradores según la w3School para el mes de Mayo de 2018 | 50 |

Capítulo 1

Generalidades

1.1. Introducción

El Sistema Integral del Tecnológico Superior de Misantla (SITM) surge de la necesidad de adecuar a las condiciones actuales los sistemas informáticos que hoy intervienen en los procesos administrativos y académicos del Tecnológico de Misantla y por la necesidad de sistematizar aquellos que aún no lo están. Mediante el SITM, se interconectan los diversos departamentos y/o funciones que intervienen en los procesos de gestión y docencia de la institución y se integra software de terceros y a terceras entidades; se dota así de la interconectividad, posibilidades de expansión y eficiencia que los sistemas actuales de la institución no poseen. El sistema está dividido en dos subsistemas: Un Backend y un FrontEnd. El Backend es un subsistema totalmente autónomo, compuesto por un agente listener extensible y configurable junto con un conjunto de clases que permiten llevar a cabo una representación orientada a objetos de las bases de datos y sus elementos, con la capacidad de interactuar con uno o más cluster de bases de datos heterogéneas [1] que pueden o no estar localizadas en un mismo lugar. El FrontEnd es un subsistema compuesto por un grupo de clientes web asíncronos y adaptativos que extienden una o más clases ECMAScript6 (European Computer Manufacturers' Association) (International, 2018) que sirven de plantilla y dotan a sus extensiones de las capacidades necesarias para interactuar con el listener del Backend y generar dinámicamente los elementos que componen la interfaz de usuario. Como resultado se ha obtenido un modelo de comunicación con el servidor muy simple, transferencia de códigos y datos reducida al mínimo, una eficiente renderización de los componentes de interfaz, y de forma inherente se ha elevado la seguridad del sistema.

1.2. Planteamiento del Problema

En el Instituto Tecnológico Superior de Misantla (ITSM) se cuenta actualmente con un sistema monolítico poco flexible para la gestión de los servicios escolares y jefaturas de carrera, el cual es considerado desfasado para las necesidades actuales que tiene la institución,

carente además de capacidades de interconectividad entre sistemas e imposible de modificar para agregar nuevos módulos. Por estrategia, el ITSM apuesta hacia el uso y explotación de las Tecnologías de la Información y las Comunicaciones (TICs) en la gestión e impartición de carreras que en su interior se brindan. Tras un análisis de la parte directiva con la parte operativa se concluyó que: es necesario contar con un sistema de gestión que integre la parte académica, administrativa, operativa y directiva en sus diferentes niveles y que sea capaz de interconectar con otros sistemas heterogéneos.

Un ejemplo de los problemas con el sistema actual es en la reinscripción de un alumno. El departamento de servicios escolares debe cerciorarse que el solicitante no tenga adeudos. Los adeudos pueden ser un pago, un libro o una actividad requerida por el plan de estudios, todos estos adeudos son registrados en otra base de datos sin conexión a la de servicios escolares, así que si el alumno por ejemplo, realiza un pago en caja pero servicios escolares no registra este pago en su propia base de datos, el joven queda como deudor cuando realmente él ya ha pagado su cuenta en caja. Para registrar los adeudos, servicios escolares actualizan su base de datos cada semestre de forma manual a partir de una lista impresa, la cual fue extraída de la base de datos en otro departamento, esta es una actividad extra para servicios escolares, además hace necesario rectificar muchas veces su propia información y este proceso se realiza por cada alumno inscrito.

Otro ejemplo es la asignación de horarios de los docentes. Los jefes de carrera fijan los horarios de cada grupo, si se trata de un grupo regular el horario se asigna con sucesión ininterrumpida de horas, pero hay casos en que el grupo no es regular y hace necesario abrir un curso que está marcado fuera de la planeación escolar, estos casos especiales son muy recurrentes y hacen estragos en las actividades haciendo más difícil crear una carga académica por grupo. Pero no solo el problema termina en el grupo, también hay alumnos que no son regulares y los cuales deben armar de forma individual sus horarios sin negarles los cursos marcados por el plan de estudios con los pocos docentes compartidos entre carreras en la institución. Estos y muchos más son los problemas que el jefe de carrera debe de resolver al crear los horarios.

1.2.1. Justificación

Debido al incremento de alumnado que ha tenido el ITSM desde su fundación, los recursos administrativos y docentes han crecido en consecuencia, siendo necesario crear sistemas heterogéneos que satisfagan las necesidades principales, mismos que han carecido de capacidades de interconectividad. Se pretende beneficiar principalmente a la Comunidad: Docente, Estudiantil, Directiva y Administrativa de la Institución brindándoles un servicio más ágil y unificado que el que actualmente se tiene. En segundo término también serán beneficiados los aspirantes a alumnos y los Padres de Familia. La aportación al área de la computación es que esta plantilla debido a que se siguieron los lineamientos del Tecnológico Nacional de

Mexico, puede ser utilizada por cualquier otro campus dependiente de ella. Además se requiere una base de datos normalizada para integrarla al BackEnd del sistema para la gestión de un cluster heterogeneo para la misma.

1.3. Objetivos

1.3.1. Objetivo general

Diseñar e implementar una jerarquía de clases en ECMAScript 6 para reducir al mínimo los códigos generados mediante componentes reutilizables y adaptativos.

1.3.2. Objetivos específicos

- Analizar, diseñar e implementar clases para la Plantilla que utilizarán los módulos que así lo requieran para la reutilización de código.
- Adaptar las funciones y características del módulo de Jefes de Carrera al Backend para representar la información orientada a objetos de la Base de Datos.
- Analizar, diseñar e implementar la función de Carga Académica para que los Jefes de Carrera puedan ofertar las materias de cada semestre.
- Analizar, diseñar e implementar la función de Alumnos para que el Jefes de carrera pueda acceder a los datos de Asistencia y Kardex de los alumnos.
- Analizar, diseñar e implementar la función de Configuración para que el Jefe de carrera pueda acceder a Definir Especialidades, Administrar grupos y Permisos de Inscripción.

1.4. Hipótesis

El diseño de un framework orientado a objetos en ECMAScript 6 disminuirá la transferencia de información entre el cliente y el servidor, además facilitará la elaboración de módulos en periodos cortos de tiempo, permitiendo inherentemente mantenimientos y actualizaciones más rápidas.

1.5. Alcances y limitaciones

1.5.1. Alcances

Para el módulo de Jefes de Carrera, se tiene contemplado incluir las funciones de:

- Carga Académica

- Tutorías
- Docentes
 - Consulta de Asistencia
 - Planeaciones
- Alumnos
 - Consulta de Asistencia
 - Consulta de Kardex
- Configuración
 - Definir Especialidades
 - Administrar Grupos
 - Permisos de Inscripción

1.5.2. Limitaciones

Para la generación de horarios de la carga académica de alumnos y docentes es necesario que las retículas de todas las carreras que se ofertan en el ITSM se encuentren completas, es decir que se conozca que materias serán impartidas en el transcurso de los semestres. La carrera de Ingeniería Petrolera al ser una carrera de reciente creación no cuenta con la información completa de las materias de especialidad que impartirá.

1.6. Estructura de la Tesis

En el capítulo 2: Marco Teórico, se presentan los conceptos que existen sobre el tema a investigar, se explica el patrón de Fase usado para el diseño de software que es cascada, se definen también los conceptos principales en los que se dividió el proyecto, el FrontEnd, el BackEnd y la Interfaz de comunicación entre ellos.

En el capítulo 3: Estado del arte, se muestra una recopilación de los autores e instituciones que en Artículos y Tesis han abordado el tema que es objeto de estudio, se realiza un análisis de los artículos mencionados donde se resume el conocimiento y las conclusiones relacionadas con el área de estudio.

En el capítulo 4: Desarrollo de la solución, primeramente se indican las herramientas utilizadas y las premisas necesarias para el desarrollo, se profundiza la manera en la trabaja el FrontEnd y la manera en la que se comunica con el Backend así como el diseño del mismo. Se incluyen al final los diagramas de Flujo, Clases y Secuencia.

En el capítulo 5: Análisis de Resultados, se indican las pruebas de verificación y validación realizadas, se muestran también como el sistema funciona en los navegadores seleccionados y además como el diseño responsivo se visualiza en diferentes dispositivos.

Capítulo 2

Marco teórico

2.1. Patrón de Fase

El patrón de fase define la secuencia de las actividades estructurales que ocurren dentro del proceso, aun cuando el flujo general de las actividades sea de naturaleza iterativa. La literatura señala 3 patrones de fase principalmente: Cascada, Basada en Componentes y Espiral, esta última descrita a continuación:

2.1.1. Espiral

Propuesto en primer lugar por Barry Boehm, el modelo espiral es un modelo evolutivo del proceso del software y se acopla con la naturaleza iterativa de hacer prototipos con los aspectos controlados y sistémicos del modelo de cascada. Tiene el potencial para hacer un desarrollo rápido de versiones cada vez más completas. Boehm describe el modelo del modo siguiente:

El modelo de desarrollo espiral es un generador de modelo de proceso impulsado por el riesgo, que se usa para guiar la ingeniería concurrente con participantes múltiples de sistemas intensivos en software. Tiene dos características distintivas principales. La primera es el enfoque cíclico para el crecimiento incremental del grado de definición de un sistema y su implementación, mientras que disminuye su grado de riesgo. La otra es un conjunto de puntos de referencia de anclaje puntual para asegurar el compromiso del participante con soluciones factibles y mutuamente satisfactorias.

Con el empleo del modelo espiral, el software se desarrolla en una serie de entregas evolutivas. Durante las primeras iteraciones, lo que se entrega puede ser un modelo o prototipo. En las iteraciones posteriores se producen versiones cada vez más completas del sistema cuya ingeniería se está haciendo (Pressman y Troya, 1988). En la Figura 2.1 se muestra el modelo espiral definido.

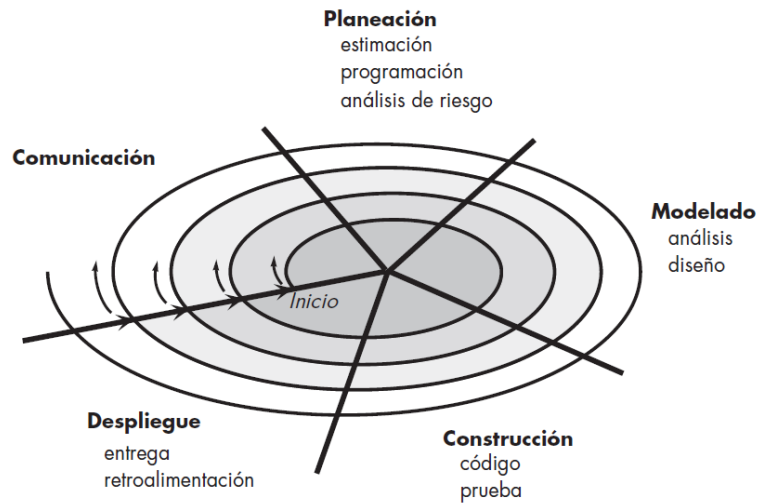


Figura 2.1: Variante del modelo en espiral (Pressman, 2010)

2.2. FrontEnd

El Front-End básicamente es todo lo relacionado con lo que el usuario ve, incluido el diseño y algunos lenguajes como HTML, CCS y Javascript.

2.2.1. MVC

El Model View Controller (modelo-vista-controlador) o MVC es un patrón de diseño que permite separar en capas nuestra aplicación para lograr un menor acoplamiento entre el código. Es una solución que sirve en todo tipo de aplicaciones no sólo las de tipo ricas en internet. El patrón de diseño sugiere que se divida nuestra aplicación en 3 Capas, y dentro de una aplicación AJAX una metodología de aplicación del patrón MVC(Figura 2.2) es separar nuestro desarrollo en:

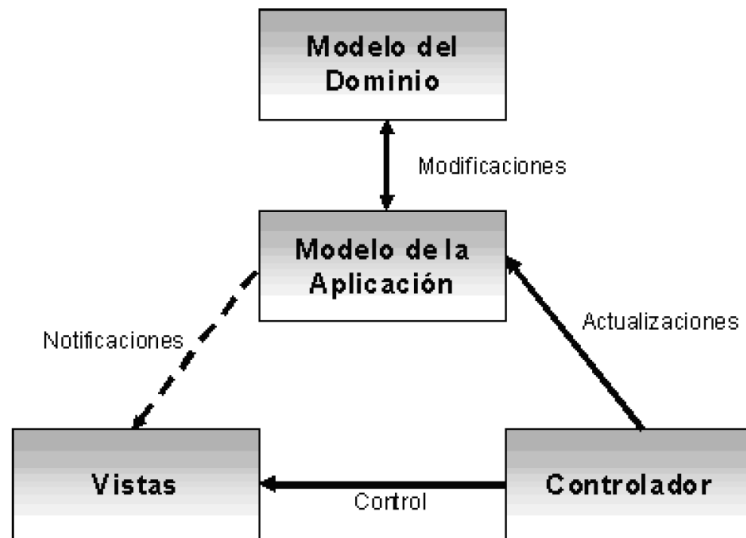


Figura 2.2: Modelo Vista Controlador tradicional.
(Pantoja, 2004)

Vista

Es la encargada de mostrar información al usuario y recibir su interacción, por lo que es el archivo XHTML junto al CSS que define la estructura de los elementos que el usuario ve junto con un identificador (id) a cada elemento con el que se quiere interactuar. El XHTML no contendrá código JavaScript alguno, ni siquiera la definición de un onClick.

HTML Hoja de estilo en cascada (CSS): La hoja de estilo en cascada o CSS (siglas en inglés de cascading style sheets) es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML y derivados. El W3C es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

La información de estilo se define en un documento separado, anteriormente se manejaba o en el mismo documento HTML, pero esto genera malas prácticas. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante las etiquetas «style»«/style» (W3C, 2015).

Diseño web adaptativo: Se aplican los principios de la web adaptativa, existen tres elementos fundamentales para lograr un diseño web adaptativo, que son:

- Cuadrícula fluida: Este concepto lo que propone es usar porcentajes para definir los tamaños de las columnas o divs, en lugar de píxeles.
- Imágenes flexibles: Las imágenes no tienen anchos fijos sino un máximo (max-width),

que en una computadora de escritorio o laptop suele mostrarse al 100 % y se ajustara al tamaño de su contenedor si este reduce su tamaño.

