



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN

**SISTEMA DE INGRESO POR MÉRITO
ACADÉMICO Y DIAGNÓSTICO INTEGRAL PARA
LA UNIVERSIDAD CENTRAL DE VENEZUELA**

Trabajo Especial de Grado presentado ante la ilustre
Universidad Central de Venezuela por

Br. Rinaldo Raffaele

Br. Romero Gary

Para optar al título de Licenciado en Computación

Tutores:

Prof. Rivas Robinson

Prof. Bottini Adrián

Caracas, octubre de 2016

UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN

ACTA

Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado “Sistema de Ingreso por Mérito Académico y Diagnóstico Integral para la Universidad Central de Venezuela” y presentado por los bachilleres: Br. Rinaldo Raffaele titular de la Cédula de Identidad V-20.783.373 y Br. Romero Gary titular de la Cédula de Identidad V-19.398.144, a los fines de optar al título de **Licenciado en Computación**, dejamos constancia de lo siguiente:

Leído como fue, dicho trabajo por cada uno de los miembros del jurado, se fijó el día 18 de Octubre de 2016, a las 3:00 PM horas, para que los autores lo defendieran en forma pública, lo que estos hicieron en la Sala PA 3 de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondieron a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar con la nota de 20 puntos.

En fe de lo cual se levanta la presente Acta, en Caracas el día 18 de Octubre de 2016.

Prof. Rivas Robinson

(Tutor)

Prof. Bottini Adrián

(Tutor)

Prof(a). Rodriguez Maria

(Jurado)

Prof. Blanco Jaime

(Jurado)

AGRADECIMIENTOS

A mis padres, pilares fundamentales de mi vida, en especial a mi madre Aura Teresa Martínez de Ranaldo, que, en ausencia física de mi padre, jugó ambos roles durante mi formación como persona y mi progreso como profesional, deseó tanto como yo la llegara con éxito a mi meta. Tenerla conmigo me dió fuerzas y motivos para alcanzar esta meta.

Agradezco especialmente a mis tutores Robinson Rivas y Adrián Bottini por brindarme su apoyo incondicional y la oportunidad de elaborar este Trabajo Especial de Grado junto con mi compañero.

A todas aquellas personas que formaron parte de mi vida, con las que aprendí y compartí grandes momentos, de las cuales algunas en especial han estado más involucradas y me brindaron su apoyo y ayuda en este proceso como lo son mis grandes amigos Yuary Pino, Oriel Hernández, Reynaldo Trujillo, Andrea Antunes, Miguel Acosta, Rafael Domínguez, Rocío Fernández, Brenda Benítez, Verónica Arráez, al igual que muchos otros compañeros universitarios, amigos, y no por menos, mi compañero de tesis Gary Romero, gracias por el apoyo y paciencia en este arduo y largo camino.

A mis compañeros de trabajo y jefes, que durante toda la jornada brindaron su apoyo y comprensión a la hora de realizar organizaciones de horarios y jornadas laborales.

A mis socio, primo y hermano, Daniel soto, el cual me motivo constantemente para culminar la carrera y seguir adelante junto a él.

A mi Universidad, la casa que vence la sombra, por permitirme ser parte de ella y ser testigo de mi crecimiento integral.

A todos, ¡Gracias!.

Ranaldo Raffaele

Principalmente le agradezco a Dios, por darme la fortaleza, la claridad y la salud y perseverancia necesaria para culminar con éxito esta vivencia extraordinaria.

A mis padres Yasmín Bautista y Roberto Romero, mi abuela Ramona Orellana, a mi tía Maribel Bautista y a mi pareja Katherine Balliache, quienes me ayudaron a vencer muchos obstáculos mediante el apoyo que me brindaron en todo momento para culminar la meta propuesta.

A mis tutores, los profesores Robinson Rivas y Adrián Bottini, por brindarnos la oportunidad de realizar este trabajo y por haber tenido la paciencia y comprensión necesaria para guiarnos en este arduo camino. Así como también a los profesores de la Escuela de Computación, especialmente a aquellos que me impartieron clases, por lograr transmitirme grandiosas enseñanzas.

A mis compañeros de estudio y amigos, algunos muy especiales a los cuales siempre recordaré y trataré de conservar la amistad. Destacando entre ellos a Andrea Antunes, Rafael Domínguez, Jhony Acevedo, Cesar Bianco, Txomin Gutiérrez, Emmanuel Galeano, Iraima Rodríguez, Félix Maiz, Royer Laya y Carlos Quijada por ofrecerme su incondicional y gran ayuda, así como sus buenos consejos e ideas, incluyendo a mi gran compañero de tesis Raffaele Ranaldo, con quien he compartido grandiosos momentos.

A todos, gracias por haber compartido conmigo esta etapa tan importante en mi vida.

Romero Gary

DEDICATORIA

A Dios, por darnos la fuerza y el coraje necesario para lograr cada una de las metas propuestas.

A nuestros padres y demás familiares y amigos por su apoyo incondicional. Este logro también es de ustedes.

Ranaldo Raffaele y Romero Gary

ÍNDICE

ÍNDICE DE ILUSTRACIONES	viii
ÍNDICE DE TABLAS	x
RESUMEN	xi
INTRODUCCIÓN	1
CAPÍTULO I – PROBLEMA DE INVESTIGACIÓN	3
1.1. Planteamiento del Problema.....	3
1.2. Objetivos de la Investigación.....	6
1.2.1. Objetivo General.....	6
1.2.2. Objetivos Específicos.....	6
1.3. Solución Propuesta.....	6
1.4. Justificación.....	10
1.5. Alcance.....	11
CAPÍTULO II – MARCO CONCEPTUAL	14
2.1. Antecedentes.....	14
2.2. Proceso de Ingreso.....	15
2.3. Universidad Central de Venezuela.....	18
2.4. Aplicación <i>Web</i>	20
2.4.1. Clasificación de las Aplicaciones <i>Web</i>	22
2.4.2. Arquitectura.....	23
2.4.3. Patrón de arquitectura Modelo-Vista-Controlador (MVC).....	26
2.4.4. Tecnologías de lado del Cliente.....	29
2.4.5. Tecnologías de lado del Servidor.....	36
2.4.6. Tecnologías de Manejo de Datos.....	38
2.4.7. Otras Tecnologías de Desarrollo.....	39
CAPÍTULO III – MARCO METODOLÓGICO	41
3.1. Metodologías convencionales.....	41
3.1.1. Cascada.....	41
3.1.2. Desarrollo iterativo y creciente (incremental).....	43
3.1.3. Espiral.....	45
3.1.4. Prototipo.....	47

3.1.5 Método AgilUs.....	48
CAPÍTULO IV – MARCO APLICATIVO	51
4.1. Iteración 1: Esquema General de la Solución.	52
4.2. Iteración 2: Diseño e Implementación del Modelo Relacional de la Base de Datos.....	56
4.3. Iteración 3: Registro de los Aspirantes.....	61
4.4. Iteración 4: Prueba Vocacional con el Cálculo y Publicación de los Resultados Obtenidos.....	70
4.5. Iteración 5: Validación de Recaudos y Publicación de la Selección de los Aspirantes.....	74
4.6. Iteración 6: Prueba Diagnóstica con el Cálculo y Publicación de los Resultados Obtenidos.....	81
4.7. Iteración 7: Módulo de Administración.	86
CAPÍTULO V - RESULTADOS	93
CONCLUSIONES.....	98
BIBLIOGRAFÍA.....	102

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Clasificación de las aplicaciones web.	22
Ilustración 2: Arquitectura Cliente/Servidor	24
Ilustración 3: Interacción – relación del patrón MVC.....	27
Ilustración 4: Comportamiento del patrón MVC.	28
Ilustración 5: Estructura semántica de HTML5	31
Ilustración 6: Sintaxis CSS3.....	32
Ilustración 7: Interfaz de TortoiseGit.	40
Ilustración 8: Ciclo de vida del Modelo en Cascada	42
Ilustración 9: Ciclo de vida del Método Incremental	43
Ilustración 10: Ciclo de vida del Método en Espiral	46
Ilustración 11: Ciclo de vida del Modelo en Prototipo.	47
Ilustración 12: Ciclo de vida del método AgilUs	50
Ilustración 13: Estructuras de la Base de Datos en MySQL creadas por Laravel.	57
Ilustración 14: Archivos de migraciones creadas.....	59
Ilustración 15: Archivos de migraciones para la tabla Usuarios.	60
Ilustración 16: Interfaz de inicio de sesión.....	62
Ilustración 17: Interfaz para crear cuenta.	63
Ilustración 18: Interfaz de registro exitoso.....	64
Ilustración 19: Correo enviado al aspirante para activar su registro.	64
Ilustración 20: Carga de notas académicas de forma automática.....	65
Ilustración 21: Listado de las notas académicas obtenidas del archivo PDF proporcionado.	66
Ilustración 22: Interfaz de carga de notas académicas de forma manual.	67
Ilustración 23: Ingreso de las carreras a las se postula el aspirante.....	68
Ilustración 24: Resumen de los datos proporcionados por el aspirante.....	69
Ilustración 25: Interfaz de inicio para la prueba vocacional.	71
Ilustración 26: Interfaz de prueba vocacional.	72
Ilustración 27: Interfaz de publicación de los resultados de los aspirantes seleccionados.....	73
Ilustración 28: Buscador de aspirantes.....	75
Ilustración 29: Interfaz del reporte con los datos registrados del aspirante seleccionado.....	76

Ilustración 30: Interfaz limpia para buscar un nuevo aspirante.....	77
Ilustración 31: Listado de los registros de los aspirantes seleccionados en la fase anterior por estado.....	78
Ilustración 32: Reporte con datos del registro de un aspirante y observaciones hechas por el operador.	79
Ilustración 33: Listado de aspirantes aprobados o no para la siguiente fase.	80
Ilustración 34: Interfaz de publicación de los resultados de los aspirantes seleccionados.	81
Ilustración 35: Interfaz de inicio para la prueba diagnóstica.....	83
Ilustración 36: Interfaz de prueba verbal.....	84
Ilustración 37: Interfaz de prueba lógica matemática.	85
Ilustración 38: Interfaz de publicación de los resultados de los aspirantes seleccionados.	86
Ilustración 39: Crear un nuevo periodo.....	87
Ilustración 40: Modificación y finalización de las fases del sistema.	88
Ilustración 41: Selección de materias de bachillerato por carrera ofertada por la universidad.....	89
Ilustración 42: Carga de cupos por carrera.....	90
Ilustración 43: Bitácora de validaciones.	91
Ilustración 44: Gráfico de aspirantes inscritos.	93
Ilustración 45: Gráfico de aspirantes inscritos por estados.	94
Ilustración 46: Gráfico de aspirantes admitidos por estados.....	94
Ilustración 47: Gráfico de relación admitidos e inscritos.....	95
Ilustración 48: Gráfico de aspirantes por fase del sistema.	96
Ilustración 49: Gráfico de carreras por facultad.....	967
Ilustración 50: Gráfico de cupos por facultad	968

ÍNDICE DE TABLAS

Tabla 1: Definición de las iteraciones de la aplicación <i>web</i>	52
Tabla 2: Herramientas seleccionadas	55
Tabla 3: Guía de Estilo.....	55

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

**SISTEMA DE INGRESO POR MÉRITO ACADÉMICO Y DIAGNÓSTICO INTEGRAL
PARA LA UNIVERSIDAD CENTRAL DE VENEZUELA**

Autores: Rinaldo Raffaele

Romero Gary

Tutores: Prof. Rivas Robinson

Prof. Adrián Bottini

Año: 2016

RESUMEN

Una de las funciones que debe realizar la Universidad Central de Venezuela es llevar a cabo el proceso de ingreso de nuevos estudiantes para los bachilleres que desean estudiar alguna de las carreras que ella ofrece. Actualmente, parte de este proceso en algunas de sus fases se ejecuta de forma manual y los estudiantes deben dirigirse a la institución, para realizar la mayoría de las actividades que involucran el proceso, entre ellas, presentar una prueba diagnóstica. Sin embargo, esto genera un gran gasto en la utilización de recursos y el costo de eso es muy alto para la universidad al igual que al estudiante, debido al traslado y los aranceles que debe cancelar para costear la misma. Con el objetivo de solventar estos inconvenientes, surgió este Sistema de Ingreso por Mérito Académico y Diagnóstico Integral para la Universidad Central de Venezuela, el cual fue desarrollado con tecnologías de información que permiten la automatización de los procesos realizados para la selección de estudiantes que quieren ingresar en la institución, abarcando así desde el registro de los aspirantes (estudiantes) hasta la realización de una prueba diagnóstica que permitirá conocer a los estudiantes preparados para recibir una formación de calidad y garantizar la prosecución de estudios con un egreso exitoso.

Palabras Claves: Sistema de Ingreso Integral, Mérito Académico y Diagnóstico, UCV, Método Incremental, Aplicaciones *Web*.

INTRODUCCIÓN

El proceso de ingreso es uno de los más importantes que tienen cada una de las diferentes grandes casas de estudio, tanto del país como del mundo. El sistema que utiliza el mismo, juega un rol fundamental, debido a que de él dependerá la elección y/o gestión sobre los nuevos aspirantes y futuros egresados de dichas casas, por lo cual se debería disponer de un sistema fácil de utilizar, al alcance de todos y que cumpla con las exigencias planteadas por la institución para dicho proceso.

El contexto sobre el cual se presenta este Trabajo Especial de Grado (TEG), es el de la Universidad Central de Venezuela en todas sus sedes alrededor del país, con cada una de sus respectivas Facultades y carreras. El proceso de ingreso que lleva a cabo la universidad actualmente, se divide principalmente en tres (3) grandes fases comprendidas por el registro de los aspirantes, presentación de test o mecanismo de evaluación y publicación de los resultados. No obstante, hay que destacar que el proceso de registro se realiza a través de internet, mientras que las otras fases, se realizan de manera presencial en cada una de sus sedes, lo que desencadena cierto número de problemas a nivel logístico y económico, tanto para la institución que debe disponer de los materiales para las pruebas, como para el estudiante que debe trasladarse hasta la universidad y pagar un monto estimado para presentar dichas pruebas.

En tal sentido, se hace necesario considerar la innovación de los mecanismos que abarcan el proceso de ingreso de la universidad, mediante el uso de tecnologías que permitan la automatización a través del desarrollo de un sistema que solvante la problemática suscitada, garantizando que la información sea procesada y almacenada de una forma más segura para agilizar los procesos, a fin de generar una respuesta inmediata al momento que sea requerida y cubra las necesidades de la institución.

Por tal motivo, este trabajo se enfoca fundamentalmente en el desarrollo de un Sistema de Ingreso por Mérito Académico y Diagnóstico Integral para la Universidad Central de Venezuela; y está estructurado de la siguiente manera:

El Capítulo I, contiene el planteamiento del problema que dio origen al presente trabajo, el objetivo general, los objetivos específicos, la solución propuesta y la justificación de la solución que se presenta.

En el Capítulo II, se muestra el marco conceptual, en el cual se presentan los fundamentos teóricos investigados que servirán de base para dar soporte al desarrollo de este trabajo.

El Capítulo III, describe las distintas metodologías que se pueden utilizar para el desarrollo de la aplicación permitiendo definir la metodología que se utilizará como guía para la ejecución de la solución propuesta.

En el Capítulo IV, se especifican las actividades que se llevaron a cabo para el desarrollo de la solución planteada.

El Capítulo V, refleja los resultados obtenidos durante el proceso de ejecución de la aplicación *web* propuesta.

Finalmente, se presentan las conclusiones, así como también la bibliografía y las referencias digitales consultadas para la elaboración de este trabajo.