- Media queries: Con el uso de estilos CSS permite que se personalice utilizando un ancho mínimo y máximo del navegador (min-max width) (Marcotte, 2011).

Controlador

Es el que recibe el aviso de la interacción del usuario y decide qué es lo que hay que hacer. No se encarga de hacerlo, dado que para eso invoca al Modelo, es un archivo JavaScript el que controla ala Vista mencionada antes, por lo general con el mismo nombre que el archivo XHTML, que al cargarse inicializará todo el comportamiento inicial de la aplicación y se descargará de administrar la interacción entre la vista y el modelo.

Modelo

Son distintos archivos JavaScript invocados desde el controlador. Son los que mantienen la lógica del negocio, los que se comunican con el servidor cuando sea necesario y los que le avisan al controlador de los cambios en el estado (Firtman, 2008).

2.2.2. Navegador

En una arquitectura de tipo cliente-servidor, el usuario interactúa y obtiene información desde su computadora a través de una aplicación cliente. En la Web estas aplicaciones se conocen bajo el nombre genérico de "browsers" (también llamadas en nuestro idioma "visores", "visualizadores", "navegadores" ó "exploradores"), y cumplen dos funciones básicas:

1. Transmitir a los servidores remotos las órdenes que le imparte el usuario.
2. Presentar la información en forma asequible a quien la solicite (Valzacchi, 2003).

Según Net MarketShare que incluye datos sobre la distribución y el reparto del mercado de navegadores, sistemas operativos y buscadores en ordenadores personales, tabletas y teléfonos móviles, los tres principales navegadores usados en el año 2017 fueron: Chrome con 58.90 %, Firefox con 13.29 % e Internet Explorer con 13.00 (NetMarketShare, 2018)

ECMAScript 6: Es un lenguaje de programación orientado a objetos para realizar cálculos y manipular objetos computacionales dentro de un entorno host. ECMAScript como se define aquí no pretende ser computacionalmente autosuficiente; de hecho, no hay disposiciones en esta especificación para la entrada de datos externos o salida de resultados calculados. En su lugar, se espera que el entorno computacional de un programa ECMAScript proporcione no solo los objetos y otras instalaciones descritas en esta especificación sino también ciertos objetos específicos del entorno, cuya descripción y comportamiento están más allá del alcance de esta especificación excepto para indicar que puede proporcionar ciertas propiedades a las que se puede acceder y ciertas funciones que se pueden invocar desde un programa

ECMAScript (International, 2015b).

2.3. Interfaz de Comunicación

Los formatos para la transferencia de datos entre el FrontEnd y el BackEnd más comunes y ligeros: XML y JSON, siendo este último el que se elige por ser mucho más práctico de manejar.

2.3.1. JSON

JSON: (JavaScript Object Notation - Notación de Objetos de JavaScript es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos. JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras (Json.org, 2013).

2.4. BackEnd

Frecuentemente un programador comienza creando una estructura de datos en un Lenguaje estructurado de consultas(SQL), y a continuación escribe un programa en algún otro lenguaje (como PHP) para dar acceso a estos datos (Harris, 2009).

2.4.1. Base de Datos

Base de datos: Cada día, la mayoría de nosotros nos encontramos con actividades que requieren algún tipo de interacción con una base de datos (ingreso en un banco, reserva de

una entrada para el teatro, solicitud de una suscripción a una revista, compra de productos, etc.) Estas interacciones son ejemplos de lo que se llama aplicaciones tradicionales de bases de datos (básicamente información numérica o de texto), aunque los avances tecnológicos han permitido que también existan: bases de datos multimedia, sistemas de información geográfica (GIS), almacenes de datos, sistemas de proceso analítico on-line, . . .

- Una base de datos se entenderá como una colección de datos relacionados entre sí y que tienen un significado implícito.
- Por datos queremos decir hechos conocidos que pueden registrarse y que tienen un significado implícito.

Para manipular y gestionar las bases de datos surgieron herramientas software denominadas: sistemas gestores de bases de datos. (Guevara, 2018).

Diagrama E-R

Los diagramas son simples y claros, cualidades que pueden ser responsables del amplio uso del modelo E-R. Tal diagrama consta de los siguientes componentes principales: Rectángulos, que representan conjuntos de entidades.

Elipses, que representan atributos.

Rombos, que representan relaciones.

Líneas, que unen atributos a conjuntos de entidades y conjuntos de entidades a conjuntos de relaciones.

Elipses dobles, que representan atributos multivalorados.

Elipses discontinuas, que denotan atributos derivados.

Líneas dobles, que indican participación total de una entidad en un conjunto de relaciones.

Rectángulos dobles, que representan conjuntos de entidades débiles (Korth y Silberschatz, 1993).

Normalización

La idea principal de la Normalización es la de separar nuestros datos o información en un serie de tablas que tengan relación entre sí. Cuando una base de datos es diseñada de manera correcta se evitan problemas con datos inconsistentes, problemas con los datos de operación y/o problemas con los campos listados. Para que una tabla de una base de datos se considere normalizada debe estar al menos en la tercera forma normal.

Primera Forma Normal (Eliminar Campos Listados): Una tabla está en la primera forma normal si y solo si representa una relación, no permite nulos o filas. El objetivo de esta forma normal eliminar repeticiones en la base de datos.

Segunda Forma Normal (Eliminar Redundancias): Para que una tabla esté dentro de la

segunda forma normal, deberá estar en la 1FN y además cada uno de sus atributos no principales deberán de ser completamente dependientes de solo una llave primaria o candidata. Tercera Forma Normal (Asegura la dependencia Funcional) : Una tabla estará dentro de la 3FN si está dentro de la segunda forma Normal además deberá tener una sola llave primaria en cada campo.(Elmasri, 2008).

Capítulo 3

Estado del arte

Las investigaciones mencionados en esta sección, son una recopilación de resultados que se toman en consideración por su alta relación con el tema seleccionado, se han elegido artículos, tesis y un prototipo. Se ha buscado que las investigaciones relacionadas no pasen de 6 años de antigüedad, pues el área de Tecnologías de la Información y Comunicaciones constantemente ofrece nuevas mejoras.

Lin & Chen muestran através de la aplicación en proyectos reales como AJAX trabaja con tipos de datos Diferentes, XML(mas robusto) y JSON(un poco más ligero), se realizan dos comparaciones. La primera consiste en comparar la velocidad de transmisión entre el cliente y el servidor usando cantidades de datos entre 100 y 2000, dando menores tiempos los Datos JSON. La segunda comparación es del lado del cliente, muestra como AJAX trabaja con los dos formatos de datos(con cantidades de datos entre 100 y 500), dando menor tiempo de respuesta y mayor estabilidad con JSON. Los autores consideran que AJAX-JSON y su correspondiente variedad de kit de desarrollo y sus características de sintaxis concisas y legibles, proporcionan una gran comodidad para los desarrolladores de aplicaciones. (Lin, Chen, Chen, y Yu, 2012).

Dhoti, & Sarate realizan un análisis de complejidad de pruebas de rendimiento en aplicaciones web basadas en Ajax, indican que las pruebas hechas comúnmente a aplicaciones web tradicionales deben ser modificadas y mejoradas para entornos basados en AJAX, proponen en primer lugar definir metas de rendimiento y métricas relevantes, como utilización de la red, número de cálculos en el cliente o servidor y la capacidad de respuesta de la aplicación, así también se deberá probar la matriz de compatibilidad con los navegadores, sistemas operativos, hardware del cliente, topología de red y velocidades de red. Proponen pruebas de blackbox, que simulen las acciones de los usuarios en los navegadores. La atención deberá centrarse en la estabilidad y los tiempos de respuesta. Los autores concluyen que para tener éxito, se debe identificar los desafíos especiales involucrados e incorporar soluciones en una metodología bien diseñada (Dhote y Sarate, 2013).

Bray, T. en el documento elimina las incoherencias con otras especificaciones de JSON y reparan los errores de especificación y ofrece una guía de interoperabilidad del formato. El documento contiene la gramática, tipos de datos y como expresarlos, así como los valores

válidos que se reconocen. Además se incluye información sobre los parsers y generadores, se menciona las aplicaciones que han usado JSON para intercambiar datos y finaliza comentando los problemas de seguridad con los lenguajes de scripting. (Bray, 2014).

Chimoy García & all. desarrollan una tesis en la institución educativa Salesiano: La implementación del sistema web en la institución para optimizar los procesos administrativos. Utilizan el lenguaje de programación del Hypertext Preprocessor (P.H.P) en base al modelo, vista, controlador (M.V.C) y como gestor de base de datos MySql que es de código abierto y multiplataforma. Como resultado tienen un sistema donde registran, modifican o eliminan la información acerca de los alumnos, y de esta manera la institución logró la disminución de la pérdida de información, tiempos de atención más cortos (de 40 min. a 17). (Chimoy, 2016).

Gude & Munawat realizan un estudio empírico sobre un gran corpus de programas escritos en JavaScript, puesto que los avances en este no han sido respaldados por estudios empíricos a gran escala. El documento abarca dos características: Una es como se declaran las variables, su alcance dentro del bloque y su alcance de la función, la segunda característica es la forma en que las sentencias for in son usadas. Los programas recolectados para el corpus se obtuvieron de 3 maneras: 1.- Rastreadores web, 2.- Lista de 100 sitios top y 3.- Complementos de Mozilla Firefox. El corpus fue analizado mediante Los complementos de Mozilla Firefox V8 que es el motor de JavaScript de código abierto de Google. En los resultados, en el uso de variables sugiere la declaración de variables locales que se puede implementar mediante el estándar EcmaScript 6, para que las variables solo existan en el bloque declarado. (Gude, Hafiz, y Wirfs-Brock, 2014)

Montenegro describe diferentes formatos de marcado que permiten generar fragmentos enriquecidos dentro de una página Web, comenta que estos fragmentos permiten mejorar los resultados de búsqueda en la web. Menciona que los fragmentos enriquecidos son entidades usadas en las páginas Web para describir un tipo concreto de información. Comenta el autor que el objetivo del artículo es mostrar paso a paso las actividades que deben ser ejecutadas por un editor de páginas Web para enriquecer fragmentos de código HTML con información semántica. Mejorando la visibilidad de sitios Web usando tecnología semántica (Montenegro, Ochoa, y Espinoza-Mejía, 2016).

Sanchez realiza una investigación e implementación de un ERP de gestión, que cubriera las áreas funcionales de la empresa Servi Frio. El autor realiza una aplicación en un modelo de conexión de tres capas. La primera Capa (Capa de presentación) utiliza un servidor apache y un framework de diseño bootstrap, la segunda capa (Capa de Negocio) es programada en PHP, recibe las peticiones hechas por el usuario desde la capa de presentación y se envían las respuestas después de realizar los procesos correspondientes, y por último la tercera capa (Capa de Datos) cuenta con un motor de base de datos MySQL permite el almacenamiento, modificación y extracción de la información. El autor concluye que al usar herramientas de Software Libre se reducen significativamente los costos de implementación y recomienda un mantenimiento periódico (Sánchez López, 2017).

Kulkarni et. all examinan los diferentes ERP en el mercado del entorno educativo. Primeramente identifican los factores críticos para cada una de las 6 fases, y elijen cuatro productos

con sus características y definen sus características principales. Seguido de ello proponen seis pasos para una evaluación ERP efectiva y selección de software, dentro de estos pasos resaltan una Lista de requisitos genéricos detallados que ayudan a analizar de manera cuantitativa. Realizan una tabla comparativa tomando en cuenta una muestra, indican la metodología y los aspectos claves estudiados. Los autores consideran que esta elección y/o consideraciones se deben tomar mucho en cuenta, ya que las implementaciones pueden llegar a durar 10 años o más. Los autores concluyen que el estudio ayuda a elegir la solución correcta para las instituciones educativas en función de los factores mencionados anteriormente(Kulkarni y cols., 2015).

Wan Deng indican en su artículo el diseño y la aplicación de Front-End MVCC Framework software basado en la plataforma de gestión, proponen un modelo de desarrollo similar al MVC basado en el front end, combinado con las necesidades de la aplicación web para mejorar el código complejo que resulta al elaborar un proyecto de grandes magnitudes. Indican que con el uso de JQuery para controlar la vista se vuelve cada vez más obsoleto por lo que es necesario integrar características de gestión para diseñar frameworks distintos de JQuery. El modelo MVVW que proponen cuenta con enlace de datos bidireccional, modularidad, separación entre el cliente y el servidor y el control de ruta. Los autores resumen mediante el modelo propuesto se puede desarrollar un sistema contráctil y ampliable, que sea fácil de mantener(WANG y DENG, 2016).

Huang & Sundberg registran una patente que ellos definen como: "método, aparato y producto de programa utilizable por ordenador para almacenar datos de sesión del HTTP", este se utiliza para introducir a un navegador web estados en dentro de una aplicación. Comentan los autores que otro método que existe actualmente para la persistencia de datos es almacenar los datos de sesión mediante cookies, pero es frecuente que la configuración del navegador no permita almacenarlas y además estas son almacenadas en memoria lo que podría ocasionar que gran cantidad de información pudiera saturarle, también mencionan otros métodos menos efectivos como reescritura de URL's y almacenamiento de datos de sesión en una base de datos, siendo este último más lento para la recuperación de datos.(Huang y Sundberg, 2016)

Capítulo 4

Desarrollo de la solución

4.1. Herramientas

Estrictamente todas las herramientas son del tipo OpenSource, se utiliza; PHP Versión 7.1.1, HTML5, Apache Versión 2.4.25, MariaDB Versión 10.1.21, Sublime-Text Versión 3 (versión sin registro). Se desarrolló sobre plataforma Windows y plataforma Linux. Como paradigma de programación, la orientación a objetos (POO).

4.2. El enfoque del sistema

Se establecieron las siguientes premisas:

- Mediante la reutilización y el uso de la herencia reducir al mínimo los códigos generadores de la interfaz de usuario y también la transferencia de la información entre los clientes y el servidor.
- Un cliente web debe estar basado o extender un componente plantilla adaptativo.
- Los principios de la POO deben prevalecer a lo largo de todo el desarrollo con la finalidad de promover la creación de componentes de software y la reutilización.
- Todas las herramientas de desarrollo han de ser OpenSource en categoría de estables.
- La comunicación entre los clientes y el servidor siempre será asíncrona y solo será posible mediante el listener del módulo BackEnd, el cual arbitra el acceso al servidor.

4.3. Diseño del FrontEnd

La Arquitectura de Software utilizada es inspirada en el MVC(Modelo Vista Controlador) al que se le adaptó un Motor de Comunicaciones basado en Ajax. Los principios del MVC(González y Romero, 2012) han sido la base de la Arquitectura aplicada, por ejemplo,

se aplica la separación de los datos de la lógica de negocio con la finalidad del mejoramiento de reusabilidad, mantenimiento y escalabilidad del código. Con la utilización de este enfoque se estará previendo que, cuando se realice alguna modificación en las vistas tenga el menor impacto en la lógica de los datos. En la figura 4.1 se describe los elementos.

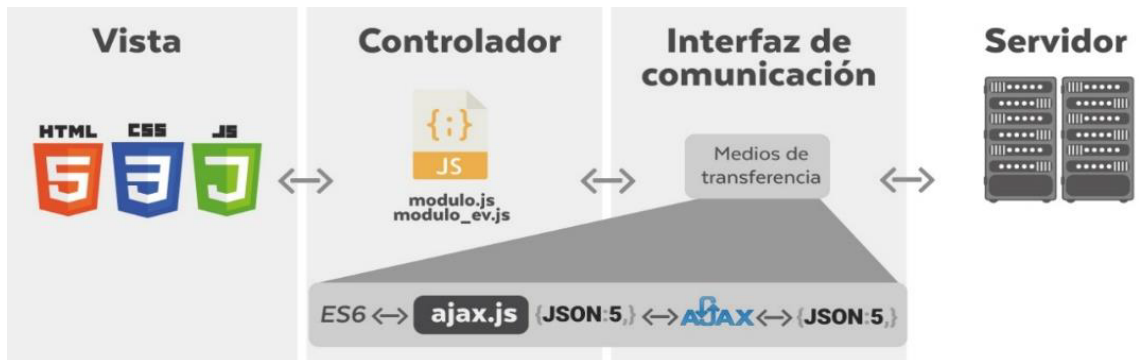


Figura 4.1: Arquitectura MVC adaptada al FrontEnd Fuente: Propia del Proyecto

4.3.1. Vista

En la vista destacan dos elementos implementados: El Diseño Web adaptativo y las Clases utilizadas.

Diseño Web Adaptativo

Se aplican los principios de la web adaptativa (Marcotte, 2011). Marcotte indica que existen tres elementos fundamentales para lograr un diseño web adaptativo, que son: Cuadrícula Fluida, Imágenes Flexibles y Media queries.

Clases utilizadas

Dado el número de veces que se podría ocupar en diferentes clientes el componente de materias y tablas, se decidió la implementación de las siguientes clases escritas en el lenguaje de programación JavaScript que en conjunto son capaces de llevar una representación orientada a objetos. De forma descendente la clase superior consume a la clase inferior como lo indica la Figura 4.3 y Figura 4.2. Actualmente se utiliza en el módulo de Alumnos y el Módulo de Jefes de Carrera.

Tabla: Con esta clase se define una estructura que puede agrupar uno o más objetos de tipo registro según se requiera, esta agrupación que se hace con la ayuda del contenedor div permite representar la información al usuario final como una tabla.

Registro: Permite la agrupación múltiples objetos de tipo Campo, tantos como sea necesario incluir dentro de un mismo registro, para mostrarlos en la interfaz se utiliza un contenedor

de tipo div.

Campo: Esta clase se utiliza para crear un campo que es la unidad mínima que contiene el dato JSON que estamos recibiendo, en el constructor se recibe nombre y contenido de dicho campo, estos datos se muestran en la interfaz por medio de un contenedor de tipo span.

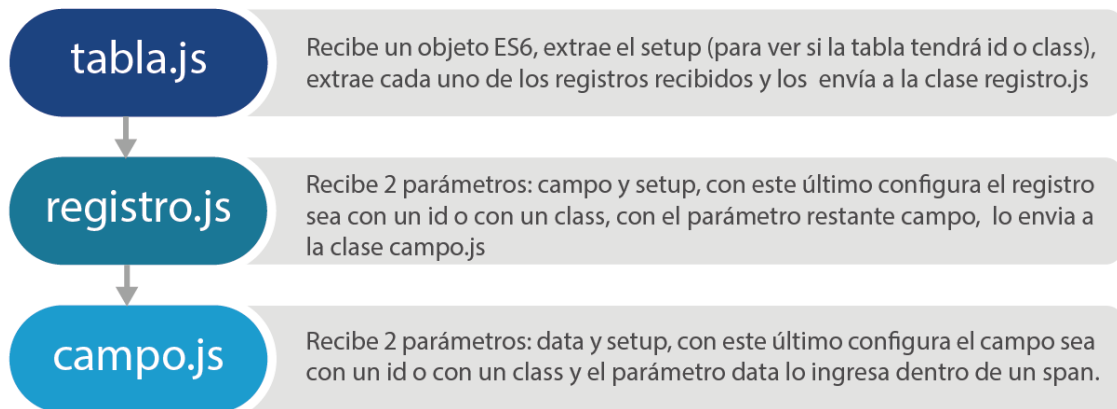


Figura 4.2: Imagen muestra la manera en la que se consume cada una de las clases para implementar tabla, registro y campo.

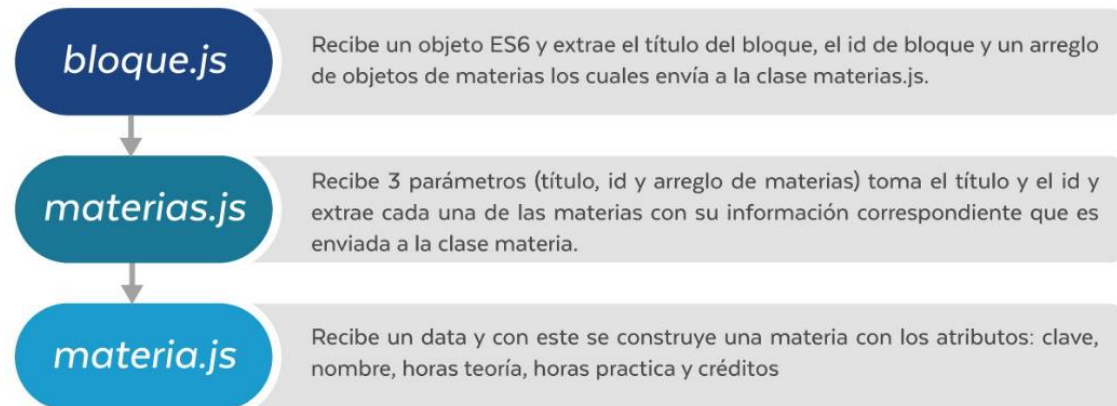


Figura 4.3: Imagen muestra la manera en la que se consume cada una de las clases para implementar bloques, materias y materia.

4.3.2. Controlador

El cliente inicia con la clase `modulo_ev.js`, la cual crea una instancia de `modulo.js` que a su vez extiende a `sitm.js`.

En la clase `modulo.js` se definen los parámetros necesarios para armar los componentes junto con sus funciones, se agregan elementos en el área de contenido y se obtiene la Data en Session Store proveniente de la consulta hecha al BackEnd retornada por `Ajax.js`.

La clase `sitm.js` es la plantilla y es la responsable del armado de la vista, en ella se realizan las funciones de:

- Designar la estructura y áreas correspondientes para insertar cada componente.
- Define la estructura de los componentes principales (Menú lateral, Menú superior y encabezado).
- Define el comportamiento y las propiedades de cada componente principal.
- Realiza el insertado de cada componente principal al finalizar su creación

4.3.3. Interfaz de comunicación

El segmento de interfaz de comunicación está constituido por la clase denominada `ajax.js` que crea y envía peticiones asíncronas de métodos AJAX consumiendo objetos de tipo `ES6` que son convertidos a JSON para ser enviados al servidor, este a su vez devuelve una respuesta en formato JSON que se convierte en respuesta `ES6` al cliente. JSON es consumido a través de los métodos `select`, `insert`, `update` y `delete` que realizan consultas básicas a la base de datos de manera transparente al cliente, mientras que el método `method` realiza la solicitud para realizar una operación al servidor donde se obtienen datos relevantes de acceso del cliente y mejorar la seguridad.

4.4. Diseño del BackEnd

Dada su magnitud, el BackEnd, fue enfocado desde una perspectiva Down-Top. Aplicando diseño en espiral, la implementación concluyó en cuatro clases, que, en conjunto, son capaces de llevar a cabo una representación orientada a objetos de una base de datos y sus elementos. De forma descendente, la clase superior consume a la clase inferior, sin embargo, cada clase por sí misma permite la construcción de objetos autónomos y autodescriptivos. En la Figura 4.4 se resume la lógica de implementación aplicada.

4.4.1. Clases para describir el modelo de datos

Existen 4 Clases que forman parte del modelo de datos, programadas en PHP.

Campo: Constituye el molde para crear la unidad mínima que puede ser representada en un sistema de bases de datos, el campo. Su constructor permite definir el nombre, el tipo de



Figura 4.4: Modelo de factorización Down-Top aplicado en la implementación del BackEnd

dato e indicar si se trata de un campo clave. Posee también un método con el cual se puede generar una representación JSON (International, 2015a) de sí mismo.

Registro: Encapsula el conjunto de campos que definen un registro de base de datos. Emulando polimorfismo, su constructor permite crear objetos de diferentes maneras para satisfacer las necesidades de construcción que fueron detectadas durante el diseño. Posee también un método para crear un objeto JSON con la descripción del registro y los valores de sus campos. Estos objetos también sirven de contenedores para las consultas o procedimientos almacenados donde el resultado es un valor o un solo registro. Adicionalmente es capaz de generar el código SQL de las consultas SELECT, INSERT, UPDATE y DELETE tomando como base su propia estructura y los valores de sus campos.

Tabla: Los productos de esta clase son capaces de almacenar tantos objetos de tipo Registro como sean necesarios para describir la estructura de una tabla de base de datos, pueden ser empleados para ejecutar procedimientos almacenados de la base de datos a la cual hayan sido asociados en donde el resultado sea un valor, un registro o un conjunto de registros y son capaces de contener el dataset resultante de una consulta SQL en array dinámico de objetos de tipo Registro. Su constructor polimórfico satisface las necesidades de creación Tablas que fueron detectados durante el diseño. Adicionalmente es capaz de derivar los tipos de datos de los campos de los registros cuando el dataset resultante puede emitir uno u otro tipo de datos.

BD: Esta clase es una extensión de la librería PDO (Popel, 2007) de PHP 7.0 que consigue co-

nectar de forma transparente a diversos gestores de bases de datos como MySQL o MaríaDB, MSSQL, Oracle y SQLite entre otros. Es capaz de contener un array de tablas o resultados de consultas que pueden ser cargadas de inicio para poderlas enviar al cliente como un paquete de datos de tipo JSON. Puede ejecutar la operación SELECT, INSERT, UPDATE Y DELETE de un registro o una tabla. Los descendientes de esta clase, quedan habilitados para implementar un Singleton y así evitar la sobrecarga de conexiones que podría ocasionar una solicitud masiva de peticiones de un cliente. Por defecto, el constructor de la clase conecta con un gestor MySQL.

4.4.2. Intercomunicación con los clientes web

Como interfaz de comunicación con los clientes web, se programó la clase “Proceso.php” lista para ser extendida para crear clases que definan el comportamiento de objetos cuya función será trabajar como un listener y dar atención a los mensajes POST que llegan al servidor.

Cada cliente que se comunique con el server debe hacerlo mediante una petición asíncrona de tipo AJAX y enviar/recibir datos en formato JSON. Inicialmente, el cliente solicitará al intérprete de PHP la creación de un objeto listener específico, el cual, se mantendrá a la escucha de sus peticiones y llevará a cabo las solicitudes de operación que indique, siempre y cuando éstas sean legales.

El objeto listener, funge también como un árbitro que acepta o rechaza solicitudes y sirve de intermediario entre las bases de datos alojadas en el server y las necesidades de información y actualización de ellas que indican los clientes comunicantes, de tal forma que el acceso a las bases de datos solo será posible cuando la operación solicitada sea legal, los parámetros sean los correctos y esté expresado en el formato específico, de lo contrario, el servidor asumirá una amenaza de ataque y devolverá un mensaje de error que quizás terminaría ocasionando un mal funcionamiento en el cliente. Lo anteriormente descrito se muestra en la Figura 4.5.

4.5. Diagramas de Flujo

Se elaboraron diagramas de flujo para facilitar el entendimiento de manera gráfica el algoritmo o proceso para cada una de las opciones del menú del módulo de jefes de carrera (Figura 4.6) y además se indica el diagrama de flujo para la opción de Carga Académica, siendo esta última la clave y proceso principal del módulo anteriormente mencionado (Figura 4.7).

4.6. Diagramas de Clases

En los diagramas de clase que se muestran a continuación(Figura 4.8 y Figura 4.9), se describe la estructura del sistema Front End mostrando las clases del sistema y sus relaciones entre ellos. En la Figura 4.8 se muestra el diagrama de clases del módulo Jefes de carrera y

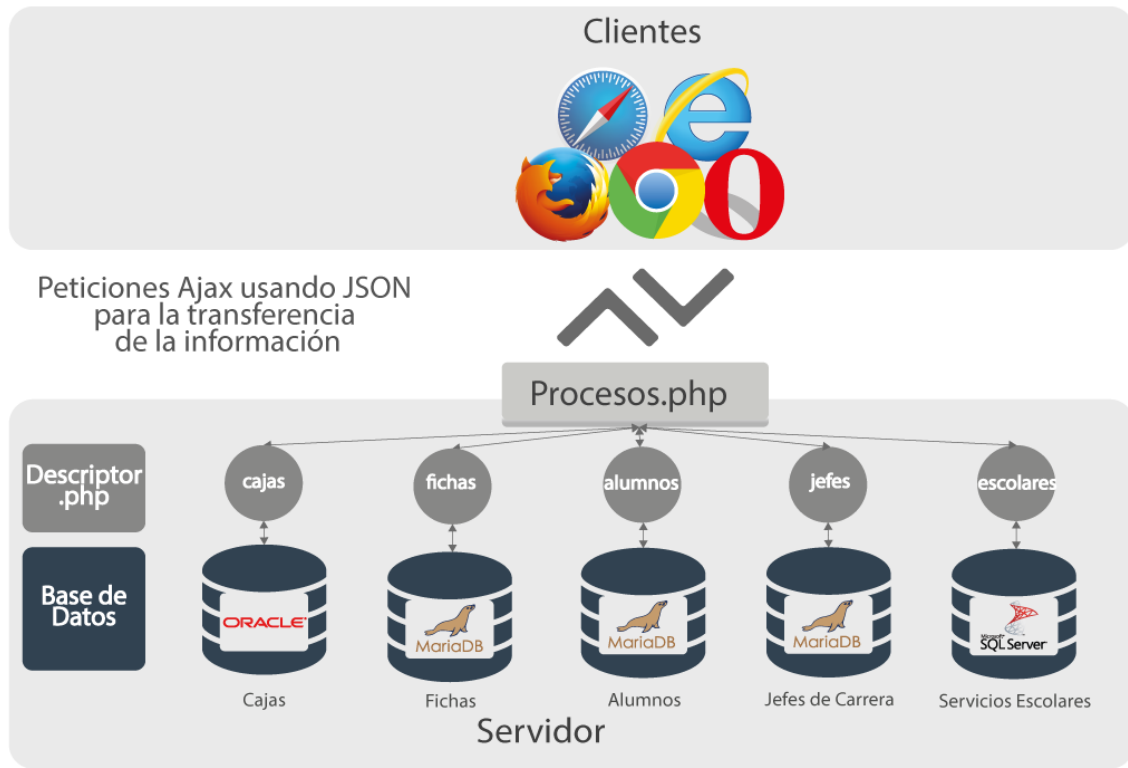


Figura 4.5: Comunicación de los diferentes cliente con el Servidor mediante peticiones Ajax.

en la Figura 4.9 se indica la herencia de los descriptors de la clase descriptor.php con que están contenidas dentro del backend.