CAPÍTULO I

PROBLEMA DE INVESTIGACIÓN

Este capítulo abarca el estudio de la problemática asociada a la gestión del proceso de selección por mérito académico y diagnóstico de los aspirantes que tienen como intención ingresar por primera vez a la Universidad Central de Venezuela (UCV), ya que parte de dicho proceso actualmente aún mantiene fases que se llevan de forma manual e involucran manipulación por parte del personal, lo que ocasiona gran utilización de recursos económicos, tiempo y personal, debido a que la universidad debe ofertar un proceso de registro, aplicación de evaluación y selección por cada aspirante que se postule.

1.1. Planteamiento del Problema

La UCV es una de las universidades con mayor población estudiantil e importancia productiva, que a través de sus años de trayectoria se ha convertido en una universidad prestigiosa por su alta calidad educativa, esto causa que el sueño de cientos de jóvenes estudiantes sea convertirse en un estudiante ucevista, razón por la cual, año tras años la institución recibe una alta demanda de aspirantes para ingresar.

Sin embargo, desde sus inicios la UCV ha llevado a cabo un proceso de ingreso que con los años ha ido cambiando, a fin de captar estudiantes del último año de la Educación Media, Diversificada y Profesional con talentos, capacidades y vocación para el ingreso al conocimiento, y así, contar con estudiantes preparados para recibir una formación de calidad y garantizar la prosecución de estudios con un egreso exitoso.

Actualmente, la universidad utiliza un proceso en el cual el estudiante una vez seleccionada el área de conocimiento, realiza un registro en línea indicando las carreras a las que quiere ingresar, con una selección máxima permitida de tres (3) opciones. Seguidamente, se lleva a cabo un proceso de validación de documentos, donde cada aspirante debe acudir físicamente a la Facultad correspondiente a su primera opción de carrera seleccionada, para que consigne y le verifiquen los documentos necesarios, a fin de obtener el comprobante de inscripción que incluye el número, fecha y horario al que debe acudir para la presentación de la evaluación pertinente.

No obstante, la realización de dicha evaluación al igual que el registro, se divide por áreas, y a su vez, se efectúa comúnmente a los fines de semana, lo que causa que esta fase tome alrededor de un mes de duración, dado que cada prueba toma varios días en llevarse a cabo aun cuando se define de uno (1) a tres (3) turnos diarios.

Una vez ocurrido lo anterior, todas las pruebas son recogidas en cada una de las Facultades para ser evaluadas a través de una máquina de lector óptico que dispone la universidad, y así, obtener de ellas un resultado que es publicado a fin de que el aspirante conozca si fue o no seleccionado para ingresar a la universidad. Sin embargo, hay que destacar que, aunque dichas correcciones se encuentran automatizadas, aun presentan pequeños problemas como lo son el traspapelado o atascamiento de una hoja, los errores humanos suscitados por parte del aspirante debido al incumplimiento de las normativas de llenado de las pruebas y los errores o problemas que podrían surgir durante la manipulación o traslados de esas evaluaciones dado que están expuestas a agentes externos.

Partiendo de este hecho y desde una perspectiva general, el proceso actual de selección de un aspirante genera un arduo trabajo, ya que se requiere del cumplimiento de una o varias fases, que, a su vez, cada una de ellas puede contar con diversos procesos, los cuales pueden o no tomar un resultado de una fase previa o/y generar otro resultado que puede ser utilizado por una fase posterior.

La gestión del proceso de ingreso del aspirante, puede ser dividida en dos fases importantes, la primera de ellas está relacionada con el período de registro de los aspirantes, y la segunda, es la fase de selección de los mismos; cada fase está compuesta por procesos y cada proceso es un conjunto de actividades que permiten que éste cumpla con su objetivo propio.

La fase de registro de los aspirantes tiene como finalidad la recolección de todos los datos y recaudos necesarios asociados al aspirante. Mientras que, la fase de selección de los aspirantes, tiene por fin la escogencia de los mejores candidatos, basada en los resultados obtenidos de las evaluaciones diagnósticas, verbales y lógicas que se les aplican a dichos aspirantes.

Llevar a cabo esas fases por cada periodo académico involucra la participación de ciertos elementos que son considerados claves dentro del proceso de ingreso, como lo son el uso de una gran cantidad de personas para ejecutar la logística, la recepción y validación de recaudos, la ejecución, recepción y revisión de las evaluaciones que se le aplican a los aspirantes, así como también, el gasto monetario para la universidad por la utilización de recursos materiales y personal para la realización de cada fase del proceso, por lo cual el aspirante se veía en la obligación de cancelar un arancel el cual es utilizado para cubrir dichos gastos, debido a que la universidad no dispone de un presupuesto destinado directamente al proceso de ingreso. sumado a la disposición y posibilidades de los aspirantes para movilizarse hasta la UCV a presentar las evaluaciones.

Todo esto da como resultado una gran cantidad de trabajo, ya que satisfacer la alta demanda de aspirantes se ha convertido en un reto continuo, tornando dicho trabajo en uno muy costoso, pesado y repetitivo porque al no contar con un sistema automatizado que agilice y haga más eficiente gran parte de este proceso de ingreso, además de generar altos gastos monetarios por materiales. Asimismo, este procesamiento manual no deja registrado un historial con los resultados obtenidos del proceso en cuanto a referencias e índices de los aspirantes que participan en el proceso.

Por tal motivo, este Trabajo Especial de Grado se basa fundamentalmente en solventar las dificultades que se presentan con el proceso de ingreso que dispone la universidad actualmente, empleando tecnología *web* para el desarrollo de una aplicación que permita proporcionar un excelente servicio a la comunidad estudiantil aspirante a ser un estudiante ucevista y hacer un uso eficiente de los recursos de la universidad tanto materiales como humanos.

1.2 Objetivos de la Investigación

1.2.1 Objetivo General

Desarrollar una aplicación *web* para la UCV que permita la gestión, organización, evaluación y selección de los aspirantes a ingresar en ella por cada Facultad, a fin de automatizar estos procesos cumpliendo con cada una de las etapas que se llevan a cabo actualmente de forma manual.

1.2.2 Objetivos Específicos

- Identificar los requerimientos funcionales y no funcionales del sistema.
- Definir las fases que conformarán todo el proceso de ingreso.
- Definir la arquitectura de tecnológica que dará soporte a toda la aplicación.
- Diseñar una base de datos sólida partiendo del análisis previo, que permita proporcionar de forma rápida y efectiva un acceso conveniente a la información que desea para obtener resultados precisos.
- Elaborar las pruebas y correcciones necesarias, tomando como criterio las consultas realizadas a los tutores y el departamento de secretaría, con la finalidad de verificar que el sistema cumpla con las expectativas deseadas.

1.3. Solución Propuesta

Para solucionar la problemática expuesta, se plantea el desarrollo de una aplicación *web* que permita el ingreso de los aspirantes que desean pertenecer a la UCV con criterios que preserve la igualdad de oportunidades con la calidad respectiva. Para ello se propone el diseño de un sistema de ingreso de acuerdo a las políticas académicas y a las capacidades, dimensiones y recursos de cada Facultad y carrera, de conformidad con las atribuciones que le confiere la Ley de Universidades vigente en cuanto a establecer sus propios mecanismos de ingreso, y con base en la autonomía para la organización de los procesos internos establecida en la Constitución Nacional.

Para lograr todo eso, se plantean tres (3) secciones que son establecidas de acuerdo al usuario de la aplicación. La primera sección es creada para los aspirantes (estudiantes que desean ingresar a la universidad), la segunda para los operadores (personal administrativo que recibe los recaudos en físico que se le solicitan al aspirante) y supervisor (personal administrativo que supervisa las tareas del operador y asiste las observaciones que dicho operador realice), y la tercera para el administrador (personal administrativo encargado de gestionar el proceso de ingreso). A continuación, se procede a definir las fases que componen las distintas secciones de la aplicación:

- **Fase de Registro:** Es lo primero en llevarse a cabo y comprende la sección del aspirante. Esta fase contempla un conjunto de procesos donde el aspirante ingresa a la aplicación y ésta le solicita que proporcione sus datos, tanto personales, como geográficos al igual su información académica (notas) y la información pertinente al plantel donde culminó sus estudios recientemente (localización, nombre del plantel). Posteriormente, se le consulta al aspirante por cuál de las cuatro (4) áreas de conocimiento que se presentan (Ciencias de la Salud, Ciencia y Tecnología, Ciencias Sociales y Políticas y Ciencias del Agro y del Mar) desea optar y cuáles son las carreras que desea cursar independientemente de la Facultad, siempre y cuando estas Facultades pertenezcan al área de conocimiento seleccionada. Una vez completado y enviados todos esos datos, la aplicación le mostrará al aspirante un resumen de toda la información suministrada.
- **Fase de Prueba Vocacional:** Este proceso involucra también solo al aspirante. Aquí, la aplicación le realizará un *test* vocacional al aspirante, permitiendo con ello evaluar su vocación con respecto al área de conocimiento seleccionada en el proceso de registro. Esto se hace con la finalidad de otorgar una mayor prioridad a los aspirantes que muestran una mayor afinidad a dicha área. Luego, partiendo de lo anterior, se hace una selección donde la aplicación calcula el promedio general de los aspirantes y los resultados de la prueba, asignado un promedio ponderado por cada opción seleccionada por el aspirante, para generar un listado con los calificados por carrera para continuar con la siguiente fase.
- **Fase de Validación:** La aplicación dispondrá de una sección administrada por un operador para realizar la función de comprobar la validez de los datos de los aspirantes en el sistema. Durante este proceso el aspirante deberá llevar los recaudos solicitados en físico a la universidad, ante el operador correspondiente a la Facultad asociada a la carrera que desea cursar, y en caso de haber discrepancias en la información, la aplicación le proporciona al

operador un espacio donde podrá colocar las observaciones asociadas al aspirante, sobre su proceso, hasta una verificación posterior que involucra la participación de un supervisor a través de la misma aplicación, para certificar y publicar la aprobación o rechazo de un aspirante registrado.

- **Fase de Prueba Verbal y Lógica Matemática (Diagnóstica):** En este proceso la aplicación realiza una evaluación por cada aspirante la cual consta de dos (2) partes, una verbal y una lógica matemática, cada una con sus propios modelos de evaluación. El aspirante deberá contestar un conjunto de preguntas por parte, para lo cual cuenta con un máximo de 90 minutos de duración para la ejecución de toda la evaluación, es decir, 45 minutos para la subprueba verbal y otros 45 minutos para la subprueba lógica matemática. Para este proceso se disponen varios bloques de horarios a fin de que la evaluación sea dinámica y no sea igual para todos los aspirantes implementando los distintos modelos de prueba disponibles.

Asimismo, esta fase posee un proceso de selección en donde la aplicación, dado los resultados de la prueba diagnóstica, genera una lista enumerada que indica los aspirantes que son admitidos en la carrera que escogieron por Facultad. En caso de empates, éstos se resolverían tomando en cuenta el promedio específico de la carrera. Por otra parte, al culminar esa selección, el sistema le permitirá al aspirante ingresar sus datos en la aplicación para saber si ha sido admitido ó no, para el periodo actual de ingreso a la universidad.

Adicionalmente, la aplicación contará con una sección de administración, en donde una persona administrativa cumple el rol de administrador dentro de dicha aplicación. Esta persona será la encargada de gestionar todo lo que involucra el proceso de ingreso a través de la aplicación ya que le permitirá crear nuevos modelos de evaluación verbal y lógica matemática, definir los tiempos para cada fase y proceso, así como también consultar los resultados obtenidos por los aspirantes a nuevo ingreso en la UCV y en base a eso tomar decisiones oportunas y necesarias que permitan mejorar todo este proceso.

Sin embargo, a fin de desarrollar toda la aplicación, esta solución será implementada mediante el desarrollo de una arquitectura cliente/servidor para lo cual se hará uso de un conjunto de herramientas y tecnologías como lo son:

- **Tecnologías de lado del Cliente:**

- **HTML5 (*HyperText Markup Language*, versión 5):** Actualmente es considerado un lenguaje estándar en lo referente al desarrollo de páginas *web*.
- **CSS3 (*Cascading Style Sheets*, versión 3):** Lenguaje de hojas de estilos creado para controlar el aspecto o presentación de las páginas *web*.
- **JavaScript:** Es un lenguaje que fue desarrollado por la compañía Netscape con el objetivo de agregar dinamismo a las páginas *web*.
- **AngularJS:** Es un *framework* MVC (Modelo, Vista, Controlador) basado en *JavaScript* para el desarrollo *web front-end* que permite crear aplicaciones SPA (*Single-Page Applications*).
- **JQuery:** Es una biblioteca rápida, pequeña y con funciones de JavaScript. Permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas *web*.
- **AJAX:** Es una técnica que se lleva a cabo mediante programas escritos en *JavaScript* la cual permite que un servidor y un navegador intercambien información, posiblemente en XML del lado del cliente para realizar peticiones asíncronas al servidor.
- **JSON:** Acrónimo de *JavaScript Object Notation*, es un formato de texto ligero para el intercambio de datos.

- **Tecnologías de lado del Servidor:**

- **PHP:** Su propia organización lo define como un lenguaje de programación rápido, flexible y pragmática de uso general el cual funciona del lado del servidor y originalmente diseñado para el desarrollo *web* de contenido dinámico.
- **Laravel:** Es un *framework* de código abierto para desarrollar aplicaciones y servicios *web* con PHP.

- **Tecnologías de Manejo de Datos:**

- **MySQL:** Es un sistema de gestión de bases de datos relacional, está considerada como la base datos *open source* más popular del mundo.

- **Otras Tecnologías a Utilizar en el Desarrollo**
 - **Git:** Es un sistema de control de versiones libre y de código abierto, está diseñado para manejar todo, desde pequeños proyectos hasta aquellos que son realmente grandes, todo con bastante rapidez, eficacia y sencillez.
 - **TortoiseGit:** Es una interfaz de *Windows Shell* para *Git* siendo de código abierto.

1.4. Justificación

Los procedimientos empleados por la UCV en la manipulación de los datos y el tratamiento de la información para el proceso de ingreso de nuevos estudiantes se podrían catalogar como poco actualizados y de larga duración.

En tal sentido, es necesario que la información sea procesada y almacenada de una forma más versátil y segura para agilizar los procesos, a fin de generar una respuesta inmediata al momento que sea requerida, mediante el desarrollo de un sistema que abarque las necesidades y una mayor relación con los requerimientos de la institución.

Por tal motivo, se considera necesario la innovación de los mecanismos que abarcan el proceso de ingreso de la universidad, ya que este proceso contempla fases que se realizan de forma manual, lo que conlleva a posibles errores y largos periodos de ejecución lo cual se complica debido a la gran demanda de aspirantes, sin mencionar que cada uno de esos aspirantes debe acudir físicamente al establecimiento para realizar cualquier parte del proceso.

Partiendo de lo anterior, se plantea como solución, la realización de una aplicación *web*, la cual permita la automatización del proceso de ingreso a la universidad, proporcionando al aspirante un sistema robusto en el cual podrá registrar sus datos, participar en todas las fases y obtener respuestas de las mismas, sin tener la necesidad de acudir físicamente a la UCV, exceptuando el caso de entrega y verificación de documentos, ya que la confirmación de éstos se hace mediante la comparación de la información suministrada al sistema por el aspirante con los recaudos originales solicitados en físico.

Asimismo, genera que la universidad disminuya los gastos monetarios por uso de materiales, ya que no habrá necesidad de imprimir las evaluaciones, por ejemplo; el tiempo de respuesta con el resultado que obtiene el aspirante en su evaluación se reduce considerablemente y no existirá la manipulación de datos manual. Además, toda la información referente al proceso de ingreso quedaría almacenada en una base de datos a fin de poder ser consultada a futuro a través de reportes e indicadores que refleje lo ocurrido en todos los procesos realizados.