4.7. Diagrama de Secuencia

Los diagramas de secuencia mostrados a continuación indican la interacción entre el usuario, el Frontend y el Backend. El primer diagrama de secuencia ilustra el comportamiento de la opción de carga académica (ver Figura 4.10), seguido de el diagrama de secuencia para la opción de Especialidades dentro de la función de Configuración (ver Figura 4.11), el diagrama para la opción de Administrar Grupos la función de Configuración (ver Figura 4.12) y finalmente el diagrama para la opción de Permisos de Inscripción dentro de la función de Configuración (ver Figura 4.13).

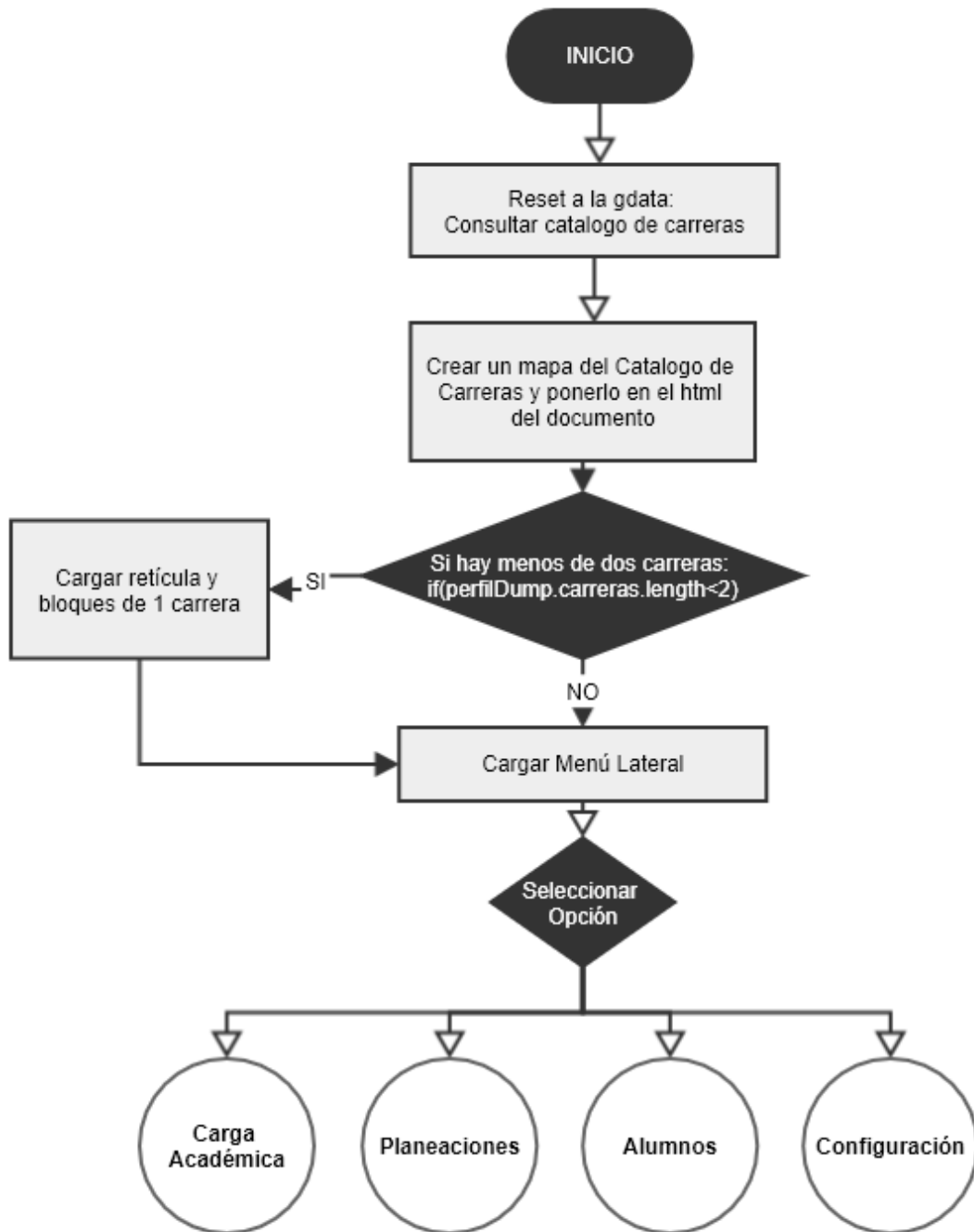


Figura 4.6: Diagrama de Flujo de carga de la Pantalla para el módulo de Jefes de Carrera.

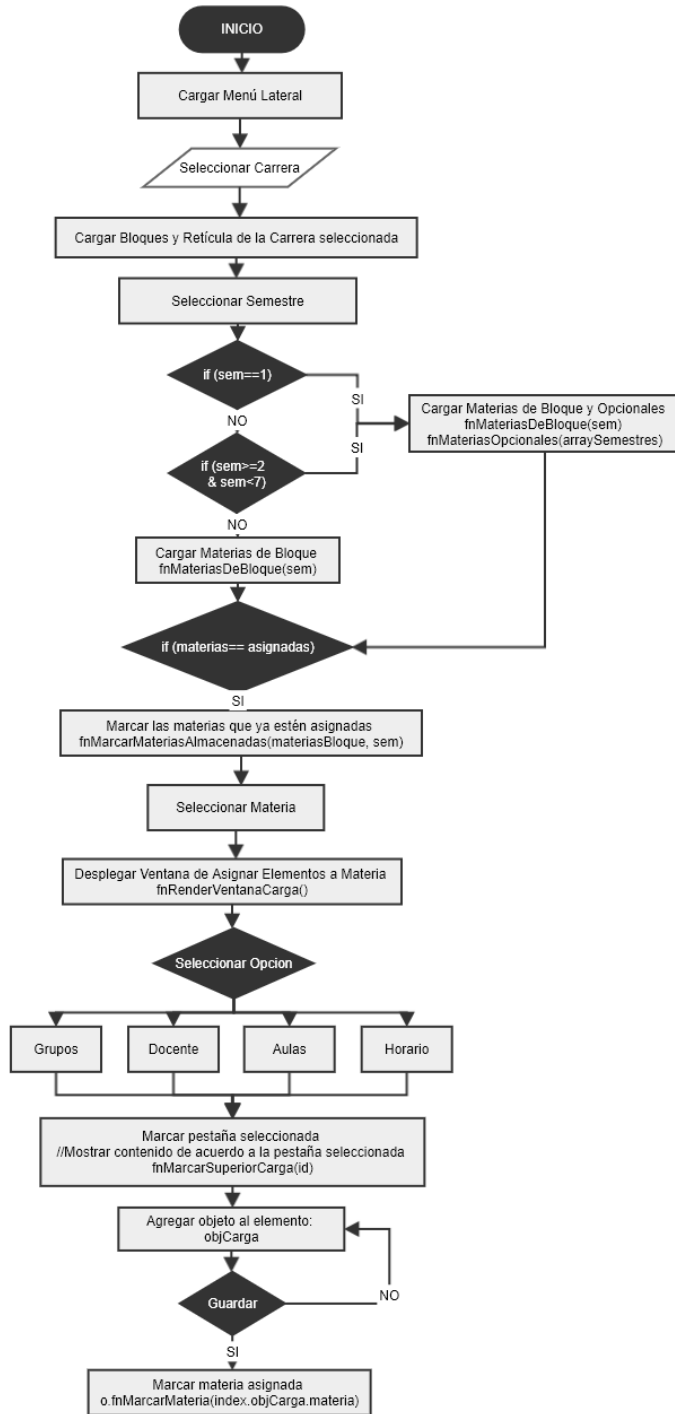


Figura 4.7: Diagrama de Flujo de la Opción Carga Académica.

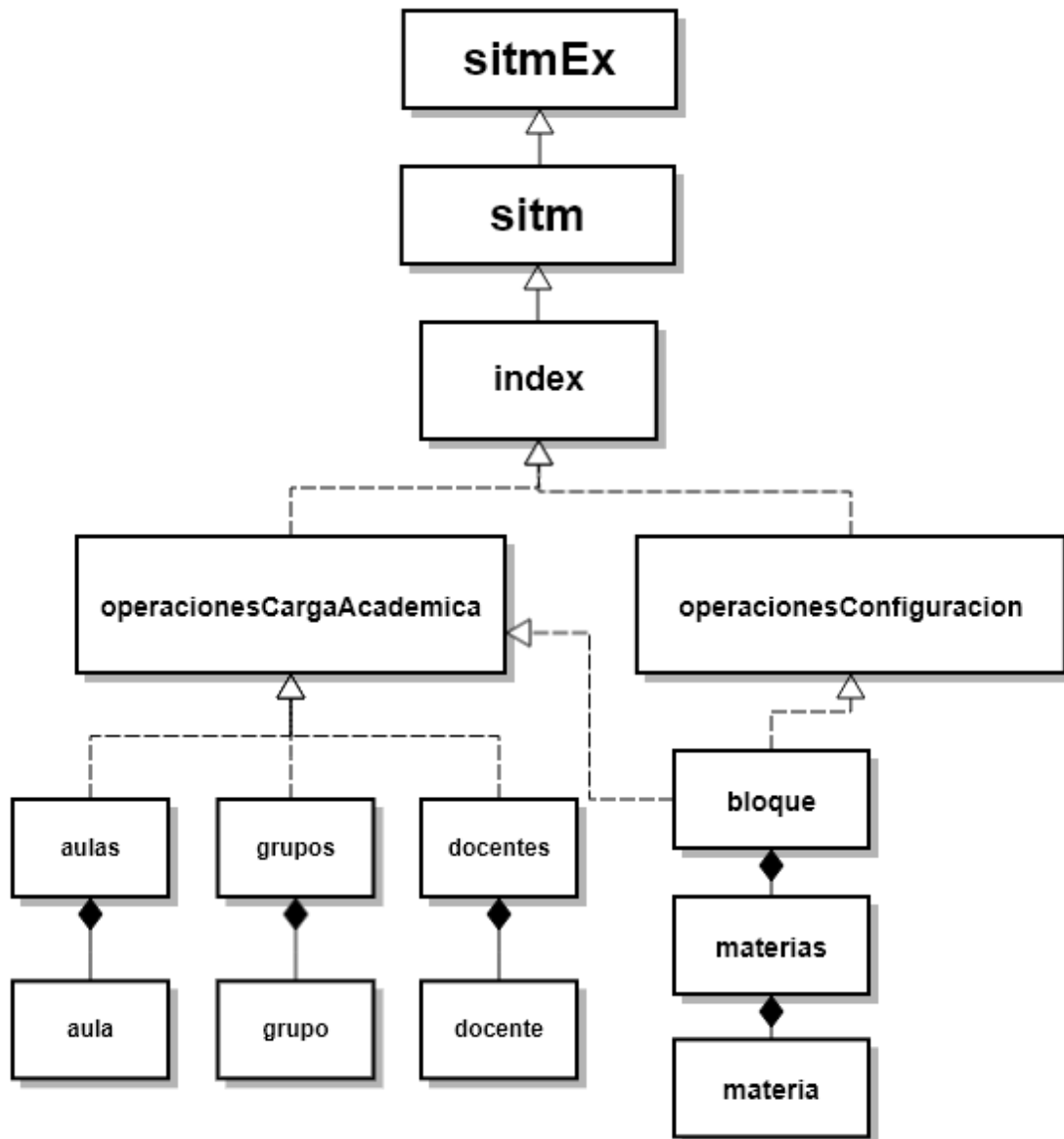


Figura 4.8: Diagrama de Clases FrontEnd del módulo Jefes de carrera.

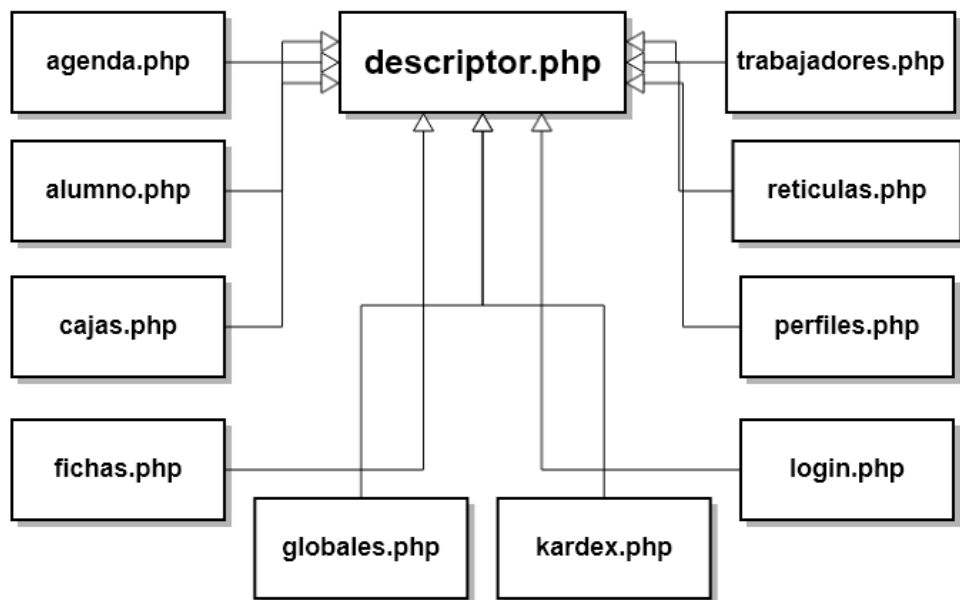


Figura 4.9: Diagrama de Clases de los descriptores del Backend.