Por tanto, se automatizará en su totalidad el proceso de registro, evaluación y obtención de resultados de los aspirantes, colocando a su alcance un sistema ordenado, de fácil acceso y rápida respuesta. También se desarrollará una parte administrativa del sistema donde el personal administrativo autorizado podrá gestionar todo lo relacionado a los tiempos de ejecución de cada fase del sistema, crear los modelos de evaluación y consultar los datos procesados en base a la información que proporciona cada aspirante y los resultados que éstos obtuvieron en las evaluaciones realizadas.

En consecuencia, al establecer este sistema contribuirá a mejorar la imagen actual de la universidad desde una perspectiva tecnológica y facilitando la participación de los estudiantes en los procesos de admisión, al tiempo que se logra comunicar de manera clara, confiable y oportuna la información sobre proceso de ingreso de los aspirantes a ser estudiantes ucevistas.

1.5. Alcance

El Sistema de Ingreso por Mérito Académico y Diagnóstico Integral se desarrollará exclusivamente para la UCV e incorporará una nueva modalidad como proceso de ingreso interno a esta Casa de Estudios. Será una aplicación *web* que permitirá una participación de carácter voluntario y estará dirigida principalmente a todos aquellos estudiantes que adquieren el título de Bachiller de la República Bolivariana de Venezuela en el mismo año lectivo que cuando se lleve a cabo el proceso de ingreso. Sin embargo, el sistema podrá admitir información de otros años anteriores, de acuerdo a la decisión de los organismos universitarios.

El sistema permitirá conocer los méritos académicos de los bachilleres que aspiran cursar alguna carrera de la UCV ya que la posibilidad de ingreso, se fundamenta en los méritos alcanzados durante el desempeño académico de 1^{ro} a 4^{to} año de bachillerato de los aspirantes.

La aplicación se ejecutará a nivel de toda la universidad, permitiendo que los aspirantes, dentro de un tiempo definido, ingresen sus datos personales y académicos, y a su vez, puedan seleccionar hasta un máximo de tres (3) carreras asociadas exclusivamente al área de conocimiento de su preferencia, siendo ésta seleccionada entre un conjunto de cuatro (4) áreas existentes, lo cual da un primer indicio de su interés individual.

Partiendo de lo anterior, la aplicación proporcionará la información adecuada que será procesada y ordenada de acuerdo a una tabla de ponderación para los diferentes ítems, acerca del desempeño académico previo del aspirante, así como un perfil de sus inclinaciones vocacionales derivado de las respuestas obtenidas de la prueba vocacional que debe realizar. Prueba que será planteada por un profesional de psicología de la universidad seleccionado por las autoridades competentes.

Una vez llegado a ese punto, el sistema realizará la preselección de aspirantes por cada Escuela basada en los resultados obtenidos de los procesos anteriores. Esos resultados serán publicados para hacerlo de conocimiento a los involucrados de manera que los aspirantes seleccionados pasen al proceso de verificación de datos, donde se contrastará la información proporcionada por ellos en la aplicación contra los soportes en físico de fuentes oficiales (Ministerio del Poder Popular para la Educación y liceos) que deberán entregar; proceso para el cual participará personal calificado de las Divisiones de Control de Estudios de las diferentes Facultades.

Si el proceso de verificación fue exitoso, el sistema procederá a realizarle una prueba diagnóstica vía *web* al aspirante preseleccionado. Para ello se aplicará dicha prueba a una cantidad de estudiantes que no supere el doble de los cupos ofertados por cada carrera. Esta prueba tiene una duración de 90 minutos, será completamente automatizada ya que se basará en un conjunto de modelos de pruebas que serán cargados en la base de datos, los cuales no podrán ser modificados.

Los resultados de dicha prueba serán procesados y ordenados, de acuerdo a las directrices de las autoridades. A partir de esto, se publicarán los resultados obtenidos y los aspirantes sabrán si fueron o no seleccionados para ingresar a la cohorte correspondiente de la UCV.

En general, el procesamiento de los datos a realizar en la aplicación consiste en el cálculo de un índice para cada aspirante conformado por el promedio general de notas de 1ro a 4to año de bachillerato con un peso del 30% y el promedio específico de las notas de tres asignaturas de 1ro a 4to año de bachillerato dependiendo de la carrera a la cual se aspire, con un peso del 40%, y el resultado de la prueba vocacional, con un peso de 30%. Este índice permitirá seleccionar la cantidad de aspirantes según el cupo establecido por carrera.

CAPÍTULO II

MARCO CONCEPTUAL

Todo trabajo intelectual debe comenzar con el establecimiento de un marco teórico, con una revisión bibliográfica, con el señalamiento de antecedentes, esto en razón de que el conocimiento científico tiene un carácter acumulativo, lo que significa que el investigador utiliza y fundamenta su trabajo en los hallazgos de otros investigadores. De esta manera consigue relacionar su investigación con los conocimientos existentes, contribuyendo a ampliar la comprensión de su área específica, hacia la correcta interpretación de nuevos problemas. Sostiene la Universidad Nacional Abierta (UNA, 1990): “Sería un error muy grande que un científico pretendiera ignorar los aportes hechos por otros investigadores. De esta manera el conocimiento científico no habría avanzado mucho por cuanto siempre se estaría recomenzando el camino”. Por esta razón, se presentan a continuación las bases teóricas necesarias que darán sustento a este trabajo de grado.

2.1 Antecedentes

De acuerdo con (Siguan, 1973), el crecimiento vertiginoso del alumnado universitario no es un fenómeno exclusivamente de un país, sino que en mayor o menor proporción ocurre en todo el mundo. Se ha calculado que, en los últimos quince años, en el conjunto de las universidades del mundo este crecimiento ha sido de un 10% anual con tendencia a aumentar en un futuro inmediato.

Las razones de este crecimiento son las mismas que explica el crecimiento de la educación en todos los niveles y que alimentan el llamado “boom” de la educación: la civilización técnica exige un mayor grado de conocimientos para iniciar cualquier ejercicio profesional, el desarrollo económico ofrece los medios para mantener a una parte importante de la población dedicada a prepararse para su ejercicio profesional y los ideales democráticos exigen que toda la población, al menos en teoría, pueda gozar de los beneficios de una educación superior.

Estas razones deben producir el establecimiento de una enseñanza básica unificada y generalizada. Pero de hecho producen, además, y generalmente antes, un crecimiento extraordinario de la enseñanza media. Asimismo, con este crecimiento acelerado de los alumnos que acaban la enseñanza media, se provoca a la vez, el crecimiento de alumnos universitarios.

Pero el crecimiento de las universidades, por fuerte que sea, no basta para absorber el crecimiento acelerado de la enseñanza media. Esta desproporción, comprobable a nivel mundial, aunado a la capacidad de recursos que disponen estas instituciones educativas, tanto privadas como públicas, explica el interés por los sistemas que contemplen un proceso de ingreso para la admisión en una universidad.

2.2. Proceso de Ingreso

Inicialmente se debe tener claro qué es un proceso de ingreso, pero antes, hay que definir qué es un proceso y un ingreso. De acuerdo con (Real Academia Española, 2016), un proceso es un conjunto de las fases sucesivas de un fenómeno natural o de una operación artificial, que también se puede entender como la acción de ir hacia delante. Mientras que un ingreso, según lo define la misma institución, es el acto de ser admitido en una corporación o de empezar a gozar de un empleo u otra cosa, así como la acción de entrar a algún lugar. Partiendo de esos dos conceptos claves se puede definir como proceso de ingreso, al conjunto de fases o actividades que se ejecutan para poder acceder a algo o a un lugar determinado.

Son muchos los casos donde se aplica un proceso de admisión, bien sea de forma manual o automatizada, de acuerdo con las necesidades de quien lo aplique, cumpliendo siempre con el o los reglamento(s) que se establecen para ello en caso de existir alguno o varios.

En la actualidad, uno de los lugares donde se aplica este proceso es en las universidades. Según la (Real Academia Española, 2016), una universidad es una institución de enseñanza superior que comprende de diversas facultades, y que confiere los grados académicos correspondientes que de acuerdo a las épocas y países.

No obstante, de acuerdo con (Definición ABC, 2015), en la mayor parte del mundo se dispone de instituciones universitarias de carácter público y de otras tantas de administración privada, regulada en todos los casos por organismos gubernamentales encargados del control de los estamentos educativos. Estas instituciones complementan a la etapa secundaria y tienen la enorme responsabilidad de formar profesionales.

El proceso de ingreso a las universidades consiste en la aceptación de una persona para iniciar estudios dentro de la institución permitiéndole la matriculación en el curso académico inicial de un programa a través de la selección de estudiantes que aprueban un conjunto de pruebas previas y/o requisitos definidos por dicha institución en conjunto con el cumplimiento de las normativas que establece cada país (Red Iberoamericana para la Acreditación de la Calidad de la Educación Superior, 2004).

Según (Siguan, 1973), una de las razones por las cuales se realiza la selección de estudiantes para el ingreso a una universidad, es la capacidad que posee el establecimiento universitario. Capacidad determinada por los elementos materiales de la enseñanza: capacidad de las aulas, bibliotecas y laboratorios, habitación en el caso de instituciones residenciales y también por el número de profesores y, en general, por la cantidad y calidad del equipo pedagógico.

En algunos países las universidades, ya sean instituciones privadas o públicas con autonomía, han favorecido esta limitación. Cada universidad acepta sólo la cantidad de alumnos que considera que puede educar eficazmente. Sin embargo, en todos los países, se consideran las previsiones sobre las necesidades del empleo dentro de la sociedad, las cuales orientan al crecimiento de las instituciones universitarias, aunque la capacidad de éstas sigue condicionando la admisión del estudiante y como el número de los aspirantes aumenta en mayor proporción que este crecimiento, son muchas las universidades que se ven obligadas a establecer procedimientos de selección cada vez más rígidos.

Unas universidades pueden hacerlo con relativa facilidad porque la selección forma parte de su política de estado, otras en cambio se resisten a introducirla, lo que causa un descenso en la calidad de la enseñanza debido a la congestión de estudiantes y con problemas insolubles de empleo para sus graduados. El desempleo de los graduados universitarios, se convierte así en el principal argumento en favor de la selección para la universidad en los países más reacios a ella.

En un informe preparado para el seminario del I.I.P.E., citado por el autor (Siguan) y en el que se resumen los datos de una encuesta realizada sobre una muestra representativa de todas las universidades del mundo, se indica que el 80% de las universidades utilizan sistemas de ingreso en la totalidad de sus facultades y escuelas, mientras el 20% restante sólo los utiliza en algunas facultades o en ninguna. Destacando que, en las universidades en que la limitación del alumnado sólo se aplica en algunas facultades, la de Medicina está siempre entre ellas y generalmente, en primer lugar.

El mecanismo selectivo puede apoyarse en:

- a.** Las calificaciones conseguidas a lo largo de la enseñanza media o en la prueba final que la termina.
- b.** Las calificaciones conseguidas en una prueba específica de ingreso a la enseñanza universitaria.
- c.** Una combinación de los dos procedimientos anteriores.

La valorización y ponderación de las calificaciones conseguidas en la educación media pueden hacerse de muy distintas maneras y cada universidad o cada país adopta su propia solución. Lo mismo puede decirse del contenido de la prueba para el ingreso y de la manera de combinar las calificaciones obtenidas de la educación media y las calificaciones de la prueba de ingreso.

En su gran mayoría, las instituciones universitarias muy solicitadas exigen una prueba de ingreso. Para esta prueba por lo general, se utiliza una selección gradual; sólo los que han conseguido cierta calificación en la enseñanza media pueden presentar una prueba inicial y sólo los que han conseguido cierto nivel de calificación en esta prueba pueden presentarse a la prueba de ingreso a la universidad.

Sin embargo, existen otras cuestiones más generales que hay que considerar. Cuando el proceso selectivo incluye una prueba de ingreso a la universidad, ésta puede concebirse más bien como un examen de aptitud. Por lo general, las universidades que insisten en valorar las aptitudes de los candidatos dan por supuesto que la posesión de los conocimientos mínimos está asegurada por las calificaciones obtenidas en la educación media y que lo que interesa a las universidades es valorar las aptitudes de los candidatos, bien sean sus aptitudes para aprovechar la enseñanza universitaria o sus aptitudes para el ejercicio profesional al que esta institución prepara, convirtiéndose esta prueba en una de orientación profesional.

Por otro lado, el autor destaca que es posible que el sistema de selección sea único para todos los centros que constituyen una universidad y son preferidos los candidatos que han recibido mejor puntuación en la prueba de selección. Sin embargo, muchas universidades utilizan procedimientos selectivos que en todo caso o en parte son distintos para las diversas facultades o centros de enseñanza.

La parte común, si existe, consiste en una valoración de las calificaciones de la educación media (sólo pueden aspirar a ingresar en la universidad los que han alcanzado ciertas calificaciones) o en una prueba de ingreso común a toda la universidad. No obstante, la parte específica propia de cada facultad puede consistir en establecer una determinada ponderación de las calificaciones recibidas en la educación media (por ejemplo, la Facultad de Ciencias puede dar más importancia a las calificaciones recibidas en las asignaturas de ciencias). O puede consistir, como es muy frecuente, en un examen distinto para cada facultad o para cada grupo de facultades a fin de seleccionar aquellos estudiantes que reúnan lo necesario para cursar alguna de las carreras que se imparten en la universidad.

Partiendo de todo lo anterior y siguiendo en el mismo contexto, dentro de Venezuela existen muchas universidades, pero entre todas ellas se destaca la universidad llamada Universidad Central de Venezuela, conocida por sus siglas UCV.

2.3. Universidad Central de Venezuela

Venezuela desde el 22 de diciembre del año 1721, se ha caracterizado por contar con una de las mejores casas de estudio en el mundo, la Universidad Central de Venezuela, una gran institución que tras múltiples gestiones por parte de los venezolanos, el Rey Felipe V concedió facultad al Seminario de otorgar grados mediante la Real Cédula y al año siguiente de su fundación, el Papa Inocencio XIII le otorgó carácter de Pontificia (UCV, 2009).

La UCV en un principio impartía clases de teología, medicina, filosofía y derecho exclusivamente en el idioma latín y actualmente, es una institución que cuenta con la mayor población estudiantil e importancia productiva entre las universidades de Venezuela y se reconoce como la más antigua del país.

Al decir Universidad Central de Venezuela, es sinónimo de buenos profesionales y esto se refleja en los reconocimientos que ha tenido a nivel internacional ya que ocupa el puesto número 18 en la lista de las mejores universidades de América Latina según un estudio realizado por la firma *QS World University Rankings* para el año 2016, además, la Ciudad Universitaria de Caracas, sede de la Universidad Central Venezuela, fue declarada Patrimonio Mundial, Cultural y Natural de la Humanidad por la UNESCO en el año 2000.

La UCV llamada "La casa que vence las sombras", celebra sus 294 años de fundada y actualmente la conforman 11 facultades, que son integradas por escuelas o carreras autónomas:

- Facultad de Agronomía
- Facultad de Arquitectura y Urbanismo
- Facultad de Ciencias
- Facultad de Ciencias Económicas y Sociales
- Facultad de Ciencias Jurídicas y Políticas
- Facultad de Ciencias Veterinarias
- Facultad de Farmacia
- Facultad de Humanidades y Educación
- Facultad de Ingeniería
- Facultad de Medicina
- Facultad de Odontología

No obstante, la UCV por ser una de las mejores casas de estudio, recibe año tras año una demanda de estudiantes que aspiran entrar a la universidad (aspirantes) que abarca más del total de estudiantes que se encuentran actualmente cursando alguna de las carreras que ofrece esta institución. Todo esto aunado a las limitaciones de recursos y otros aspectos como los que se mencionaron en puntos anteriores, causan la necesidad de realizar un proceso de ingreso para los aspirantes, a fin de captar aquellos que realmente reúnen los conocimientos y aptitudes requeridas para ingresar a esta universidad, considerando las políticas de ingreso basadas en el mérito académico y diagnóstico del conocimiento, orientadas hacia la selección con criterios que preserve la igualdad de oportunidades y la democratización con calidad.