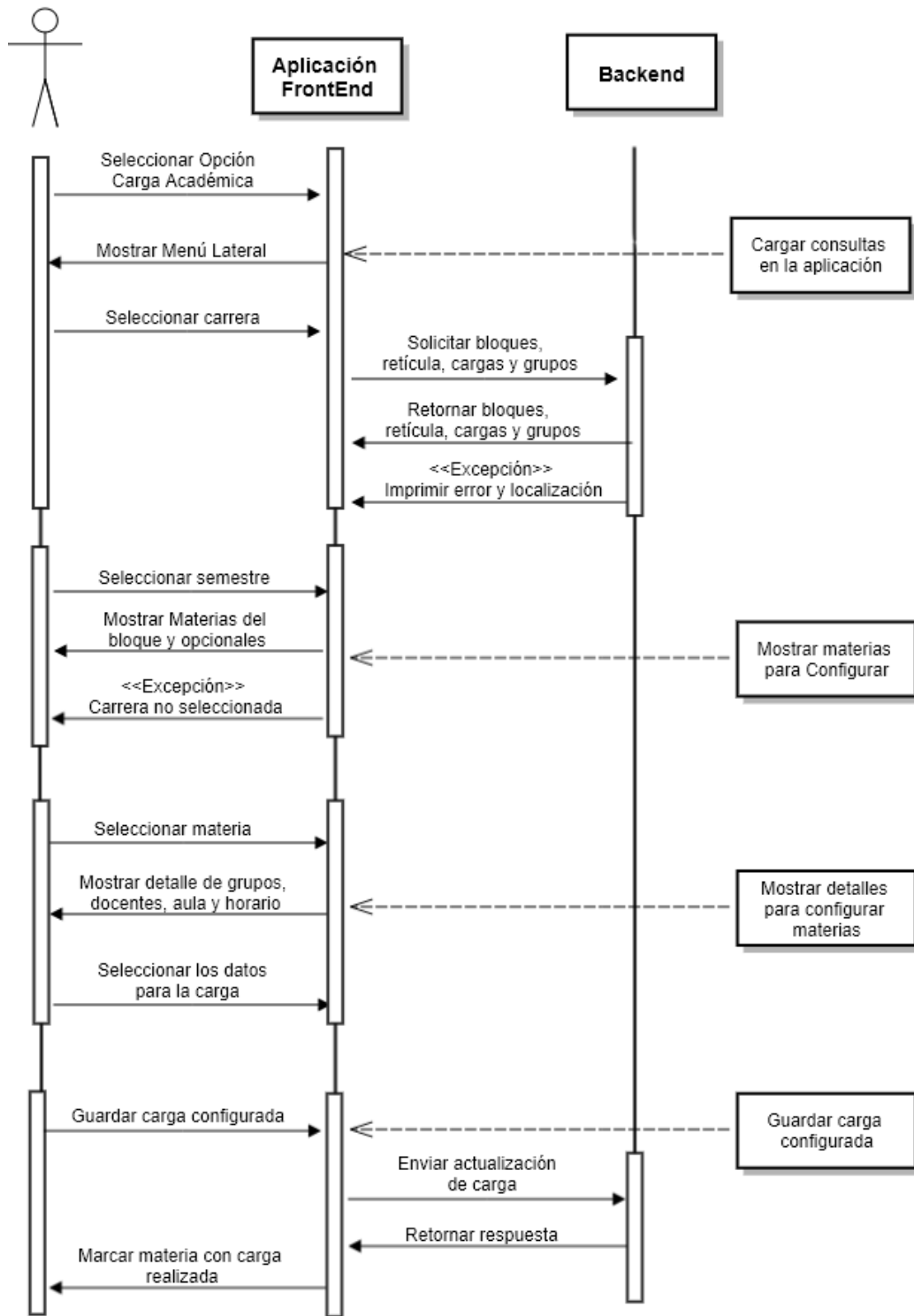


Figura 4.10: Diagrama de Secuencia para la función de Carga Académica.

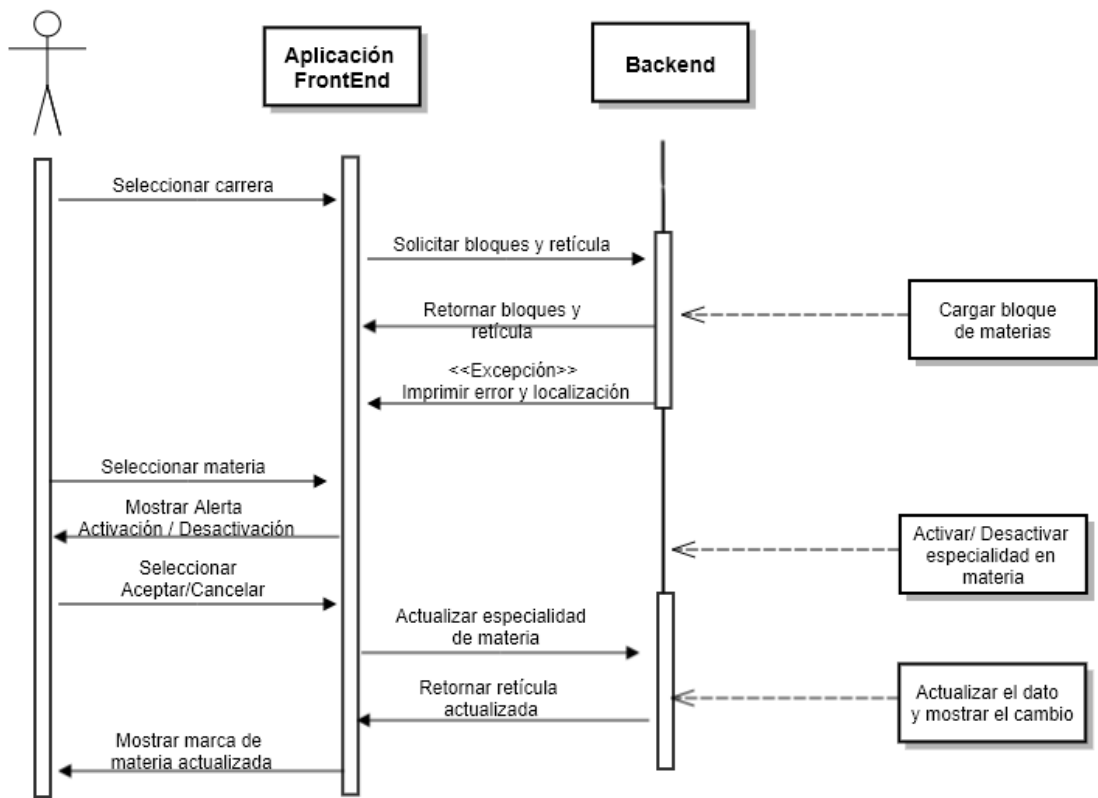


Figura 4.11: Diagrama de Secuencia para la opción de Especialidades dentro de la función de Configuración

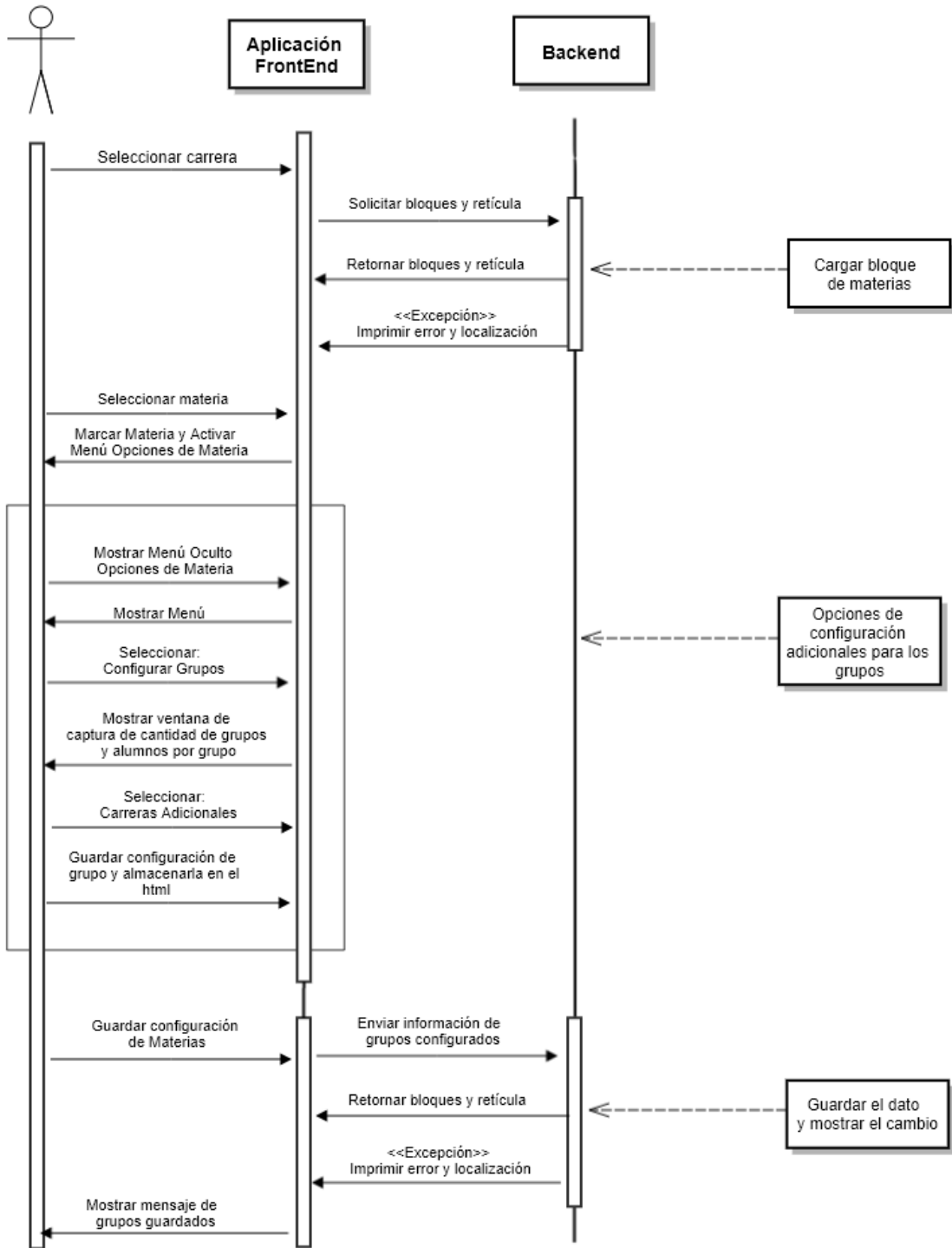


Figura 4.12: Diagrama de Secuencia para la opción de Administrar Grupos la función de Configuración

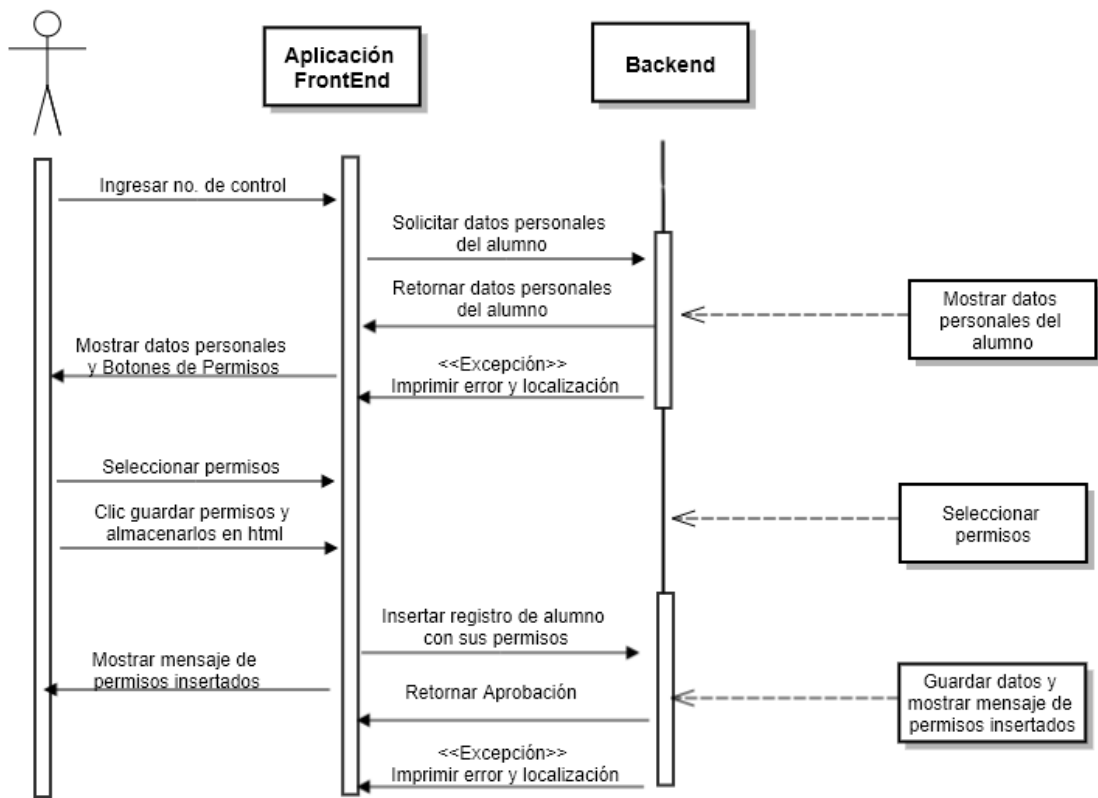


Figura 4.13: Diagrama de Secuencia para la opción de Permisos de Inscripción dentro de la función de Configuración

Capítulo 5

Análisis de resultados

5.1. Pruebas de Verificación y Validación

Las pruebas de verificación que se muestran a continuación garantizan que el software se implementa correctamente una función específica. Tal es el caso de la librería de mensajes que fue construida en base a las premisas que se establecieron dentro del desarrollo de la propuesta en el capítulo 4.