Para ello, la universidad define el proceso de ingreso de acuerdo a las políticas académicas y a las capacidades, dimensiones y recursos de cada facultad y carrera, de conformidad con las atribuciones que le confiere la Ley de Universidades vigente en cuanto a establecer sus propios mecanismos de ingreso, y con base en la autonomía para la organización de los procesos internos establecida en la Constitución Nacional.

Basado en lo anterior, la UCV establece las distintas formas por las cuales un aspirante puede ingresar a la universidad, entre las que se destaca el proceso de ingreso mediante la aplicación de una prueba vocacional y otra de conocimiento, proceso que se realiza actualmente de forma manual, el cual consiste principalmente en preparar a un conjunto de personas quienes participan en la logística y aplican las pruebas, la impresión de todo el material necesario para dichas pruebas que deben realizar todos los aspirantes dentro de los espacios dispuestos para ello en la UCV; espacios conformados por una gran cantidad de aulas reservadas por facultad.

Con el pasar del tiempo, como se mencionó anteriormente, la cantidad de aspirantes fue creciendo cada vez más y más causando que el proceso de ingreso se volviera pesado, repetitivo, con altos gastos por la utilización de recursos, entre otros. Debido a esto, la UCV se ve en la necesidad de tratar de solventar esa problemática la cual cada año empeora más, mediante el planteamiento de convertir dicho proceso en uno automatizado gracias a los avances tecnológicos que con el tiempo se hicieron presentes y beneficiosos en cuanto a la realización de tareas; originando con ello, el desarrollo de este TEG a fin de atacar cada uno de los inconvenientes que presenta el proceso de ingreso actual tal como se explicó en el Capítulo I. No obstante, para automatiza dicho proceso se requiere investigar y tener claro ciertos conceptos, los cuales se especifican a continuación.

2.4. Aplicación Web

Con la llegada de Internet han surgido grandes cambios tecnológicos entre ellos el origen de las aplicaciones *web*. Este tipo de aplicaciones permiten usar la infraestructura de la *Web* (protocolos, lenguajes, entre otros) para desarrollar aplicaciones globales. Reportes recientes indican que las aplicaciones *web* representan más de la mitad del total de todas las aplicaciones de la industria de *software*. Esa cifra indica que las aplicaciones *web* siguen creciendo y ganando popularidad en un mercado muy competitivo, y que se perfilan a ser las aplicaciones del futuro (Tahuiton Mora, 2011).

En el contexto informático, según (Peñafiel, 2015) una aplicación *web* es un conjunto de aplicaciones que le permiten a los usuarios acceder a un servidor *web* a través de Internet o de una intranet mediante algún navegador *web* (Mozilla Firefox, Google Chrome, Opera, Internet Explorer, entre otros). Estas aplicaciones son un *software* codificado en un lenguaje entendible por los navegadores *web* a los cuales se les confía su ejecución. Entre sus principales características se pueden resaltar las siguientes:

- Permiten una mayor fluidez en la comunicación entre clientes y servidores, abstrayendo a los usuarios sobre complicados protocolos de comunicación.
- Su mantenimiento y actualización se realiza de manera sencilla a través de internet, evitando que se deba distribuir e instalar *software* específico o versiones individuales para cada usuario.
- Permiten al usuario un acceso interactivo a la información almacenada en el servidor, respondiendo a cada una de sus acciones con documentos codificados a través del estándar HTML, que, en conjunto con otros lenguajes interpretados de lado del cliente, permiten desplegar interfaces de usuario con el contenido solicitado.
- Consumen menor cantidad de recursos del lado del cliente.

De acuerdo a lo expresado por (Peñafiel, 2015) los beneficios de una aplicación *web* pueden ser resumidos de la siguiente forma:

- Permiten realizar trabajo a distancia con mayor facilidad.
- Utilizar una aplicación *web* es posible siempre y cuando se cuente con un dispositivo computacional que tenga un navegador *web* y conexión a internet. Además, no importa la hora o sitio donde el usuario se encuentre.
- Para su adecuado uso, no deberían ser requeridos grandes conocimientos en informática.
- Una aplicación *web* permite centralizar todas las áreas de trabajo en un solo sitio.

Del mismo modo, las ventajas de una aplicación *web* se pueden expresar en los siguientes términos (Peñafiel, 2015):

- Compatibilidad multiplataforma, es decir no importa el navegador o sistema operativo que el cliente utilice para acceder a la aplicación, esta funcionará siempre de la misma manera.

- Siempre se mantiene actualizada, aunque algunas características pueden funcionar adecuadamente o no dependiendo de la antigüedad del navegador con que se acceda a la aplicación *web*.
- No requieren ser descargadas, instaladas o configuradas para poder darles uso, en la mayor parte de los casos basta con simplemente crear una cuenta de usuario.
- Consumen menos recursos del computador ya que la mayor parte del procesamiento de los datos se realiza en el servidor, por lo que también suelen ocurrir menos fallos o errores.
- Pueden trabajar con múltiples usuarios al mismo tiempo, ofreciendo la misma experiencia de uso para todos, dependiendo de la capacidad o recursos del servidor *web*.

2.4.1. Clasificación de las Aplicaciones Web

De acuerdo con (Tahuiton Mora, 2011), las aplicaciones *web* no están limitadas a un dominio, es decir que se puede tener aplicaciones para todo tipo de necesidades.

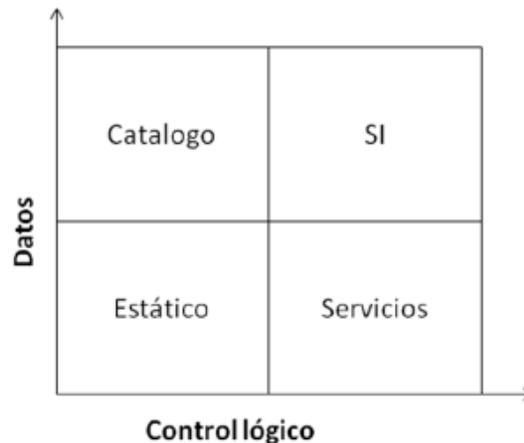


Ilustración 1: Clasificación de las aplicaciones web.

Fuente: (Tahuiton Mora, 2011).

Tal como se muestra en la Ilustración 1, existen cuatro tipos principales de aplicaciones *web* que son:

- **Estáticas:** Este tipo de aplicaciones representan a la primera generación y están compuestas por páginas *web* estáticas, imágenes y texto, pero no cuentan con una lógica de negocio. Dentro de este tipo de aplicaciones se encuentran las páginas personales.

- **Orientadas a Servicios:** Estas aplicaciones intentan ofrecer un servicio especializado, por lo cual implementan una lógica de negocio acorde al servicio ofrecido. Durante su mantenimiento los desarrolladores necesitan comprender claramente la lógica de negocio. Un ejemplo de este tipo de aplicaciones son los servidores de correo electrónico.
- **De Datos:** Este tipo de aplicaciones está enfocado en proveer una interfaz para acceder a una gran cantidad de datos y no en la lógica de negocio, por lo tanto, los desarrolladores necesitan comprender el flujo de datos. Un ejemplo de este tipo de aplicaciones son los catálogos en línea de las bibliotecas.
- **Sistemas de Información:** combinación de las aplicaciones orientadas a servicios y de datos. Los desarrolladores necesitan comprender claramente el flujo de datos y a su vez la lógica de negocio (especialmente en la manipulación de los datos). Ejemplos de este tipo de aplicaciones son la banca en línea y los portales de comercio electrónico.

Sin embargo, para llevar a cabo el desarrollo de cualquiera de estas aplicaciones *web*, hay que tener en cuenta ciertas consideraciones, como, por ejemplo, definir la arquitectura sobre la cual se va a trabajar.

2.4.2. Arquitectura

Según (Almaraz Hernández, Campos Cantero, & Castelo Delgado, 2011) desde hace algún tiempo, los sitios *web* tradicionales que se limitaban a mostrar información se han convertido en aplicaciones capaces de una interacción más o menos sofisticada con el usuario. Inevitablemente, esto ha provocado un aumento progresivo de la complejidad de estos sistemas y, por ende, la necesidad de buscar opciones de diseño nuevas que permitan dar con la arquitectura óptima que facilite la construcción de los mismos.

De acuerdo con (Solórzano, 2004), para que la exploración de un sitio sea placentera o frustrante depende mucho de su estructura, de la claridad de sus jerarquías de información y de la relevancia y versatilidad de sus hipervínculos. Para navegar con éxito, los usuarios deben saber dónde están, confiar en su meta y saber que pueden regresar con facilidad. Esto depende de una exitosa organización de la información y de hacerla clara mediante el diseño de controles de navegación.

Asimismo, el usuario interactúa con las aplicaciones *web* a través del navegador. Como consecuencia de la actividad del usuario, se envían peticiones al servidor, donde se aloja la aplicación y que normalmente hace uso de una base de datos que almacena toda la información relacionada con la misma. El servidor procesa la petición y devuelve la respuesta al navegador para que el usuario la visualice (Almaraz Hernández, Campos Cantero, & Castelo Delgado, 2011).

Partiendo de esto, surgen un conjunto de arquitecturas, entre las cuales podemos encontrar la arquitectura cliente/servidor.

Desde el punto de vista funcional, se puede definir el modelo cliente/servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma (Márquez Avendaño & Zulaica Rugarcía, 2004).

En dicho modelo, el rol del cliente es interpretado por aquel dispositivo encargado de realizar peticiones solicitando un determinado servicio a través de un navegador *web*, mientras que el servidor se encarga de enviar uno o varios mensajes con la respuesta pertinente a dicha solicitud. En algunos casos el servidor no solo interactúa con los clientes, sino que también lo puede hacer con otros servidores, ya sea para solicitar servicios externos o gestionar los datos del propio sistema, tal como se muestra en la Ilustración 2.

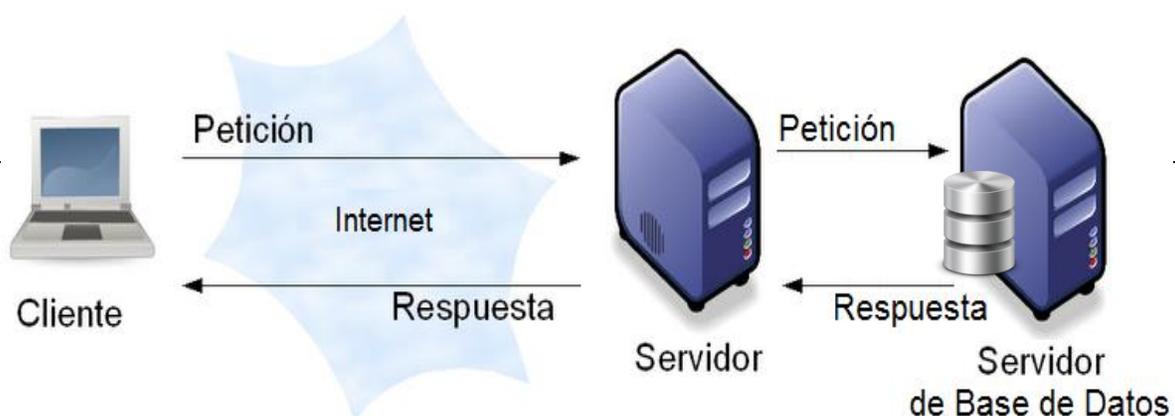


Ilustración 2: Arquitectura Cliente/Servidor

Según lo expresa (Luján Mora, 2001), esta arquitectura implica la existencia de una relación entre procesos que solicitan servicios (clientes) y procesos que responden a estos servicios (servidores). Destacando que normalmente, los servidores son computadoras potentes dedicadas a gestionar unidades de disco (servidor de archivos), impresoras (servidor de impresoras), tráfico de red (servidor de red), datos (servidor de bases de datos) o incluso aplicaciones (servidor de aplicaciones), mientras que los clientes son máquinas menos potentes y usan los recursos que ofrecen los servidores.

También afirma que la arquitectura cliente/servidor implica la realización de aplicaciones distribuidas. La principal ventaja de esta arquitectura es que permite separar las funciones según su servicio, permitiendo situar cada función en la plataforma más adecuada para su ejecución. Además, también presenta las siguientes ventajas (Luján Mora, 2001):

- Las redes de computadoras permiten que múltiples procesadores puedan ejecutar partes distribuidas de una misma aplicación, logrando concurrencia de procesos.
- Existe la posibilidad de migrar aplicaciones de un procesador a otro con modificaciones mínimas en los programas.
- Se obtiene una escalabilidad de la aplicación. Permite la ampliación horizontal o vertical de las aplicaciones. La escalabilidad horizontal se refiere a la capacidad de añadir o suprimir estaciones de trabajo que hagan uso de la aplicación (clientes), sin que afecte sustancialmente al rendimiento general. La escalabilidad vertical permite la migración hacia servidores de mayor o menor capacidad y velocidad o de un tipo diferente.
- Posibilita el acceso a los datos independientemente de donde se encuentre el usuario.

De acuerdo a lo expresado por (Márquez Avendaño & Zulaica Rugarcía, 2004) tanto el cliente como el servidor poseen una serie de características propias, las cuáles se relacionan directamente a sus funcionalidades principales, como se describe a continuación:

- **Cliente:** El cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor, se le conoce también con el término *front-end*. El cliente se encarga principalmente de las funciones de manipulación y despliegue de los datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario, así como acceder a los servicios distribuidos en cualquier parte de la red. En resumen, sus funciones esenciales son:

- Interacción con el usuario.
 - Procesar la lógica de la aplicación y hacer validaciones locales.
 - Generar requerimientos de bases de datos.
 - Recibir resultados del servidor.
 - Formatear resultados, en elementos que puedan ser entendidos por el usuario final.
- **Servidor:** Es el proceso encargado de atender a múltiples clientes que hacen peticiones sobre algún recurso administrado por él, también se le conoce con el término *back-end*. Maneja normalmente todas las funciones relacionadas a la gran parte de las reglas del negocio y recursos de datos. Sus funciones pueden resumirse de la siguiente forma:
 - Aceptar los requerimientos de bases de datos que hacen los clientes.
 - Procesar requerimientos de bases de datos.
 - Formatear los datos para transmitirlos a los clientes.
 - Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

Teniendo en cuenta estas características en la arquitectura de las aplicaciones *web*, existen un conjunto de patrones de diseño de aplicación básica que facilitan el diseño para este tipo de sistemas. Uno de los patrones que ha demostrado ser fundamental a la hora de diseñar aplicaciones *web* es el patrón Modelo-Vista-Controlador.

2.4.3. Patrón de arquitectura Modelo-Vista-Controlador (MVC)

Un patrón se define como aquella solución a un problema que dado un contexto y ciertas características es considerada como exitosa. Un patrón de arquitectura plantea por tanto un modelo de estructura para las diferentes partes de un todo, como en el caso de las aplicaciones *web*, se refiere a la organización de los archivos de código fuente, de configuraciones, imágenes, entre otros (Bascón Pantoja, 2004).

Para (Almaraz Hernández, Campos Cantero, & Castelo Delgado, 2011), el patrón MVC propone la separación en distintos componentes de la interfaz de usuario (vistas), el modelo de negocio y la lógica de control.

A grandes rasgos, una vista es una “fotografía” del modelo (o una parte del mismo) en un determinado momento. Un control recibe un evento disparado por el usuario a través de la interfaz, accede al modelo de manera adecuada a la acción realizada, y presenta en una nueva vista el resultado de dicha acción. Por su parte, el modelo consiste en el conjunto de objetos que modelan los procesos de negocio que se realizan a través del sistema.