5.1.1. Líneas de Código: Librería mensajes.js

Alertify.js¹ es una pequeña biblioteca que ofrece diálogos de navegador ligeros, bootstrap-dialog es igualmente una librería de código abierto usada para el framework Bootstrap², mensajes.js es una pequeña librería creada exclusivamente para el proyecto sitm.

| | mensajes | alertify | bootstrap-dialog |
|------------------|----------|----------|------------------|
| Version | 1 | 1.11.0 | |
| Líneas de código | 102 | 3595 | 1394 |
| Peso (kb) | 6.4 | 136 | 47 |
| Responsivo | Si | Si | Si |
| Uso JQuery | No | No | Si |

Cuadro 5.1: Comparativa de la librería mensajes, alertify y bootstrap.dialog en Javascript

Al comparar la librería mensajes.js vs alertify y bootstrap en cuadro 5.1.1, observamos que mensajes.js tiene menos líneas y kb, por lo que la transferencia de datos entre el cliente y el servidor se verá reducida, mantiene aún así su diseño responsivo y elimina el uso del jquery haciendolo más eficiente.

¹Librería de código abierto, disponible desde <http://alertifyjs.com/>

²Librería de código abierto, disponible desde <https://getbootstrap.com/docs/4.0/components/modal/>

5.1.2. Análisis de Seguridad Web

Acunetix Vulnerability Scanner ³ es un aplicación sobre un entorno web que realiza pruebas de seguridad de aplicaciones web automatizadas. Se han sometido a un análisis las aplicaciones web de SII Minatitlan ⁴, Veracruz⁵ y Ciudad Madero⁶, estos sistemas fueron elegidos debido a que a las Instituciones pertenecen al Tecnológico Nacional de México, para que el análisis fuera en igualdad de condiciones, el SITM se montó sobre un servidor Apache Versión 2.4.34. A continuación se comparan las vulnerabilidades de cada sistema SII y del SITM.

| Sistema / Vulnerabilidades | Alta | Media | Baja | Informacional |
|----------------------------|------|-------|------|---------------|
| SITM | 0 | 0 | 3 | 14 |
| SII Madero | 20 | 24 | 3 | 24 |
| SII Veracruz | 0 | 6 | 2 | 4 |
| SII Minatitlán | 5 | 5 | 3 | 1 |

Cuadro 5.2: Comparativa entre entre SITM e implementaciones del SII

En el cuadro 5.1.2 podemos ver como el SITM no tiene vulnerabilidades de alta magnitud comparados con sistemas similares pertenecientes al TecNM. Las vulnerabilidades principales que mostraron los SII de Madero y Minatitlán fueron de Inyección de Código SQL y Cross-site Scripting.

5.2. Compatibilidad entre navegadores

Debido a que los navegadores responden de diferente manera con un mismo código, se tomo en cuenta los 3 navegadores más utilizados actualmente: Chrome, Edge/IE y Firefox. A continuación se muestran como el contenido se adapta a ellas Figuras: 5.1, 5.2, y 5.3,.

5.3. Diseño Responsivo en diferentes tamaños de Pantalla

En ambas Figuras 5.4 y 5.5 se puede ver cómo se distribuyen las pantallas en equipos con diferentes sistemas operativos. Se tomaron como base 1 dispositivo por cada uno de los sistemas operativos más utilizados (Mac OS, Android y Linux).

³Software para auditar seguridad Web, disponible desde <https://www.acunetix.com/vulnerability-scanner/> versión trial

⁴<https://sii.itmina.edu.mx/>

⁵<https://http://sii.itver.edu.mx/>

⁶<https://http://sii.itcm.edu.mx//>



Figura 5.1: Imagen del módulo Jefes de Carrera con el navegador Chrome.

5.3.1. Validación de la Arquitectura

La validación de la arquitectura y metodología propuesta se realizó mediante la construcción del módulo Adeudos, este módulo tienen Acceso tanto Jefes de Departamento donde se realizen prestamos o evaluaciones, así como los alumnos. El objetivo fue demostrar que el sistema que se está modelando con la arquitectura opera satisfactoriamente. En la Fig. 5.6 se muestra el cronograma de actividades seguidas.



Figura 5.2: Imagen del módulo Jefes de Carrera con el navegador Edge



Figura 5.3: Imagen del módulo Jefes de Carrera con el navegador Firefox.



a) Apple iPad Air 2 2048x536

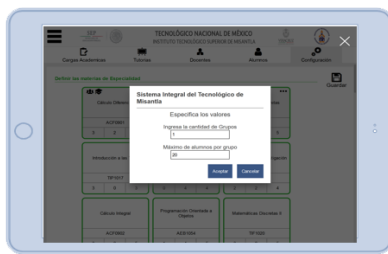


b) Samsung Galaxy Note 3 1920x1080



c) Laptop 1366x768

Figura 5.4: Imagen del módulo Jefes de Carrera, opción carga académica.



a) Apple iPad Air 2 2048x536



b) Samsung Galaxy Note 3 1920x1080



c) Laptop 1366x768

Figura 5.5: Imagen del módulo Jefes de Carrera, opción configuración.

| ACTIVIDADES | SEMANAS | | | |
|---|---------|---|---|---|
| | 1 | 2 | 3 | 4 |
| PROGRAMADOR 1 | | | | |
| FUNCION ALTA DE PRESTAMOS | ■ | ■ | | |
| FUNCION BAJA DE PRESTAMOS | | ■ | | |
| FUNCION MODIFICACION DE PRESTAMOS | | | ■ | |
| FUNCION VISUALIZACIÓN DE PRESTAMOS | | | | ■ |
| PRUEBA Y EVALUACIÓN DE LAS FUNCIONES | | ■ | ■ | ■ |
| PROGRAMADOR 2 | | | | |
| ELABORACION DE BASE DE DATOS ADEUDOS Y PROCEDIMIENTOS ALMACENADOS | ■ | | | |
| ANÁLISIS Y ELABORACIÓN DEL DESCRIPTOR PARA LA BASE DE DATOS ADEUDOS | ■ | | | |
| CREACIÓN DE CONSULTAS TIPO AJAX PARA LAS FUNCIONES DE PRESTAMOS | | ■ | | |
| CRECIÓN DE REPORTES DE ALUMNOS Y JEFES DE DEPARTAMENTO | | | ■ | ■ |
| PRUEBA Y EVALUACIÓN DE LAS FUNCIONES | | ■ | ■ | ■ |

Figura 5.6: Cronograma de actividades módulo: Adeudos.

Capítulo 6

Conclusiones y Trabajos futuros

6.1. Conclusiones

El proyecto SITM se encuentra en fase de desarrollo. Su enfoque orientado a componentes permite establecer un sistema de mantenimiento en cascada y hacerlo crecer mediante la integración de nuevos módulos que cumplan las premisas de desarrollo que fueron establecidas.

La capacidad de reutilización del módulo BackEnd y los componentes de la plantilla del FrontEnd podrán ser utilizados en futuros desarrollos de software.

Mediante la reutilización y la herencia se redujeron significativamente las líneas de código por lo que la transferencia entre el cliente y el servidor será más rápida.

El sistema en fase alfa mediante las pruebas de verificación y validación que se realizaron reflejaron un comportamiento aceptable frente a aplicaciones similares.

Las pruebas de seguridad realizadas permitieron corroborar que el sistema no permite la inyección de código sql.

6.2. Trabajo futuro

Mediante la utilización de alguna técnica de Inteligencia Artificial se propone realizar la generación automática de horarios. La estructura actual de la base de datos permite crear matrices de restricciones duras y blandas para optimizar horarios flexibles, que cumplan las restricciones duras y optimicen las restricciones blandas. Algunas de las técnicas abordadas en la literatura, son los algoritmos genéticos, colonia de hormigas y sistemas multiagentes. Al concluir la primera implantación del sistema se procederá a agregar un módulo que integre al SITM la gestión de la biblioteca, tutorías y los exámenes en línea.

Referencias

- Bray, T. (2014). The javascript object notation (json) data interchange format.
- Chimoy, T. (2016). Implementación de un sistema de matrícula web para optimizar los procesos administrativos utilizando la metodología del Modelo Vista Controlador en la Institución Educativa Salesiano.
- Dhote, M. R., y Sarate, G. (2013). Performance testing complexity analysis on ajax-based web applications. *IEEE software*, 30(6), 70–74.
- Elmasri, R. R. (2008). *Fundamentals of database systems* (n.º Sirsi) i9789701513286).
- Firtman, M. R. (2008). *Ajax: Web 2.0 para profesionales* (n.º Sirsi) i9789701513286).
- González, Y. D., y Romero, Y. F. (2012). Patrón modelo-vista-controlador. *Revista Telemática*, 11(1), 47–57.
- Gude, S., Hafiz, M., y Wirfs-Brock, A. (2014). Javascript: The used parts. , 466–475.
- Guevara, L. V. D. (2018). *Gestión de bases de datos*. <https://media.readthedocs.org/pdf/gestionbasesdatos/latest/gestionbasesdatos.pdf>.
- Harris, A. (2009). *Programación con php 6 y mysql*. Anaya Multimedia.
- Huang, S., y Sundberg, S. M. (2016, octubre 4). *Implementing browser based hypertext transfer protocol session storage*. Google Patents. (US Patent 9,459,888)
- International, E. (2015a). 404: The json data interchange format. , 9.
- International, E. (2015b). *Ecmascript 6, tipo @ONLINE*. Descargado 2018-02-15, de <https://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>
- International, E. (2018). *Ecmascript® 2018 language specification*. <https://media.readthedocs.org/pdf/gestionbasesdatos/latest/gestionbasesdatos.pdf>.
- Json.org. (2013). *Introducción a json*. Descargado 2018-02-15, de <https://www.json.org/json-es.html>
- Korth, H., y Silberschatz, A. (1993). *Fundamentos de bases de datos*. Madrid.
- Kulkarni, A., Hegde, N., Sharma, M., Kulkarni, A. A., Hegde, N., y Sharma, M. (2015). Educational erp systems in the market a comparative study. *International Journal of Innovative Research Science in Technology*, 1.
- Lin, B., Chen, Y., Chen, X., y Yu, Y. (2012). Comparison between json and xml in applications based on ajax. , 1174–1177.
- Marcotte, E. (2011). *Responsive web design* (2da Edition ed.). A Book Apart.

- Montenegro, L., Ochoa, V., y Espinoza-Mejía, M. (2016). Mejorando la visibilidad de sitios web usando tecnología semántica. *Maskana*, 6 (Ed. Esp.).
- NetMarketShare. (2018). *Browser market share*. Descargado 2018-02-22, de <https://netmarketshare.com>
- Pantoja, E. B. (2004). El patrón de diseño modelo-vista-controlador (mvc) y su implementación en java swing. *Acta Nova*, 2(4), 493.
- Popel, D. (2007). *Learning php data objects*. Packt Publishing Ltd.
- Pressman, R. S. (2010). *Ingeniería del software* (7ma Edition ed.). McGraw-Hill.
- Pressman, R. S., y Troya, J. M. (1988). *Ingeniería del software*. McGraw Hill.
- Sánchez López, S. I. (2017). *Sistema erp orientado a la web para el control de inventarios de productos y servicios de la empresa servi frio en la ciudad de quevedo*. (B.S. thesis).
- Valzacchi, J. R. (2003). *Internet y educación aprendiendo y enseñando en los espacios virtuales*. INTERAMER.
- W3C. (2015). *Hoja de estilo en cascada (css)*. Descargado 2018-02-16, de <https://www.w3.org/wiki/Es/CSS>
- WANG, J.-y., y DENG, F. (2016). The design and application of front end mvc framework based on management platform. *Destech Transactions on Engineering and Technology Research*(iceea).

Anexos

6.3. Encuesta sobre Navegadores más utilizados

Los navegadores más populares según la W3Schools ¹ que en base a más de 50 millones de visitas mensuales que recibe, arroja la siguiente información:

| Navegador | Chrome | Edge/IE | Firefox | Safari | Opera |
|-------------------|--------|---------|---------|--------|-------|
| Porcentaje de uso | 79.0 % | 3.9 % | 10.9 % | 3.2 % | 1.6 % |

Cuadro 6.1: Uso de exploradores según la w3School para el mes de Mayo de 2018

En la tabla se puede notar como es que los principales navegadores utilizados son Chrome: Edge/IE y Firefox.

6.4. Código HTML

Actualmente se está trabajando para el módulo de Jefes de Carrera, en el apartado de Horarios, donde inicialmente el Jefe de Carrera Visualizará las materias a ofertar en el semestre correspondiente. Inicialmente para elaborar el diseño se comenzó a realizar estáticamente para generar la estructura y los estilos deseados.

Código HTML para generar materias dentro de contenedores estáticamente. Son 113 líneas.