Sin embargo, (Bascón Pantoja, 2004) expresa que este patrón considera dividir una aplicación en tres módulos claramente identificables y con funcionalidad bien definidas, siendo estos los siguientes:

- **El Modelo:** Es un conjunto de clases que representan la información del mundo real que el sistema debe procesar. No toman en cuenta la forma en que la información va a ser desplegada, ni el mecanismo mediante el cual será almacenada la información contenida dentro del mismo. Son la interfaz principal que utiliza la aplicación para comunicarse con las bases de datos.
- **La Vista:** Son el conjunto de clases que se encargan de mostrar la información contenida dentro de los modelos.
- **El Controlador:** Es un objeto que se encarga de dirigir el flujo de control de la aplicación debido a mensajes externos, como datos introducidos u opciones del menú seleccionadas por el usuario. El controlador se encarga de modificar los modelos y abrir o cerrar vistas.

En la Ilustración 3, que se muestra a continuación, es posible observar la interacción y relación a nivel general, entre los módulos de este patrón.

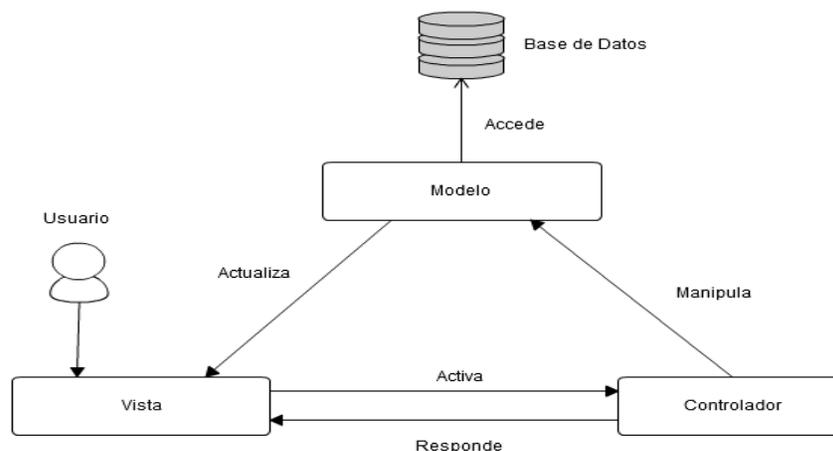


Ilustración 3: Interacción – relación del patrón MVC

Fuente: (Bascón Pantoja, 2004).

En una aplicación *web*, de acuerdo con (Almaraz Hernández, Campos Cantero, & Castelo Delgado, 2011), las vistas serían las páginas HTML que el usuario visualiza en el navegador. A través de estas páginas el usuario interactúa con la aplicación, enviando eventos al servidor a través de peticiones HTTP. En el servidor se encuentra el código de control para estos eventos, que en función del evento concreto actúa sobre el modelo convenientemente. Los resultados de la acción se devuelven al usuario en forma de página HTML mediante la respuesta HTTP (Ilustración 4).

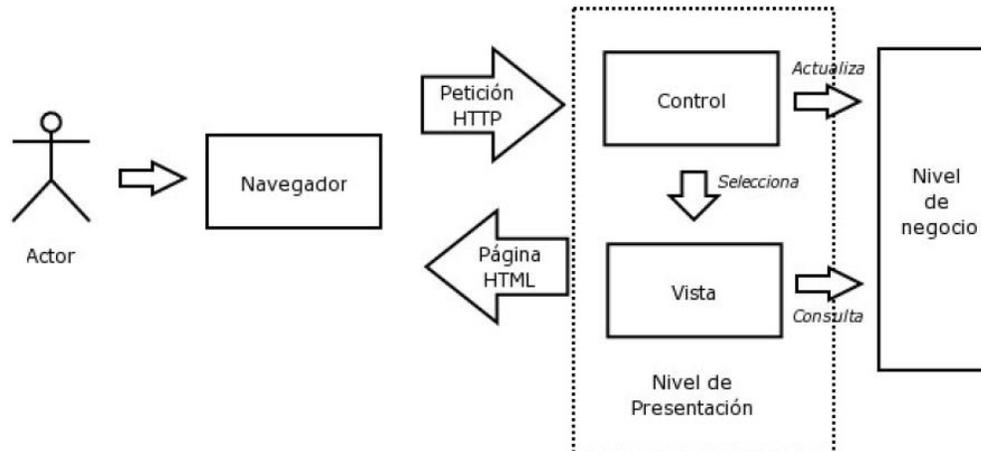


Ilustración 4: Comportamiento del patrón MVC.

Fuente: (Almaraz Hernández, Campos Cantero, & Castelo Delgado, 2011)

En base a lo explicado por los autores citados anteriormente, la clave de este patrón está en la separación entre vista y modelo. El modelo suele ser más estable a lo largo del tiempo y menos sujeto a variaciones, mientras que las vistas pueden cambiar con frecuencia, ya sea por cambio del medio de presentación (por ejemplo, HTML a WAP o a PDF) o por necesidades de usabilidad de la interfaz o simple renovación de la estética de la aplicación. Con esta clara separación las vistas pueden cambiar sin afectar al modelo y viceversa. Los controladores son los encargados de hacer de puente entre ambos, determinando el flujo de salida de la aplicación (qué se ve en cada momento).

Entre las ventajas de utilizar este patrón se encuentran las siguientes (Bascón Pantoja, 2004):

- La aplicación se implementa modularmente.
- Sus vistas siempre muestran información actualizada, pues la obtienen a través de los propios modelos, quienes se encargan de automáticamente de este proceso.
- Es considerado un patrón de diseño bien elaborado, pues las aplicaciones donde se implementan presentan una gran extensibilidad y mantenibilidad en comparación con las que se basan en otros patrones.

En cuanto a las desventajas o limitaciones del patrón se tienen las siguientes (Bascón Pantoja, 2004):

- El tiempo requerido para desarrollar una aplicación siguiendo este patrón es mayor durante la fase inicial que en aquellas que no la implementan, debido a que requiere que el programador defina una gran cantidad de clases, sin embargo, esta característica es la que en una fase más avanzada la aplicación sea fácilmente extensible y perdurable.
- Este patrón se basa en la programación orientada a objetos, por lo que su implementación en entornos que no sigan este paradigma es sumamente costosa.

No obstante, para llevar a cabo el desarrollo de cualquiera de estas aplicaciones *web*, haciendo uso de la arquitectura y patrón antes expuestos, es necesario hacer uso de una serie de herramientas y tecnologías las cuales serán descritas a continuación.

2.4.4. Tecnologías de lado del Cliente

Comprende todas aquellas tecnologías que permiten el despliegue y funcionamiento de la aplicación *web*. En estas aplicaciones su visualización y ejecución es posible mediante un cliente *web* (navegador), como es el caso de Mozilla Firefox, Google Chrome o cualquier otro disponible en el dispositivo donde se desee acceder a la aplicación.

Para su desarrollo, se hace uso de un conjunto de herramientas como *block* de notas o un entorno de desarrollo integrado (IDE). Dentro de estas herramientas se programa la aplicación, mediante el uso de tecnologías *web* compatibles con la mayoría de los clientes *web* y que actualmente son ampliamente conocidas, por ejemplo: HTML5, CSS3, JavaScript, entre otras. A continuación, se consideran algunas de las tecnologías *web* antes mencionadas.

- **HTML5 (*HyperText Markup Language*, versión 5):** Es el lenguaje con el que se escribe la estructura y la semántica del contenido de un documento *web* a partir del uso de etiquetas. Actualmente es considerado un lenguaje estándar en lo referente al desarrollo de páginas *web*. La quinta versión es definida por (Walker, 2011) como un lenguaje cuyo principal objetivo es estandarizar la creación de interfaces de usuario, facilitando de esta manera el diseño y desarrollo de aplicaciones *web*. Además, indica que HTML5 proporciona:
 - Etiquetas que describen exactamente lo que están diseñadas a contener.
 - Formularios mejorados, que reducen la necesidad de utilizar código JavaScript.
 - Un API basado en SQL que permite almacenamiento a nivel local para el cliente.

De acuerdo con (W3Schools, 2012), HTML5 mejora la interoperabilidad y reduce los costes de desarrollo, mediante reglas precisas sobre la forma de manejar todos los elementos HTML, y cómo recuperarse de errores. Asimismo, expone que algunas de las nuevas características de HTML5 son funciones de audio para la incrustación, vídeo, gráficos, almacenamiento de datos del lado del cliente, y los documentos interactivos.

En la Ilustración5, se puede observar algunas etiquetas que conforman la estructura semántica que se definieron para esta versión de HTML.

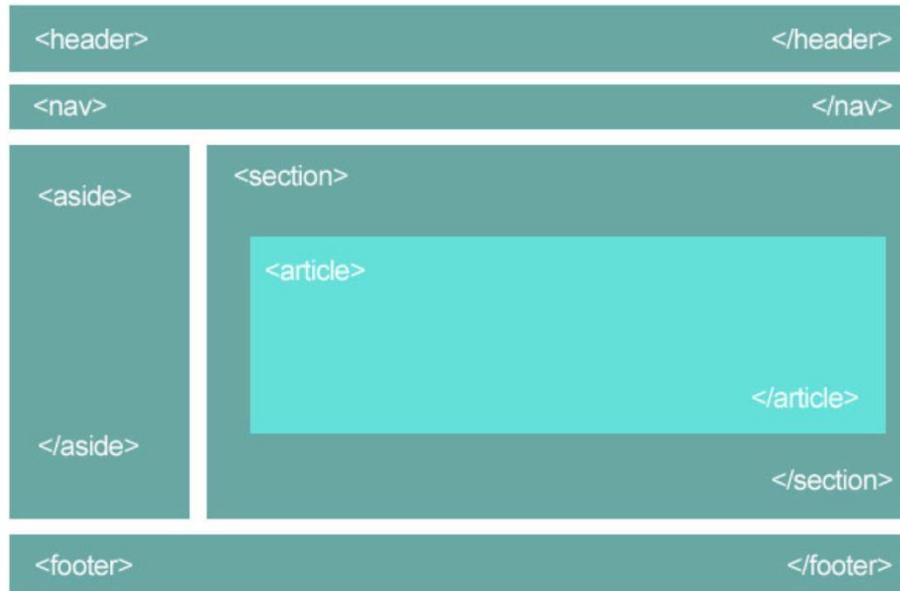


Ilustración 5: Estructura semántica de HTML5

Fuente: <http://lprogramacionii.blogspot.com/2015/02/estructura-html-y-etiquetas-html5.html>

- **CSS3 (*Cascading Style Sheets*, versión 3):** CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS nace con la filosofía de separar de mejor forma los contenidos y su presentación y es imprescindible para crear páginas *web* complejas. Este lenguaje fue publicado como un estándar de la W3C en el año 1996, y hasta la fecha cuenta con tres versiones (W3C, 2012).

Además, ofrecen la posibilidad de definir reglas y estilos de representación en dispositivos, ya sean con la capacidad de desplegar contenido *web*. A partir del año 2005 se empezó a definir la versión conocida como CSS3 (Damián, 2010).

Según explica (W3C, 2012), la sintaxis del lenguaje consta de elementos llamados reglas, las cuales se dividen en selectores y declaraciones, donde el selector es el identificador que se le quiere dar a la declaración de un conjunto de propiedades con sus valores y las propiedades son similares a los atributos de las etiquetas en HTML (Ilustración 6).



Ilustración 6: Sintaxis CSS3.

Fuente: <http://www.hongkiat.com/blog/keep-css3-markup-slim/>

Algunas de las funcionalidades que destacan de esta versión son:

- Modelo de cuadro.
 - Fondos y bordes.
 - Los valores de imagen y contenido sustituido.
 - Efectos de texto y animaciones.
 - 2D / 3D Transformaciones.
 - Disposición de columnas múltiples.
 - Interfaz de usuario.
-
- **JavaScript:** Es un lenguaje de script, es decir, se trata de código de programación que se inserta dentro de un documento HTML. Es interpretado por los navegadores permitiendo agregar dinamismo al contenido de un sitio *web*. Fue desarrollado por la compañía Netscape como parte del desarrollo de su navegador Navigator (Sánchez, 2003).

Según destaca Sánchez, contrario a lo que se interpreta, JavaScript no guarda relación alguna con el lenguaje Java, pues mientras este último se utiliza en la construcción de programas o aplicaciones independientes, mientras que JavaScript a pesar de ser más sencillo, depende de otros lenguajes como HTML para cumplir sus funciones, es decir que por sí solo no permite construir aplicaciones que trabajen de forma independiente.

Entre las muchas características que posee el lenguaje, se hace mención de aquellas consideradas como las más interesantes: Es orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico; como su principal inspiración fue el lenguaje C, comparte diversas estructuras con el mismo, como lo son condicionales, bucles, definición de variables, entre otros.

Su dinamismo deriva de las variables, ya que estas no se asocian por nombre sino por valor, por lo que una misma variable puede adoptar diversos tipos de datos a lo largo del mismo ciclo de ejecución.

Actualmente existen una gran cantidad de librerías basadas en este lenguaje como lo son backbone.js, three.js, angular.js, entre muchos otros, sin embargo, JQuery es probablemente una de las más sencillas y populares.

Con la llegada de HTML5, las capacidades ofrecidas por este lenguaje han aumentado, derivándose diversas tecnologías dedicadas a implementar aplicaciones más estructuradas de lado del cliente, con el objetivo de lograr una integración más sencilla con las tecnologías utilizadas de lado del servidor.

- **AngularJS:** Es un *framework* MVC (Modelo, Vista, Controlador) de código abierto desarrollado por Google y basado en JavaScript para el desarrollo *web front-end* (Flores, 2014).

Su primera versión de 2009 es mantenido por Google. Este *framework* trabaja del lado del cliente y adapta y amplía el HTML tradicional para servir mejor contenido dinámico a través de un enlace *de datos* bidireccional que permite la sincronización automática de modelos y vistas, trabajando de la mano con otras tecnologías, así como librerías de terceros.

Según expresa Flores, el equipo de AngularJS define este *framework* como: “Un *framework* estructural para páginas *web* dinámicas”. Resaltando que, un *framework* permite estructurar, organizar y escribir código de una manera más eficiente y en menos tiempo, haciéndolo más rápido de acuerdo a la manera en la que evolucionan los motores de *render* de los navegadores.

AngularJS permite realizar aplicaciones de tipo SPA (*Single Page Applications*), lo que significa que podemos construir una aplicación *web* en donde una parte de la misma cargue dinámicamente, sin que se tenga que recargar toda la página. Esto permite hacer una aplicación *web* más rápida y fácil.

- **JQuery:** Es una librería rápida y pequeña basada en el lenguaje JavaScript. Sus características hacen más sencilla la manipulación de los elementos definidos dentro de un documento en formato HTML, así como el manejo de eventos, animaciones e implementación de AJAX. Es compatible con una gran cantidad de navegadores, además es extensible y versátil, pues simplifica la sintaxis con la que debe escribirse código JavaScript (jQuery, 2015).

jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. Sin embargo, este *framework* se caracteriza por (Villarreal Fuentes, 2013):

- Selección de elementos.
 - Eventos.
 - Manipulación de la hoja de estilos CSS.
 - Efectos y animaciones.
 - Animaciones personalizadas.
 - AJAX.
 - Soporta extensiones.
 - Utilidades varias como obtener información del navegador, operar con objetos y vectores.
-
- **AJAX:** Es una técnica que se lleva a cabo mediante programas escritos en Javascript, la cual permite que un servidor y un navegador intercambien información, posiblemente en XML del lado del cliente para realizar peticiones asíncronas al servidor, es decir: sin la participación directa del usuario, las cuales permiten actualizar ciertas secciones de la página, sin necesidad de recargarla por completo. En ambientes donde no se implementa esta técnica, cualquier actualización no será visible hasta tanto se recargue por completo el contenido de la página (W3Schools, 2015).

El término AJAX se presentó por primera vez en el artículo "*Ajax: A New Approach to Web Applications*" publicado por Jesse James Garrett el 18 de febrero de 2005. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de aplicación *web* que estaba apareciendo. El artículo define AJAX como el hecho de que no es una tecnología en sí mismo, en realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes. Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
 - DOM, para la interacción y manipulación dinámica de la presentación.
 - XML, XSLT y JSON, para el intercambio y la manipulación de información.
 - XMLHttpRequest, para el intercambio asíncrono de información.
 - JavaScript, para unir todas las demás tecnologías.
-
- **JSON (*JavaScript Object Notation*):** Según (JSON, 2016), es un formato de texto ligero para el intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Es un subconjunto de la notación literal de objetos de JavaScript. A su vez, 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. Se considera un formato de lenguaje independiente.

JSON está constituido por dos estructuras (JSON, 2016):

- 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.

No obstante, 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.

2.4.5. Tecnologías de lado del Servidor

Son aquellas que se utilizan en la implementación del módulo encargado de lógica de negocio. Algunas de las tecnologías que se utilizan de este lado son las siguientes:

- **PHP:** Su propia organización lo define como un lenguaje de programación rápido, flexible y pragmática de uso general el cual funciona del lado del servidor y originalmente diseñado para el desarrollo *web* de contenido dinámico.

De acuerdo con (PHP Org. , 2016), fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor *web* con un módulo de procesador de PHP que genera la página *web* resultante.

PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores *web* al igual que en casi todos los sistemas operativos y plataformas.

En vez de escribir un programa con muchos comandos para crear una salida en HTML, el lenguaje PHP permite que se escriba el código HTML con cierto código PHP embebido (introducido) en el mismo, lo cual producirá cierta salida. El código PHP se incluye entre etiquetas especiales de comienzo y final que permitirán entrar y salir del modo PHP (PHP Org. , 2016).

Lo que distingue a PHP de la tecnología Javascript, la cual se ejecuta en el lado del cliente, es que el código PHP es ejecutado en el servidor como se mencionó anteriormente y el cliente solamente recibiría el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido. En general, sus desarrolladores expresan que este lenguaje es extremadamente simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales.

- **Laravel:** Es un *framework* de código abierto para desarrollar aplicaciones y servicios *web* con PHP. Su filosofía es desarrollar código PHP de forma elegante y simple, fue creado en 2011 y cuenta con características como (Blanco, 2013):
 - Sistema de ruteo.
 - Motor de plantillas (Blade).
 - Basado en Composer.
 - Soporte para el caché.
 - Usa componentes de Symfony.
 - Adopta las especificaciones PSR-2 y PSR-4.

Laravel, propone una forma de desarrollar aplicaciones *web* de un modo mucho más ágil. Por ejemplo, en Laravel opcionalmente se puede usar el patrón de diseño MVC (Modelo-Vista-Controlador) tradicional, donde al igual que otros *frameworks* PHP, el controlador es programado como una clase.

Asimismo, según expresa Blanco, Laravel incluye una valiosa pieza de *software*, llamada Eloquent ORM. Este ORM se funda en patrón *active record* y su funcionamiento es en extremo sencillo. Un ORM (*Object Relational Mapper*) en PHP es un *software* que permite tratar la capa de persistencia de los datos, como simples accesos a métodos de una clase u objeto en PHP.

La funcionalidad interna del ORM es mapear los objetos de PHP a las tablas en la base de datos, para el caso en que la persistencia de los datos de la aplicación es proporcionada por una DB. Sin embargo, en Laravel es opcional el uso de Eloquent, pues también dispone de otros recursos que facilitan interactuar con los datos, o específicamente la creación de modelos.

No obstante, se puede destacar que la influencia de Laravel ha crecido rápidamente desde su lanzamiento. En la comunidad de desarrolladores es considerado como alternativa sencilla de usar pero que tiene todas las funcionalidades que debe tener un *framework*. Ha sido descargado más de 320.000 veces, y se espera que supere en popularidad a otros *frameworks* ya establecidos más antiguos (Blanco, 2013).

2.4.6. Tecnologías de Manejo de Datos

Son aquellas que se utilizan para la persistencia de los datos. Para lograr esto, se hace uso de Sistemas Manejadores de Bases de Datos (SMBD), los cuales son un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Uno de los más usados en la actualidad es el manejador:

- **MySQL:** Es un SMBD relacional desarrollado, distribuido y soportado por la Corporación Oracle. Su facilidad de uso, seguridad, escalabilidad y velocidad lo han convertido en uno de los más populares que existen actualmente. Sus principales características se definen a continuación (MySQL, 2011):
 - Está escrito en C y C++, lo que permite su funcionamiento en diversas plataformas como Microsoft Windows, Linux, Mac OS, entre otras.
 - Existen *API's* capaces de conectar este sistema manejador con diferentes lenguajes de programación como PHP, C, C++, Perl, Java, Ruby, Python, entre otros, permitiendo el desarrollo de aplicaciones de diversos tipos.
 - Utiliza múltiples hilos a nivel de *kernel*, lo que permite aprovechar al máximo un sistema que disponga de múltiples procesadores.
 - Permite el uso de índices tipo Árbol-B, que ofrecen tiempos de respuesta óptimos para las consultas. Una tabla puede tener hasta un máximo de 64 índices asociados.
 - Soporta grandes cantidades de datos. MySQL Server puede contener según estadísticas de la organización, hasta 50 millones de registros, 200 mil tablas y 5 mil millones de columnas.
 - El sistema de privilegios y contraseñas es flexible pero seguro, utiliza verificación basada en host además de encriptar todas las contraseñas que trafiquen entre cliente y servidor.

En resumen, es un sistema manejador con características que le convierten en una opción bastante flexible, robusta y sencilla de utilizar al momento de desarrollar aplicaciones, no solo para ambientes *web*, sino también locales.

2.4.7. Otras Tecnologías de Desarrollo

Estas tecnologías no son otras que las herramientas a las cuales el equipo de desarrollo puede recurrir para facilitar el proceso de construcción del software. No existe un estándar que obligue a utilizar alguna u otra en particular, dejando siempre a criterio de los desarrolladores el uso o no de las mismas. Además, resultan transparentes para el usuario final y no afectan en nada a las tecnologías utilizadas tanto de lado del cliente como del servidor. Algunas de ellas son:

- **Git:** Es un sistema de control de versiones libre y de código abierto, está diseñado para manejar todo, desde pequeños proyectos hasta aquellos que son realmente grandes, todo con bastante rapidez, eficacia y sencillez.

Un sistema de control de versiones o SCV se define como aquel que permite realizar un seguimiento de la historia de una colección de archivos y además incluye la funcionalidad de revertir la colección de archivos actual hacia versión anterior.

En Git cada versión podría considerarse como una fotografía del estado de la colección en un momento determinado del tiempo. Cuando se aplica este concepto a un ambiente distribuido, se cuenta con un servidor central donde se almacena el repositorio principal y cada usuario puede hacer una copia completa del mismo mediante un proceso llamado "clonación" (Fiquis LTDA, 2014).

Git es un sistema de control de versiones distribuido que se origina a partir del desarrollo del *kernel* de Linux y es utilizado en muchos proyectos populares de código abierto.

- **TortoiseGit:** De acuerdo a la página oficial de TortoiseGit, éste es un cliente de código abierto para el sistema de control de versiones Git. Es decir, TortoiseGit gestiona los archivos con el tiempo (Ilustración 7). Los archivos se almacenan en un repositorio local. El repositorio es como un servidor de archivos ordinario, excepto que recuerda todos los cambios hechos a sus archivos y directorios. Esto le permite recuperar versiones antiguas de sus archivos y examinar la historia de cómo y cuándo cambiaron sus datos, y quién lo cambió.

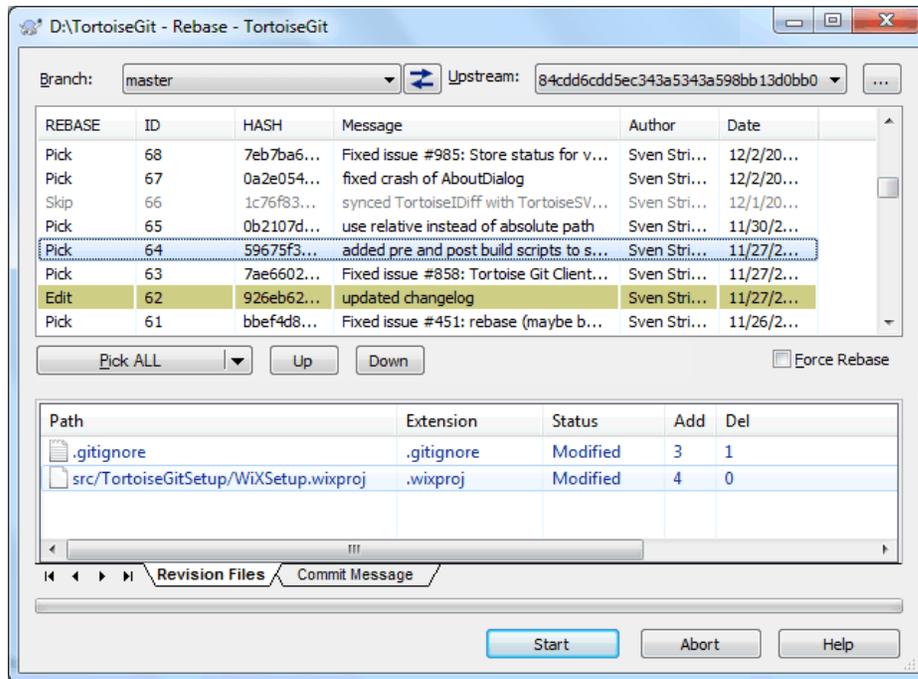


Ilustración 7: Interfaz de TortoiseGit.
Fuente: <https://tortoisegit.org/about/screenshots/>

Algunos sistemas de control de versiones también son sistemas de gestión de configuración de software (SCM). Estos sistemas están diseñados específicamente para manejar árboles de código fuente, y tienen muchas características que son específicas para el desarrollo de software - como el entendimiento nativo de lenguajes de programación, o el suministro de herramientas para la construcción de software. Git, sin embargo, no es uno de estos sistemas; se trata de un sistema general que puede ser utilizado para manejar cualquier colección de archivos, incluyendo el código fuente (TortoiseGit, 2015).

Git es un sistema de control de versiones de código abierto, distribuido diseñado para manejar todo, desde pequeños a grandes proyectos con rapidez y eficacia. Según (TortoiseGit, 2015), cada clon de un repositorio Git es de pleno derecho con la historia completa y capacidades de seguimiento de revisión, que no dependen del acceso a la red o un servidor central.

CAPÍTULO III

MARCO METODOLÓGICO

3.1 Metodologías convencionales

Para empezar, es necesario conocer qué es una metodología de desarrollo de *software*, por qué es necesario usar una o varias según sea el caso, así como algunos de los tipos existentes, características, ventajas y desventajas de cada una de ellas, de forma que sea posible escoger la o las que mejor se adapten para el desarrollo de este Trabajo Especial de Grado.

Un método de desarrollo de *software* define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, el cual servirá como referencia para enfrentar y resolver nuevos problemas de índole similar. Al mismo tiempo, es usado para estructurar, planificar y controlar el proceso de sistemas de información.

Partiendo de lo anterior, las metodologías se pueden dividir en los siguientes tipos de enfoques:

3.1.1 Cascada

Según (Pressman, 2002), este es el más básico de todos los modelos y ha servido como base para los demás paradigmas de ciclo de vida, es un proceso secuencial de desarrollo visto con pasos de arriba hacia abajo, durante todas las fases (requerimiento, diseño, implantación, pruebas, integración y mantenimiento) por lo que el desarrollo es secuencial.

Es caracterizado por ordenar de manera rigurosa las etapas del ciclo de vida de *software*, dado que el comienzo de cada etapa debe esperar a la finalización de la inmediata anterior. Cuando la revisión determina que el proyecto no está listo para pasar a la siguiente etapa, permanece en la etapa actual hasta que esté preparado. Debido a que el proceso está planeado es más fácil determinar costos y los plazos. Este modelo puede ser visto como un modelo con forma de cascada de agua con varios saltos, en la que cada salto representa cada una de las fases del ciclo de vida (Ver Ilustración 8).

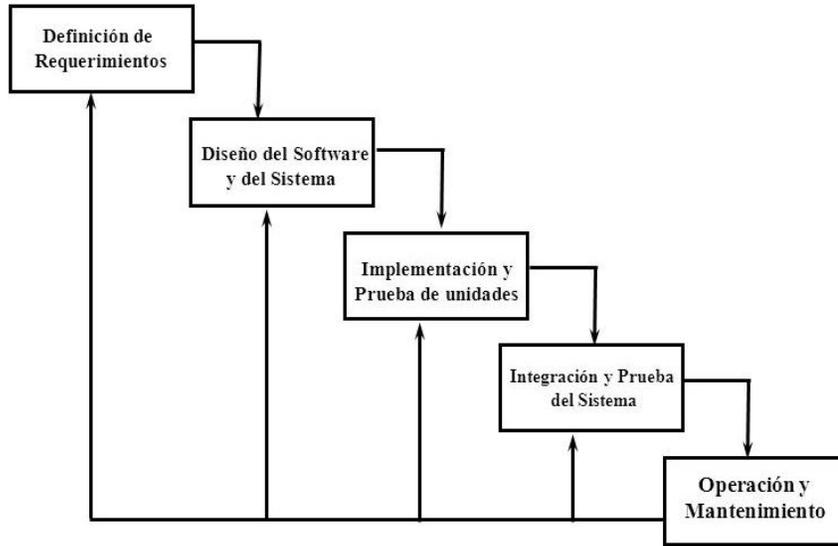


Ilustración 8: Ciclo de vida del Modelo en Cascada

Fuente: <http://slideplayer.es/slide/1464123/>

Este tipo de metodologías contemplan que la planificación del proyecto completo se realice de una sola vez. En proyectos muy grandes, este tipo de metodologías no ofrecen un adecuado manejo de los riesgos, por lo que es difícil asegurar un producto final que satisfaga adecuadamente las necesidades del usuario.

- **Ventajas**

- Es un modelo lineal y efectivo pues se trabaja en fases, definir antes que diseñar y diseñar antes que codificar.
- Es fácil de implementar y entender.
- Está orientado a documentos.
- Es conocido y utilizado con frecuencia.

- **Desventajas**

- No se puede volver atrás, es decir, si la fase de diseño ha ido mal, las cosas se pueden complicar en la fase de ejecución
- Una etapa determinada del proyecto no se puede llevar a cabo sin que se haya culminado la etapa anterior.
- El *software* no está en las manos del cliente hasta que la etapa final del ciclo de desarrollo se ha completado.

- Los proyectos de *software* requieren muchos cambios así que rara vez sigue una secuencia lineal, esto crea una mala implementación del modelo, lo cual hace que lo lleve al fracaso.

3.1.2. Desarrollo iterativo y creciente (incremental)

De acuerdo con (EduRed, 2012), es un proceso de desarrollo de *software* que surge como respuesta a las debilidades presentadas por el modelo tradicional de cascada. Las metodologías que manejan este tipo de esquema por lo general dividen el problema en partes, compuestas por un conjunto de mini-cascadas, es decir, para cada una de ellas se realizan todas las fases, dejando como resultado parte del producto final en funcionamiento. La idea es generar una a una, las funcionalidades del sistema, hasta llegar al producto final.

Con esto se mantiene al cliente en constante contacto con los resultados obtenidos en cada incremento, lo cual permite que sea el mismo cliente el que incluya o deseche elementos al final de cada incremento a fin de que el *software* se adapte mejor a sus necesidades reales, repitiendo el proceso hasta que se elabore el producto completo. De esta forma el tiempo de entrega se reduce considerablemente.