Código JavaScript. Código de JavaScript Orientado a Objeto en el cual se crea el objeto "Materia".

```
1 class materia {
2 constructor (data) {
3 this.html = '<div class="materia"><span id="materia">'+ data.materia + '</span
  > <span id="Cl">'+ data.clave + '</span><span id="Ht">'+ data.hTeoria + '</
  span><span id="Hp">'+ data.hPractica + '</span><span id="Cr">'+ data.
  Creditos + '</span></div>';
4 }
5 }
```

¹sitio web para aprender tecnologías web en línea, disponible desde <https://www.w3schools.com/browsers/> Mayo 2018

Código de JavaScript Orientado a Objeto en el cual se crea el objeto "Contenedor" que contendrá a varios objetos de tipo: "Materia". Son 5 líneas.

```
1 class contenedor{
2 constructor (titulo , id , data) {
3 this.html = "";
4 for (var i = 0; i <data.length; i++)
5 {
6 this.html += new materia(data[i]).html;
7 }
8 this.html = '<section class="contenedor" id="' + id + "'><h3>' + titulo + '</h3>'
   + this.html + '</section>';
9 }
10 }
```

Código CSS

Cada estilo utilizado en el sistema esta mostrado a continuación. Son 10 líneas.

```
1 *{
2 margin: 0px;
3 padding: 0px;
4 border: 0px;
5 font-family: Roboto, arial, sans-serif;
6 list-style: none
7 }
8
9
10 .contenedor{
11 padding: 20px 0px;
12 width: 94%;
13 margin:10px 3%;
14 background-color: #FFF;
15 display: flex;
16 flex-flow:row wrap;
17 justify-content: center;
18 }
19
20 .contenedor h3{
21 min-width: 96%;
22 height: 20px;
23 margin: 0 2%;
24 margin-bottom: 10px;
25 font-size:.9rem;
26 }
27
28 #obligatorias h3{
29 color:#4F4F4F;
30 border-bottom: 1px solid #4F4F4F;
31 }
32
```

```
33 #bloque h3{
34 color:#25B22A;
35 border-bottom: 1px solid #25B22A;
36
37 }
38
39 #opcionales h3{
40 color:#E5B803;
41 border-bottom: 1px solid #F0F000;
42 }
43
44 #conflicto h3{
45 color:red;
46 border-bottom: 1px solid red;
47 }
48
49 .materia{
50 border-radius: 10px;
51 color:rgb(95,95,95);
52 max-width: 191px;
53 width: 191px;
54 font-size: .8rem;
55 min-width: 191px;
56 border:1px solid rgb(264,264,264);
57 height: 160px;
58 margin: 5px 5px;
59 display: flex;
60 flex-flow:row wrap;
61 font-weight: lighter;
62 }
63
64 .materia:hover{
65 background: #f2eff1;
66 transition-duration: .5s;
67 cursor:pointer;
68 }
69
70 #bloque .materia{
71 border: solid 3px #25B22A;
72 color: #000;
73 }
74
75 #opcionales .materia{
76 border: solid 3px #F0F000;
77 }
78
79 #obligatorias .materia{
80 border:solid 3px #444343;
81 }
82
83 #conflicto .materia{
```

```
84 border:solid 3px red;
85 }
86
87 #materia{
88 height: 94px;
89 width: 100%;
90 text-align: center;
91 display: flex;
92 align-items: center;
93 justify-content: center;
94 }
95
96 #Cl{
97 border-top:1px solid rgba(0,0,0,0.5);
98 border-bottom:1px solid rgba(0,0,0,0.5);
99 height: 30px;
100 width: 100%;
101 justify-content: center;
102 display: flex;
103 align-items: center;
104 }
105
106 #Ht, #Hp, #Cr{
107 width: 32.3%;
108 text-align: center;
109 height: 30px;
110 justify-content: center;
111 display: flex;
112 align-items: center;
113 }
114
115 #Hp{
116 border-left:1px solid rgba(0,0,0,0.5);
117 border-right:1px solid rgba(0,0,0,0.5);
118 }
```

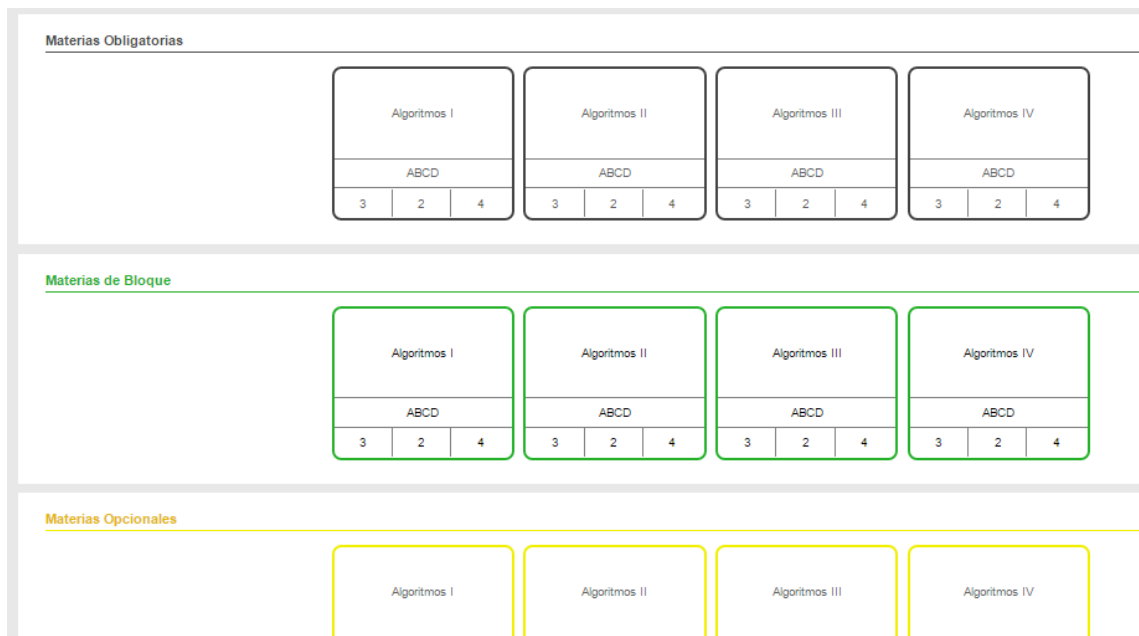


Figura 6.1: Resultado de JavaScript Orientado a Objetos.

Ingeniería Ambiental
 IAMB-2010-206

| | | | | | | | | |
|--|---|---|---|--|---|--|--|--------------|
| Química Inorgánica AEF-1060 3 2 5 | Química Analítica AEG-1059 3 3 6 | Fundamentos de Química Orgánica AEF-1033 3 2 5 | Análisis Instrumental AMF-1001 3 2 5 | Fenómenos de Transporte AEF-1027 3 2 5 | Taller de Investigación I ACA-0909 0 4 4 | Taller de Investigación II ACA-0910 0 4 4 | Seguridad e Higiene Industrial AMC-1021 2 2 4 | Especialidad |
| Cálculo Diferencial ACF-0901 3 2 5 | Álgebra Lineal ACF-0903 3 2 5 | Cálculo Vectorial ACF-0904 3 2 5 | Ecuaciones Diferenciales ACF-0905 3 2 5 | Desarrollo Sustentable ACD-0908 2 3 5 | Contaminación Atmosférica AMF-1003 3 2 5 | Potabilización de Agua AMG-1018 3 3 6 | Fundamentos de Aguas Residuales AMG-1012 3 3 6 | |
| Dibujo Asistido por Computadora AMA-1004 0 4 4 | Física AMF-1009 3 2 5 | Diseño de Experimentos Ambientales AMC-1005 2 2 4 | Balace de Materia y Energía AEF-1004 3 2 5 | Gestión Ambiental I AMF-1013 3 2 5 | Gestión Ambiental II AMC-1014 2 2 4 | Evaluación de Impacto Ambiental AMD-1008 2 3 5 | Residencia Profesional | |
| Taller de Ética ACA-0907 0 4 4 | Probabilidad y Estadística Ambiental AMF-1019 3 2 5 | Termodinámica AEF-1065 3 2 5 | Sistemas de Información Geográfica AMC-1022 2 2 4 | Mecánica de Fluidos AMF-1017 3 2 5 | Ingeniería de Costos AMC-1016 2 2 4 | Formulación y Evaluación de Proyectos AEF-1029 3 2 5 | | 10 |
| Fundamentos de Investigación ACC-0906 2 2 4 | Cálculo Integral ACF-0902 3 2 5 | Economía Ambiental AMP-1007 3 0 3 | Fisicoquímica I AMF-1010 3 2 5 | Fisicoquímica II AMF-1011 3 2 5 | Gestión de Residuos AMG-1015 3 3 6 | Remediación de Suelos AMG-1020 3 3 6 | Servicio Social | 10 |
| Biología AEF-1005 3 2 5 | Ecología AMF-1006 3 2 5 | Bioquímica AEJ-1007 4 2 6 | Microbiología AEM-1050 2 4 6 | Toxicología Ambiental AMF-1023 3 2 5 | Componentes de Equipo Industrial AMF-1002 3 2 5 | | Actividades Complementarias | 5 |
| 27 | 31 | 28 | 30 | 30 | 28 | 26 | 10 | 50 |

| | |
|-----------------------------|------------|
| Estructura Genérica | 210 |
| Residencia Profesional | 10 |
| Servicio Social | 10 |
| Actividades Complementarias | 5 |
| Especialidad | 25 |
| Total de Créditos | 260 |

Figura 6.2: Plan de Estudios Ing. Ambiental.

Ingeniería Bioquímica
 IBQA-2010-207

| | | | | | | | | |
|--|---|---|---|---|---|--|---|--|
| Fundamentos de Investigación ACC-0906 2 2 4 | Administración y Legislación de Empresas BQP-1001 3 0 3 | Cálculo Vectorial ACF-0904 3 2 5 | Programación y Métodos Numéricos BQF-1020 3 2 5 | Ingeniería Económica BQP-1015 3 0 3 | Operaciones Unitarias I BQJ-1017 4 2 6 | Taller de Investigación II ACA-0910 0 4 4 | Ingeniería de Proyectos BQC-1014 2 2 4 | Formulación y Evaluación de Proyectos AEF-1029 3 2 5 |
| Cálculo Diferencial ACF-0901 3 2 5 | Cálculo Integral ACF-0902 3 2 5 | Ecuaciones Diferenciales ACF-0905 3 2 5 | Electromagnetismo AEF-1020 3 2 5 | Fenómenos de Transporte I BQJ-1008 4 2 6 | Fenómenos de Transporte II BQJ-1009 4 2 6 | Operaciones Unitarias III BQJ-1019 4 2 6 | Ingeniería y Gestión Ambiental BQF-1016 3 2 5 | Especialidad 25 |
| Química AEF-1057 3 2 5 | Química Orgánica I BQF-1022 3 2 5 | Química Orgánica II BQF-1023 3 2 5 | Bioquímica AEJ-1007 4 2 6 | Bioquímica del Nitrógeno y Regulación Genética BQJ-1004 4 2 6 | Microbiología AEM-1050 2 4 6 | Operaciones Unitarias II BQJ-1018 4 2 6 | Ingeniería de Procesos BQF-1013 3 2 5 | Residencia Profesional 10 |
| Taller de Ética ACA-0907 0 4 4 | Biología AEF-1005 3 2 5 | Termodinámica AEF-1065 3 2 5 | Balance de Materia y Energía AEF-1004 3 2 5 | Fisicoquímica BQF-1011 3 2 5 | Seguridad e Higiene BQW-1024 2 0 2 | Ingeniería de Biorreactores BQF-1012 3 2 5 | | |
| Comportamiento Organizacional BQW-1006 2 0 2 | Química Analítica BQG-1021 3 3 6 | Física BQF-1010 3 2 5 | Análisis Instrumental BQF-1002 3 2 5 | Desarrollo Sustentable ACD-0908 2 3 5 | Cinética Química y Biológica BQF-1005 3 2 5 | | Servicio Social 10 | |
| Dibujo asistido por computadora AEO-1012 0 3 3 | Álgebra Lineal ACF-0903 3 2 5 | Estadística BQF-1007 3 2 5 | Aseguramiento de la Calidad BQQ-1003 1 2 3 | Instrumentación y Control AEF-1039 3 2 5 | Taller de Investigación I ACA-0909 0 4 4 | | Actividades Complementarias 5 | |
| 23 | 29 | 30 | 29 | 30 | 29 | 21 | 14 | 55 |

| | |
|-----------------------------|------------|
| Estructura Genérica | 210 |
| Residencia Profesional | 10 |
| Servicio Social | 10 |
| Actividades Complementarias | 5 |
| Especialidad | 25 |
| Total de Créditos | 260 |

Figura 6.3: Plan de Estudios Ing. Bioquímica.

Ingeniería Civil
 ICIV-2010-208

| | | | | | | | | |
|---|--|--|---|--|--|--|--|-----------------------------|
| Fundamentos de Investigación ACC-0906 2 2 4 | Álgebra Lineal ACF-0903 3 2 5 | Estática ICF-1014 3 2 5 | Fundamentos de la Mecánica de los Medios Continuos ICE-1016 3 1 4 | Mecánica de Materiales ICF-1024 3 2 5 | Análisis Estructural ICF-1004 3 2 5 | Análisis Estructural Avanzado ICF-1005 3 2 5 | Diseño Estructural de Cimentaciones ICC-1012 2 2 4 | Especialidad |
| Cálculo Diferencial ACF-0901 3 2 5 | Cálculo Vectorial ACF-0904 3 2 5 | Ecuaciones Diferenciales ACF-0905 3 2 5 | Métodos Numéricos ICC-1027 2 2 4 | Desarrollo Sustentable ACD-0908 2 3 5 | Instalaciones en los Edificios ICD-1021 2 3 5 | Diseño de Elementos de Concreto Reforzado ICF-1011 3 2 5 | Diseño de Elementos de Acero ICF-1010 3 2 5 | |
| Taller de Ética ACA-0907 0 4 4 | Probabilidad y Estadística ICC-1029 2 2 4 | Geología ICC-1017 2 2 4 | Mecánica de Suelos ICJ-1025 4 2 6 | Mecánica de Suelos Aplicada ICJ-1026 4 2 6 | Diseño y Construcción de Pavimentos ICG-1013 3 3 6 | Taller de Investigación II ACA-0910 0 4 4 | Formulación y Evaluación de Proyectos ICC-1015 2 2 4 | Residencia Profesional |
| Cálculo Integral ACF-0902 3 2 5 | Topografía ICT-1033 2 6 8 | Carreteras ICG-1006 3 3 6 | Maquinaria Pesada y Movimiento de Tierra ICC-1022 2 2 4 | Costos y Presupuestos ICC-1007 2 2 4 | Taller de Investigación I ACA-0909 0 4 4 | Abastecimiento de Agua ICJ-1001 4 2 6 | Alcantarillado ICC-1003 2 2 4 | |
| Software en Ingeniería Civil ICA-1031 0 4 4 | Materiales y Procesos Constructivos ICC-1023 2 2 4 | Tecnología del Concreto ICC-1032 2 2 4 | Dinámica ICF-1009 3 2 5 | Administración de la Construcción ICC-1002 2 2 4 | Hidrología Superficial ICC-1020 2 2 4 | | | Servicio Social |
| Dibujo en Ingeniería Civil ICM-1008 2 4 6 | Química AEC-1058 2 2 4 | Modelos de Optimización de Recursos ICC-1028 2 2 4 | Sistemas de Transporte ICC-1030 2 2 4 | Hidráulica Básica ICG-1018 3 3 6 | Hidráulica de Canales ICG-1019 3 3 6 | | | Actividades Complementarias |
| 28 | 30 | 28 | 27 | 30 | 30 | 20 | 17 | 50 |

| | |
|-----------------------------|------------|
| Estructura Genérica | 210 |
| Residencia | 10 |
| Servicio Social | 10 |
| Actividades Complementarias | 5 |
| Especialidad | 25 |
| Total de Créditos | 260 |

Figura 6.4: Plan de Estudios Ing. Civil.