Según (Pressman, 2002) Este modelo es particularmente útil cuando no se cuenta con una dotación de personal suficiente. Los primeros pasos los pueden realizar un grupo reducido de personas y en cada incremento se añadirá personal, de ser necesario. En la Ilustración 9, se puede visualizar el ciclo de vida de este método.

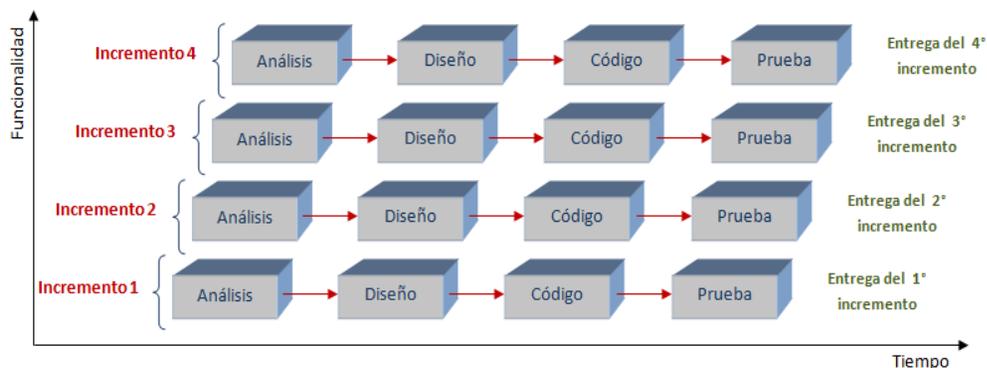


Ilustración 9: Ciclo de vida del Método Incremental

Fuente: <https://procesossoftware.wikispaces.com/Modelo+Incremental>

Durante el proceso se trata de llevar a cabo al proyecto en diferentes partes que al final terminará siendo la solución completa requerida por el cliente, pero estas partes no se pueden realizar en cualquier orden, sino que dependen de lo que el cliente esté necesitando con más urgencia, de los puntos más importantes del proyecto, los requerimientos más básicos, difíciles y con mayor grado de riesgo, ya que estos se deben hacer al comienzo, de manera que se disminuya la dificultad y el riesgo en cada versión.

- **Características**

- Se evitan proyectos largos y se entrega "algo de valor" a los usuarios con cierta frecuencia.
- El usuario se involucra más.
- Requiere gestores experimentados.
- Los errores en los requisitos se detectan tarde.
- El resultado puede ser positivo.

- **Ventajas**

- Los usuarios no tienen que esperar hasta que el sistema esté completo para usarlo ya que en el primer incremento se busca cumplir con los requerimientos más importantes de forma que pueda utilizarse el *software* al instante.
- Los usuarios pueden utilizar los prototipos iniciales y obtener experiencia para los requerimientos de los incrementos posteriores.
- Como el trabajo se ha llevado a cabo por incrementos existen muy pocas posibilidades de riesgo en el sistema, lo que conlleva a que el sistema se entregue sin inconvenientes al usuario. Aunque en ocasiones se pueden encontrar problemas en algunos incrementos.
- Como los módulos del sistema que tienen la prioridad más alta se entregan primero y los incrementos posteriores se integran en base a ellos, es muy poco probable que los usuarios encuentren fallas de funcionamiento del *software* en las partes más importantes del sistema.

- **Desventajas**

- La entrega temprana de los proyectos produce la creación de sistemas demasiados simples ante el personal que lo recibe.
- La mayoría de los incrementos se harán en base de las necesidades de los usuarios.
- Requiere de un cliente involucrado durante todo el proyecto. Sin embargo, hay clientes que no estarán dispuestos a invertir el tiempo necesario.
- La entrega de un programa que es parcial pero funcional puede hacer vulnerable al sistema debido a la falta de robustez, provocando que agentes ajenos puedan interferir con el correcto funcionamiento del programa en sí.

3.1.3 Espiral

Según (EduRed, 2013), definido por primera vez por Barry Boehm en 1986, este proceso contempla el desarrollo mediante la división del proyecto en segmentos más pequeños los cuales se planifican y tratan de forma separada, pero pasando cada uno de ellos por su respectivo conjunto de fases, donde las actividades no están fijas a ninguna prioridad, sino que se eligen en función del análisis de riesgo generado por el bucle interior; de igual forma se incluyen la evaluación de la iteración actual y la planificación de la siguiente.

Este método consta de un ciclo que se basa principalmente en una serie de iteraciones repetitivas ya que esto ayudara a que el proyecto gane madurez, y así se pueda dar un resultado más satisfactorio de la aplicación final. En la Ilustración 10, se muestran cada una de las fases de las cuales consta el modelo en espiral.

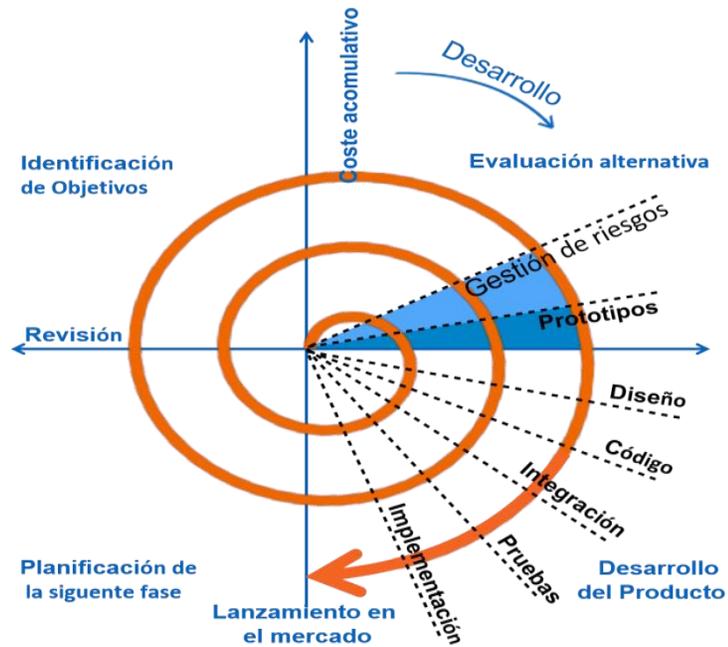


Ilustración 10: Ciclo de vida del Método en Espiral

Fuente: <https://www.emaze.com/@AIFZCZT/Carolina-Ram%C3%ADrez>

- **Ventajas**

- Incorpora objetivos de calidad.
- Es adaptable a cualquier número de cambios que pueden ocurrir durante cualquier fase del proyecto.
- Reduce riesgos del proyecto.
- Se puede estimar los costos de cada cambio fácilmente, ya que estos se efectúan sobre un segmento del proyecto y no afecta de manera global al mismo.

- **Desventajas**

- Genera mucho tiempo en el desarrollo del sistema.
- Debe ser seguido estrictamente para su buen funcionamiento.
- Funciona mejor para los grandes proyectos.
- Requiere experiencia en la identificación de riesgos.

3.1.4 Prototipo

Para (Pressman, 2002), es una metodología de desarrollo evolutivo que debe ser construida en poco tiempo pues será una representación de aquellos aspectos del *software* que serán visibles para el cliente o el usuario final sobre los cuales se implementarán poco a poco las funcionalidades requeridas. Con esta metodología se puede evaluar el proyecto prácticamente durante todo el proceso de desarrollo, verificando si las expectativas se cumplen o no. No es necesario implementar el conjunto total de funcionalidades del sistema por prototipo, pero si se considera la idea principal del mismo, en algunos prototipos pueden incluir requerimientos ya diseñados del sistema.

Esta metodología se inicia básicamente con la definición de los objetivos globales para el *software*, luego se identifican los requisitos conocidos y las áreas del esquema en donde es necesaria más definición. Entonces se plantea con rapidez una iteración de construcción de prototipos y se presenta el modelado (en forma de un diseño rápido).

El diseño rápido se centra en una representación de aquellos aspectos del *software* que serán visibles para el cliente o el usuario final (por ejemplo, la configuración de la interfaz con el usuario y el formato de los despliegues de salida), y esto conduce a la construcción de un prototipo, el cual es evaluado por el cliente o el usuario para una retroalimentación; gracias a ésta se refinan los requisitos del *software* que se desarrollará. La interacción ocurre cuando el prototipo se ajusta para satisfacer las necesidades del cliente. Esto permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo (Ver Ilustración 11).

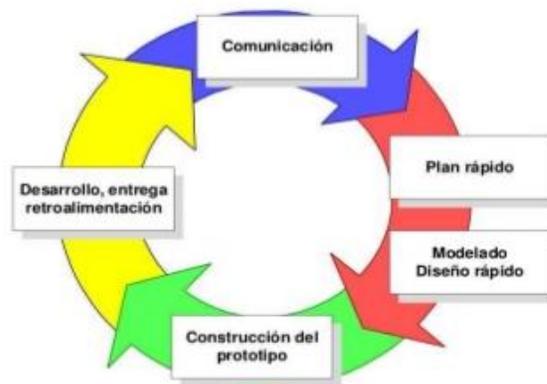


Ilustración 11: Ciclo de vida del Modelo en Prototipo.

Fuente: <http://es.slideshare.net/juriberuiz/modelo-de-prototipos-32733577>

- **Ventajas**

- Es de rápido diseño
- Reducción del tiempo y costo de trabajo
- Cada entrega genera retroalimentación, salvo la entrega final.
- Incrementa la posibilidad de aceptación del sistema.

- **Desventajas**

- Dado que el diseño debe hacerse de manera rápida el o los desarrolladores suelen tomar algunas decisiones de implementación poco convenientes, como la tecnología a usar, los módulos que tendrá, entre otros. Los cuales tienen la posibilidad de ser desechados a futuro por otros que se adapten mejor a los requisitos.
- Se necesitan varias iteraciones para lograr un producto de calidad.
- El usuario tiende a suponer que el sistema puede ser hecho de manera rápida y fácil cuando ve el prototipo y a menudo suele pensar que está cerca del sistema final.

3.1.5 Método AgilUs

De acuerdo con (Acosta, 2011), *AgilUs* es un método de desarrollo ágil, iterativo e incremental. Se basa en el concepto de usabilidad, resaltando la importancia de desarrollar *software* usable. Plantea un análisis centrado tanto en el usuario como en la participación de especialistas que contribuyan con la evolución del *software*.

Su principal valor establece que la construcción de las interfaces de usuario no debe ser un agregado estético que se da al final del proyecto sino más bien aquello que guíe el proceso de desarrollo y la toma de decisiones. En *AgilUs* son los usuarios, y no el cliente ni los programadores quienes guían el desarrollo del proyecto.

En general, este método busca proporcionar un conjunto de actividades organizadas para construir la usabilidad en el diseño de interfaces de usuario durante el desarrollo de un producto de *software*.

Se estructura en un total de cuatro fases, durante las cuales es posible realizar iteraciones sobre los artefactos y donde la participación del usuario es clave en el éxito de las mismas. Para mayor flexibilidad, este método permite que dependiendo del contexto asociado al problema que se esté resolviendo, la aplicación de algunos artefactos sea innecesaria. Las fases que involucran este método son las siguientes:

- **Requisitos:** Se realiza un análisis global del problema a solucionar, se estudian productos similares existentes, se genera un perfil de usuario, y se define la lista de requerimientos a desarrollar. Esta etapa es importante, ya que un mal análisis de requisitos traería como consecuencia un *software* que no cumple con las necesidades del usuario.
- **Análisis:** Se lleva a cabo el análisis de la solución a desarrollar, se emplean diagramas de casos de uso y modelado de objetos del dominio, siguiendo la notación UML, para definir las funcionalidades que tendrá el producto a desarrollar.
- **Prototipaje:** Se implementa un prototipo rápido de interfaz de usuario a partir de los patrones de interacción, el cual va evolucionando hasta convertirse en el producto final, se genera la guía de estilo, y se realizan las evaluaciones de usabilidad apropiadas a esta etapa.
- **Entrega:** Se aplican las pruebas al sistema para certificar que la aplicación desarrollada sea un *software* usable y sin errores para luego, ponerla en producción.

En la Ilustración 12 se puede observar el ciclo de vida del método a través de sus fases y los artefactos asociados a ellas.



Ilustración 12: Ciclo de vida del método AgilUs

Fuente: http://www.ciens.ucv.ve/portalsig/interacci%C3%B3n_humano_computador/1-2014/descarga/descargar_archivo/601

A lo largo de los años las metodologías tradicionales eran la mejor opción para desarrollar cualquier tipo de proyecto, sin embargo, en la actualidad existen factores que han llevado a reducir la tendencia del uso de los mismos. Uno de ellos y quizás el principal factor en contra, es la cantidad de documentación que las metodologías sugieren hacer a lo largo del proceso de desarrollo. Esto generará que el equipo de desarrollo deba invertir una gran cantidad de trabajo y un alto tiempo en ello para realizar documentos que mayormente no son de interés para el usuario final.

Sin embargo, luego de evaluar todas las metodologías señaladas, para efectos de este trabajo se empleará la metodología de desarrollo iterativo y creciente, ya que con ella se puede iniciar el desarrollo de la aplicación mediante una implementación simple de los requerimientos del sistema que a su vez permite iterativamente mejorar la secuencia evolutiva de versiones hasta que el sistema esté implementado completamente. En cada iteración, se puede realizar cambios en el diseño y agregar nuevas funcionalidades y capacidades al sistema, con la finalidad de crear una aplicación *web* para la gestión del proceso de ingreso de los aspirantes a entrar a la UCV con la que los usuarios pueden interactuar, y por ende retroalimentar el proceso.

CAPÍTULO IV

MARCO APLICATIVO

En este capítulo se documenta el proceso de desarrollo de las actividades que se siguieron para la implementación de la aplicación *web*, objeto del presente Trabajo Especial de Grado, tomando como base, la metodología seleccionada previamente, es decir, la Metodología de Desarrollo Iterativo y Creciente (Ver Capítulo III), la cual permite de una forma flexible, rápida y efectiva la implantación de dicha aplicación.

Con este sistema se automatiza el proceso de ingreso de estudiantes (aspirantes) a la UCV, proceso que es realizado actualmente de forma manual por el personal administrativo de cada facultad junto a la Secretaría General de la universidad, haciendo uso de un conjunto de tecnologías para el desarrollo de una aplicación *web* que le permita al aspirante, registrarse al sistema, ingresar sus datos personales y académicos, y responder varias pruebas que debe superar para ser seleccionado; con la finalidad de captar estudiantes preparados para recibir una formación de calidad y garantizar la prosecución de estudios con un egreso exitoso.

Tal como se planteó en el Capítulo I, este sistema se diseña partiendo de dos fases, las cuales se integran por un conjunto de procesos. Dentro de esas fases resaltan: el registro de los aspirantes en el sistema, así como también la selección de los mismos, la cual se conforma por la realización de una prueba vocacional con la publicación de los resultados obtenidos, hasta la realización de una prueba diagnóstica con la respectiva publicación de los aspirantes que aprobaron dicha prueba y fueron seleccionados para ingresar a la universidad.

Todas esas fases junto a los procesos que las involucran, se plantean siguiendo un conjunto de iteraciones, implementadas en forma incremental, basado en el nivel de prioridad y en la dependencia que existe en el desarrollo de las mismas.

Sin embargo, antes de dar inicio a las iteraciones que comprenden dichas fases del sistema, se define una iteración previa, la cual consiste en recaudar y estudiar de forma general los requerimientos de información proporcionados por la universidad, con la finalidad de diseñar e implementar la base de datos que da soporte a toda la aplicación. Partiendo de esto, se tiene como resultado la definición de las siguientes iteraciones (Ver Tabla 1):

Tabla 1: Definición de las iteraciones de la aplicación web.

Iteración	Actividad
1	Esquema general de la solución.
2	Diseño e implementación del modelo relacional de la base de datos
3	Registro de los aspirantes.
4	Prueba vocacional con el cálculo y publicación de los resultados obtenidos.
5	Validación de recaudos y publicación de la selección de los aspirantes.
6	Prueba diagnóstica con el cálculo y publicación de los resultados obtenidos.
7	Módulo de administración.