Ingeniería Electromecánica
 IEME-2010-210

| | | | | | | | | |
|---|---|--|---|---|--|---|--|-----------------------------|
| Química AEC-1058 2 2 4 | Estática EME-1012 3 1 4 | Dinámica EME-1008 3 1 4 | Análisis y Síntesis de Mecanismos EME-1005 3 1 4 | Diseño de Elementos de Máquina EMF-1009 3 2 5 | Máquinas y Equipos Térmicos I EMC-1018 2 2 4 | Máquinas y Equipos Térmicos II EMC-1019 2 2 4 | Refrigeración y Aire Acondicionado EMF-1023 3 2 5 | Especialidad |
| Cálculo Diferencial ACF-0901 3 2 5 | Cálculo Integral ACF-0902 3 2 5 | Cálculo Vectorial ACF-0904 3 2 5 | Ecuaciones Diferenciales ACF-0905 3 2 5 | Mecánica de Fluidos EME-1020 3 1 4 | Sistemas y Máquinas de Fluido EMJ-1026 4 2 6 | Sistemas Eléctricos de Potencia EMF-1024 3 2 5 | Sistemas Hidráulicos y Neumáticos de Potencia EMJ-1025 4 2 6 | |
| Taller de Ética ACA-0907 0 4 4 | Desarrollo Sustentable ACD-0908 2 3 5 | Electricidad y Magnetismo EMC-1011 2 2 4 | Análisis de Circuitos Eléctricos de CD EMF-1004 3 2 5 | Análisis de Circuitos Eléctricos de CA EMF-1003 3 2 5 | Máquinas Eléctricas EMJ-1017 4 2 6 | Controles Eléctricos EMF-1006 3 2 5 | Ahorro de Energía EMJ-1002 4 2 6 | Residencia Profesional |
| Álgebra Lineal ACF-0903 3 2 5 | Metrología y Normalización AEC-1047 2 2 4 | Dibujo Electromecánico EMF-1007 3 2 5 | Termodinámica EME-1029 3 1 4 | Transferencia de Calor EME-1030 3 1 4 | Instalaciones Eléctricas EMF-1015 3 2 5 | Taller de Investigación II ACA-0910 0 4 4 | Ingeniería de Control Clásico EMJ-1014 4 2 6 | |
| Introducción a la Programación EMH-1016 1 3 4 | Tecnología de los Materiales EME-1028 3 1 4 | Procesos de Manufactura EMC-1022 2 2 4 | Mecánica de Materiales EMJ-1021 4 2 6 | | Taller de Investigación I ACA-0909 0 4 4 | Administración y Técnicas de Mantenimiento EMJ-1001 4 2 6 | Subestaciones Eléctricas EMF-1027 3 2 5 | Servicio Social |
| Fundamentos de Investigación ACC-0906 2 2 4 | Probabilidad y Estadística AEE-1051 3 1 4 | | Electrónica Analógica AEF-1021 3 2 5 | Electrónica Digital AEC-1022 2 2 4 | Diseño e Ingeniería Asistidos por Computadora EMC-1010 2 2 4 | | Formulación y Evaluación de Proyectos EMC-1013 2 2 4 | Actividades Complementarias |
| 26 | 26 | 22 | 29 | 22 | 29 | 24 | 32 | 50 |

| | |
|-----------------------------|------------|
| Estructura Genérica | 210 |
| Residencia Profesional | 10 |
| Servicio Social | 10 |
| Actividades Complementarias | 5 |
| Especialidad | 25 |
| Total de Créditos | 260 |

Figura 6.5: Plan de Estudios Ing. Electromecánica.

Ingeniería Industrial
 IIND-2010-227

| | | | | | | | | | |
|---|--|---|---|--|---|---|--|--------------|--|
| Fundamentos de Investigación ACC-0906 2 2 2 4 | Electricidad y Electrónica Industrial INC-1009 2 2 2 4 | Metrología y Normalización AEC-1048 2 2 2 4 | Procesos de Fabricación INC-1023 2 2 2 4 | Administración de Proyectos INR-1003 2 1 1 3 | Taller de Investigación I ACA-0909 0 4 4 4 | Taller de Investigación II ACA-0910 0 4 4 4 | Formulación y Evaluación de Proyectos AED-1030 2 3 3 5 | Especialidad | |
| Taller de Ética ACA-0907 0 4 4 4 | Propiedad de los Materiales INC-1024 2 2 2 4 | Álgebra Lineal ACF-0903 3 2 2 5 | Física INC-1013 2 2 2 4 | Gestión de Costos INC-1014 2 2 2 4 | Ingeniería Económica AEC-1037 2 2 2 4 | Planeación Financiera INC-1021 2 2 2 4 | Relaciones Industriales INC-1026 2 2 2 4 | | 31 |
| Cálculo Diferencial ACF-0901 3 2 2 5 | Cálculo Integral ACF-0902 3 2 2 5 | Cálculo Vectorial ACF-0904 3 2 2 5 | Algoritmos y Lenguajes de Programación INC-1005 2 2 2 4 | Administración de las Operaciones I INC-1001 2 2 2 4 | Administración de las Operaciones II INC-1002 2 2 2 4 | Planeación y Diseño de Instalaciones INC-1022 2 2 2 4 | Residencia Profesional | 10 | |
| Taller de Herramientas Intelectuales INH-1029 1 3 3 4 | Ingeniería de Sistemas INR-1017 2 1 1 3 | Economía AEC-1018 2 2 2 4 | Investigación de Operaciones I INC-1018 2 2 2 4 | Investigación de Operaciones II INC-1019 2 2 2 4 | Simulación INC-1027 2 2 2 4 | Sistemas de Manufactura INF-1028 3 2 2 5 | | | Servicio Social |
| Química INC-1025 2 2 2 4 | Probabilidad y Estadística AEC-1053 2 2 2 4 | Estadística Inferencial I AEF-1024 3 2 2 5 | Estadística Inferencial II AEF-1025 3 2 2 5 | Control Estadístico de la Calidad INF-1007 3 2 2 5 | Administración del Mantenimiento INC-1004 2 2 2 4 | Logística y Cadenas de Suministro INH-1020 1 3 3 4 | Actividades Complementarias | 5 | |
| Dibujo Industrial INN-1008 0 6 6 6 | Análisis de la Realidad Nacional INQ-1006 1 2 2 3 | Estudio del Trabajo I INJ-1011 4 2 2 6 | Estudio del Trabajo II INJ-1012 4 2 2 6 | Ergonomía INF-1010 3 2 2 5 | Mercadotecnia AED-1044 2 3 3 5 | Gestión de los Sistemas de Calidad INC-1015 2 2 2 4 | | | Taller de Liderazgo INC-1030 2 2 2 4 |

27

27

29

32

30

25

25

9

56

| | |
|-----------------------------|------------|
| Estructura Genérica | 204 |
| Residencia Profesional | 10 |
| Servicio Social | 10 |
| Actividades Complementarias | 5 |
| Especialidad | 31 |
| Total de Créditos | 260 |

Figura 6.7: Plan de Estudios Ing. Industrial.

Ingeniería en Sistemas Computacionales
 ISIC-2010-224

| | | | | | | | | |
|---|---|---|---|--|--|---|--|--|
| Cálculo Diferencial ACF-0901 3 2 5 | Cálculo Integral ACF-0902 3 2 5 | Cálculo Vectorial ACF-0904 3 2 5 | Ecuaciones Diferenciales ACF-0905 3 2 5 | Desarrollo Sustentable ACD-0908 2 3 5 | Lenguajes y Automatas I SCD-1015 2 3 5 | Lenguajes y Automatas II SCD-1016 2 3 5 | Programación Lógica y Funcional SCC-1019 2 2 4 | Inteligencia Artificial SCC-1012 2 2 4 |
| Fundamentos de Programación SCD-1008 2 3 5 | Programación Orientada a Objetos SCD-1020 2 3 5 | Estructura de Datos AED-1026 2 3 5 | Métodos Numéricos SCC-1017 2 2 4 | Fundamentos de Telecomunicaciones AEC-1034 2 2 4 | Redes de Computadoras SCD-1021 2 3 5 | Conmutación y Enrutamiento en Redes de Datos SCD-1004 2 3 5 | Administración de Redes SCA-1002 0 4 4 | Especialidad 25 |
| Taller de Ética ACA-0907 0 4 4 | Contabilidad Financiera AEC-1008 2 2 4 | Cultura Empresarial SCC-1005 2 2 4 | Tópicos Avanzados de Programación SCD-1027 2 3 5 | Taller de Base de Datos SCA-1025 0 4 4 | Administración de Base de Datos SCB-1001 1 4 5 | Taller de Investigación I ACA-0909 0 4 4 | Taller de Investigación II ACA-0910 0 4 4 | Residencia Profesional |
| Matemáticas Discretas AEF-1041 3 2 5 | Química AEC-1058 2 2 4 | Investigación de Operaciones SCC-1013 2 2 4 | Fundamentos de Base de Datos AEF-1031 3 2 5 | Simulación SCD-1022 2 3 5 | Graficación SCC-1010 2 2 4 | Programación Web AEB-1055 1 4 5 | 10 | |
| Taller de Administración SCH-1024 1 3 4 | Álgebra Lineal ACF-0903 3 2 5 | Sistemas Operativos AEC-1061 2 2 4 | Taller de Sistemas Operativos SCA-1026 0 4 4 | Fundamentos de Ingeniería de Software SCC-1007 2 2 4 | Ingeniería de Software SCD-1011 2 3 5 | Gestión de Proyectos de Software SCG-1009 3 3 6 | Servicio Social 10 | |
| Fundamentos de Investigación ACC-0906 2 2 4 | Probabilidad y Estadística AEF-1052 3 2 5 | Física General SCF-1006 3 2 5 | Principios Eléctricos y Aplicaciones Digitales SCD-1018 2 3 5 | Arquitectura de Computadoras SCD-1003 2 3 5 | Lenguajes de Interfaz SCC-1014 2 2 4 | Sistemas Programables SCC-1023 2 2 4 | Actividades Complementarias 5 | |
| 27 | 28 | 27 | 28 | 27 | 28 | 24 | 17 | 54 |
| | | | | | | | | Estructura Genérica 210 Residencia 10 Servicio Social 10 Actividades Complementarias 5 Especialidad 25 Total de Créditos 260 |

Figura 6.9: Plan de Estudios Ing. en Sistemas Computacionales.

Ingeniería en Tecnologías de la Información y Comunicaciones
 ITIC-2010-225

| | | | | | | | | |
|---|---|---|---|---|--|---|---|-----------------------------|
| Cálculo Diferencial ACF-0901 3 2 5 | Cálculo Integral ACF-0902 3 2 5 | Matemáticas Aplicadas a Comunicaciones TIE-1018 3 1 4 | Análisis de Señales y Sistemas de Comunicación TID-1004 2 3 5 | Fundamentos de Redes TIF-1013 3 2 5 | Redes de Computadora TIF-1025 3 2 5 | Redes Emergentes TIF-1026 3 2 5 | Administración y Seguridad de Redes TIF-1003 3 2 5 | Especialidad |
| Fundamentos de Programación AEF-1032 3 2 5 | Programación Orientada a Objetos AEB-1054 1 4 5 | Estructuras y Organización de Datos TID-1012 2 3 5 | Programación II TIB-1024 1 4 5 | Telecomunicaciones TIF-1029 3 2 5 | Programación WEB AEB-1055 1 4 5 | Desarrollo de Aplicaciones para Dispositivos Móviles AEB-1011 1 4 5 | Auditoría en Tecnologías de la Información TIC-1006 2 2 4 | |
| Matemáticas Discretas I TIF-1019 3 2 5 | Matemáticas Discretas II TIF-1020 3 2 5 | Administración Gerencial TIC-1002 2 2 4 | Matemáticas para la Toma de Decisiones TIF-1021 3 2 5 | Administración de Proyectos TIF-1001 3 2 5 | Desarrollo de Emprendedores TID-1010 2 3 5 | Desarrollo Sustentable ACD-0908 2 3 5 | Interacción Humano Computadora TIH-1016 1 3 4 | Residencia Profesional |
| Introducción a las TIC's TIP-1017 3 0 3 | Probabilidad y Estadística AEF-1052 3 2 5 | Fundamentos de Base de Datos AEF-1031 3 2 5 | Taller de Base de Datos AEH-1063 1 3 4 | Base de Datos Distribuidas TIF-1007 3 2 5 | Taller de Investigación I ACA-0909 0 4 4 | Taller de Investigación II ACA-0910 0 4 4 | Ingeniería del Conocimiento TIC-1015 2 2 4 | |
| Taller de Ética ACA-0907 0 4 4 | Contabilidad y Costos TIF-1009 3 2 5 | Electricidad y Magnetismo TIC-1011 2 2 4 | Circuitos Eléctricos y Electrónicos TID-1008 2 3 5 | Arquitectura de Computadoras TIC-1005 2 2 4 | Sistemas Operativos I AEC-1061 2 2 4 | Sistemas Operativos II AED-1062 2 3 5 | | Servicio Social |
| Fundamentos de Investigación ACC-0906 2 2 4 | | Álgebra Lineal ACF-0903 3 2 5 | Ingeniería de Software TIC-1014 2 2 4 | Taller de Ingeniería de Software TIC-1027 2 2 4 | Tecnologías Inalámbricas TIC-1028 2 2 4 | Negocios Electrónicos I TIC-1022 2 2 4 | Negocios Electrónicos II TIC-1023 2 2 4 | Actividades Complementarias |
| 26 | 25 | 27 | 28 | 28 | 27 | 28 | 21 | 50 |

| | |
|-----------------------------|------------|
| Estructura Genérica | 210 |
| Residencia | 10 |
| Servicio Social | 10 |
| Actividades Complementarias | 5 |
| Especialidad | 25 |
| Total de Créditos | 260 |

Figura 6.10: Plan de Estudios Ing. en Tecnologías de la Información y Comunicaciones.