Asimismo, una vez bien definido el alcance de la aplicación, se estima llevar a cabo esta planificación en un periodo de 4 semanas aproximadamente, en donde cada una de las iteraciones establecidas, tiene un tiempo determinado y controlado para su buen desarrollo con el objetivo de culminar satisfactoriamente este sistema.

A continuación, se detallan las actividades correspondientes a cada iteración:

4.1. Iteración 1: Esquema General de la Solución.

La primera iteración consiste en determinar las bases sobre las cuales se construye el sistema. Para el desarrollo de este proyecto, se cuenta con dos desarrolladores y los roles de dueño del producto y líder del proyecto son asumidos por los tutores. Asimismo, se plantea realizar un chequeo diariamente en el que se verifican las actividades que se completan de forma exitosa y luego de culminar una versión del producto se mantienen reuniones con los demás miembros del equipo para verificar los resultados de cada iteración y certificarlos o proponer las modificaciones necesarias.

Por tal motivo, inicialmente se realizan reuniones con el equipo de desarrollo del proyecto para definir los requerimientos funcionales y no funcionales del sistema. Estos1 requerimientos componen la pila de producto y algunos de ellos son presentados a continuación:

- **Requerimientos Funcionales**

- Permitir el registro y la carga de los datos de usuario, personales y académicos de los aspirantes en el sistema.
- Mostrar una vista previa de los datos suministrados por el aspirante.
- Iniciar y cerrar sesión en el sistema.
- Recuperar contraseña de usuario.
- Modificar perfil de usuario.
- Gestionar materias e importancia de las carreras.
- Disponer de un conjunto de áreas temáticas para que el aspirante seleccione el área de su preferencia y con base en ello, elegir las facultades y carreras que desea cursar.
- Mostrar la prueba vocacional y la prueba diagnóstica.
- Cargar y eliminar modelos de las pruebas vocacionales y los de las pruebas diagnósticas.
- Construir, modificar, eliminar y consultar reportes que reflejen información, mediante el análisis de los datos almacenados en la base de datos del sistema.
- Calcular el promedio del aspirante.
- Almacenar y procesar las respuestas recogidas de las pruebas realizadas por el aspirante.
- Garantizar la persistencia de los datos a través de una base de datos.
- Segmentar el sistema por fases (registro y carga de datos, prueba vocacional y preselección de aspirantes, validación de recaudos y por último, la prueba diagnóstica y selección de los aspirantes que aprobaron). Cada fase puede o no componerse por pasos para una mejor interacción del usuario con el sistema.
- Seleccionar los aspirantes que cumplen las reglas del negocio.
- Resguardar la información de los usuarios.
- Publicación de los resultados de las pruebas realizadas por los aspirantes y la selección de los mismos.

- **Requerimientos No Funcionales**

- El producto final deberá ofrecer un excelente rendimiento a fin de que el usuario pueda realizar sus tareas de forma rápida y correcta.
- Debe ser un programa de uso sencillo e intuitivo, con un alto grado de usabilidad.
- Se desea que el producto sea fácilmente extensible para cualquier desarrollador que desee incorporar nuevas funcionalidades.
- El producto final debe ser capaz de prever errores a partir de la definición de validaciones.
- Ante fallas o situaciones excepcionales la aplicación debe ser capaz de dar una solución apropiada y seguir en funcionamiento.
- La estructura del código debe ser consistente y predecible.
- El producto debe ofrecer un buen desempeño ante una alta demanda de usuarios.

Una vez definidos estos requerimientos, se realiza un análisis que permite identificar los perfiles de los usuarios que interactuarán con el sistema. A continuación, se explican estos perfiles de usuario:

- **Administradores:** Son las personas encargadas de gestionar la aplicación.
- **Operadores:** Personal administrativo perteneciente a una facultad de la UCV que tiene por finalidad recibir los recaudos en físico que se le solicitan al aspirante con el objetivo de comprobar la veracidad de los datos proporcionados por dicho aspirante en el sistema.
- **Supervisores:** Personal administrativo perteneciente a una facultad de la UCV que verifica y valida las observaciones realizadas por un operador.
- **Aspirantes:** Estudiantes del último año de la Educación Media, Diversificada y Profesional que desean cursar alguna carrera dentro de la UCV.

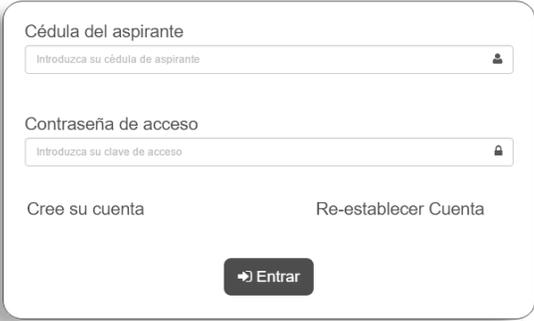
A nivel tecnológico y de acuerdo al estudio realizado en cuanto a las diversas herramientas y tecnologías existentes para el desarrollo de una aplicación *web* (Ver Capítulo II), se determinan aquellas que se van a utilizar dentro de la arquitectura tecnológica del sistema tomando en cuenta los requerimientos del sistema expuestos previamente. En la Tabla 2 se observan las herramientas seleccionadas.

Tabla 2: Herramientas seleccionadas.

Herramientas	Descripción
MySQL 5.6.24	SMBD utilizado para implementar las estructuras de datos que soportan los datos que integran el sistema de ingreso. Se selecciona debido a su robustez, buena documentación, buen desempeño y fácil uso.
Framework Laravel 5.2	Seleccionado para el desarrollo de la aplicación <i>web</i> . Además, se incorporan las tecnologías HTML5, CSS3, Javascript, jQuery, Ajax, AngularJs y JSON con la finalidad de hacer que el usuario tenga una experiencia más agradable con el sistema sin sobrecargar el servidor.
TortoiseGit 2.1.0	Seleccionado para el manejo de las versiones de desarrollo del sistema debido a que permite manejar el código y mejorar el flujo de trabajo de forma óptima logrando que el tiempo de trabajo sea productivo.

Una vez definido el diseño de la arquitectura técnica se establecen los lineamientos gráficos a utilizar en las interfaces de usuario que conforman el sistema, para ello se define una guía de estilo que resume los aspectos gráficos del sistema de información. En la Tabla 3 se muestran los colores, tipografía, botones y estilo de interacción determinados para esta aplicación.

Tabla 3: Guía de Estilo

Colores a utilizar en los elementos de interfaz	
Tipografía	Tamaño: 20px, 16px y 12px Tipo: Open Sans, Open Sans
Botones	
Estilos de interacción de forma	

4.2. Iteración 2: Diseño e Implementación del Modelo Relacional de la Base de Datos

En esta iteración se diseña e implementa el modelo relacional que soporta las entidades necesarias para almacenar la información obtenida a partir de las distintas interfaces de la aplicación *web*.

Se realizan reuniones entre los miembros del equipo de desarrollo que permiten identificar y definir las entidades que debían ser contempladas en este modelo. Dentro de estas entidades resaltan las que permiten almacenar la información de los aspirantes (personal y académica), áreas temáticas que se manejan en la universidad, así como las facultades y sus respectivas carreras. También dichas entidades permiten almacenar la información relacionada al manejo de las pruebas y las respuestas obtenidas para cada una de ellas, la duración de las distintas fases del sistema, los modelos para las pruebas, entre otros.

Como resultado, el diseño del modelo relacional de la base de datos que da sustento a los datos que manejará el sistema, se conforma por un total de veintitrés (23) tablas, y para dicho diseño se considera lo siguiente:

- Los nombres de las tablas y los atributos se definen en singular.
- Cada tabla tiene una clave primaria autoincremental de tipo entero llamada *id* para satisfacer las condiciones de nomenclatura utilizadas por el *framework* Laravel en los campos clave.
- Las claves foráneas son de tipo entero y se identifican con el nombre de la tabla a la que hacen referencia más la palabra *id*.
- Las claves primarias y foráneas son atributos que no pueden ser nulos.

Ese modelo se implementa físicamente en el Sistema Manejador de Base de Datos MySQL, definiendo los diferentes elementos que lo componen: tablas y claves. Esta implementación se realiza con el *framework* Laravel a través de migraciones, obteniendo de esta manera las estructuras para almacenar los datos de la aplicación (Ilustración 13).

<pre> tesis_tabla_inscripcion @id: int(10) unsigned @user_cedula: int(11) @user_apellido: varchar(255) @user_paso: int(11) @user_promedio_general: double @user_promedio_opcion_1: double @user_promedio_opcion_2: double @user_promedio_opcion_3: double @periodo_id: int(10) unsigned @created_at: timestamp @updated_at: timestamp tesis_estados @id: int(10) unsigned @nombre: varchar(255) @created_at: timestamp @updated_at: timestamp tesis_preguntas @id: int(10) unsigned @prueba_id: int(11) @numero_pregunta: int(11) @texto: text @respuesta_1: text @respuesta_2: text @respuesta_3: text @respuesta_4: text @tipo: int(11) @created_at: timestamp @updated_at: timestamp tesis_fases @id: int(10) unsigned @nombre: varchar(255) @tabla: varchar(255) @publicar: enum('NO','SI') @estado: enum('ABIERTA','CERRADA') @inicio: datetime @fin: datetime </pre>	<pre> tesis_materia_liceo @id: int(10) unsigned @codigo: varchar(255) @nombre: varchar(255) @periodo_id: int(10) unsigned @created_at: timestamp @updated_at: timestamp tesis_facultad @id: int(10) unsigned @area_id: int(10) unsigned @nombre: varchar(255) @periodo_id: int(10) unsigned @created_at: timestamp @updated_at: timestamp tesis_administracion @id: int(10) unsigned @adm_cedula: int(11) @clave: text @rol: int(11) @facultad: text @nombre: text @correo: varchar(255) @periodo_id: int(10) unsigned @created_at: timestamp @updated_at: timestamp </pre>	<pre> tesis_parroquia @id: int(10) unsigned @municipio_id: varchar(255) @nombre: varchar(255) @created_at: timestamp @updated_at: timestamp tesis_menciones_bachillerato @id: int(10) unsigned @codigo: varchar(255) @nombre: varchar(255) @periodo_id: int(10) unsigned @created_at: timestamp @updated_at: timestamp tesis_estructura_prueba @id: int(10) unsigned @prueba_id: int(11) @titulo: text @mensaje: text @texto: text @preguntas: text @created_at: timestamp @updated_at: timestamp </pre>	<pre> tesis_mensajes_sistema @id: int(10) unsigned @seccion: text @mensaje: text @created_at: timestamp @updated_at: timestamp tesis_horarios_pruebas @id: int(10) unsigned @modelo: varchar(255) @inicio: datetime @fin: datetime @created_at: timestamp @updated_at: timestamp tesis_migrations @migration: varchar(255) @batch: int(11) tesis_municipio @estado_id: int(10) unsigned @nombre: varchar(255) @created_at: timestamp @updated_at: timestamp </pre>	<pre> tesis_carrera @id: int(10) unsigned @codigo: varchar(255) @nombre: varchar(255) @facultad_id: int(10) unsigned @materias_importantes: varchar(255) @menciones: varchar(255) @cupos: int(10) unsigned @periodo_id: int(10) unsigned @created_at: timestamp @updated_at: timestamp tesis_bitacora_validacion @id: int(10) unsigned @modificador_id: int(11) @aspirante_id: int(11) @evento: text @periodo_id: int(10) unsigned @created_at: timestamp @updated_at: timestamp tesis_periodos @id: int(10) unsigned @nombre: text @activa: enum('SI','NO') @created_at: timestamp @updated_at: timestamp </pre>	<pre> tesis_area @id: int(10) unsigned @nombre: varchar(255) @periodo_id: int(10) unsigned @created_at: timestamp @updated_at: timestamp tesis_usuario @id: int(10) unsigned @user_name: varchar(255) @user_apellido: varchar(255) @user_cedula: int(11) @user_phone: varchar(255) @user_fecha_nacimiento: date @user_genero: enum('0','1') @user_especialidad: int(11) @user_status: varchar(255) @user_email: varchar(255) @user_password: text @user_estado: varchar(255) @user_municipio: varchar(255) @user_parroquia: varchar(255) @user_liceo: varchar(255) @user_liceo_estado: varchar(255) @user_liceo_municipio: varchar(255) @user_liceo_parroquia: varchar(255) @user_pdf: varchar(255) @user_notas: text @user_opciones: varchar(255) @api_token: varchar(255) @admitido: varchar(255) @fase_id: int(10) unsigned @periodo_id: int(10) unsigned @created_at: timestamp @updated_at: timestamp tesis_tabla_validacion @id: int(10) unsigned @user_cedula: int(11) @estado_operador: int(11) @estado_supervisor: int(11) @observaciones: text @revisiones: text @periodo_id: int(10) unsigned @created_at: timestamp @updated_at: timestamp </pre>	<pre> tesis_tabla_test_diagnostico @id: int(10) unsigned @user_cedula: int(11) @modelo: char(3) @permutacion: int(11) @verbal: text @inicio_verbal: datetime @fin_verbal: datetime @seccion_verbal: int(11) @logico: text @inicio_logico: datetime @fin_logico: datetime @seccion_logico: int(11) @estado_prueba: enum('premioc','inicio','verbal','logico','fin') @resultado_verbal: double @resultado_logico: double @resultado_final: double @indice: double @seleccion: int(11) @periodo_id: int(10) unsigned @created_at: timestamp @updated_at: timestamp tesis_tabla_test_vocacional @id: int(10) unsigned @user_cedula: int(11) @respuestas: text @resultado_1: double @resultado_2: double @resultado_3: double @resultado_promedio_1: double @resultado_promedio_2: double @resultado_promedio_3: double @finalizado: tinyint(1) @periodo_id: int(10) unsigned @created_at: timestamp @updated_at: timestamp </pre>
---	---	--	---	---	---	--

Ilustración 13: Estructuras de la Base de Datos en MySQL creadas por Laravel.

No obstante, para llevar el modelo relacional (Ilustración 3) a la base de datos, primero hay que crear el repositorio donde se almacenarán los datos y luego conectar el *framework* a MySQL, accediendo al archivo “database.php” localizado en la ruta: app/config/database.php para configurar el *driver* de la conexión indicando que el motor de base de datos con el que se trabajará es MySQL, a través de la siguiente forma:

```
'default' => env('DB_CONNECTION', 'mysql').
```

Posteriormente, se configura el archivo .env ubicado en la raíz del proyecto para indicar el host, nombre de la base de datos a trabajar y el usuario con el cual se accede a ella, estableciéndole los valores correspondientes a las siguientes variables:

- Db_Host
- Db_Database
- Db_Username
- Db_Password

Una vez realizado todo lo anterior, se ejecuta el siguiente comando para construir las migraciones de Laravel mediante la terminal que posee:

```
php artisan migrate
```

Las migraciones en Laravel es una herramienta que permite crear un sistema de control de versiones de bases de datos donde se pueden definir tablas con Programación Orientada a Objeto en vez de sentencias SQL, dado que internamente este *framework* genera el SQL solo.

Las tablas que se crean a raíz del comando anterior si se ejecuta por primera vez, son: *migrations*, *password_resets*, *users*, destacando que el plural de sus nombres es debido a la convención que adopta el *framework*. Luego de eso, para crear una tabla en la base de datos se usa el comando:

```
php artisan make:migration crear_nombre_migracion --create=nombre_tabla
```

Partiendo de lo anterior, se proceden a crear veintitrés (23) nuevas migraciones las cuales representan una migración por cada tabla que posee el modelo relacional de base de datos, originando un archivo por migración dentro del proyecto, tal como se muestra en la Ilustración 14.

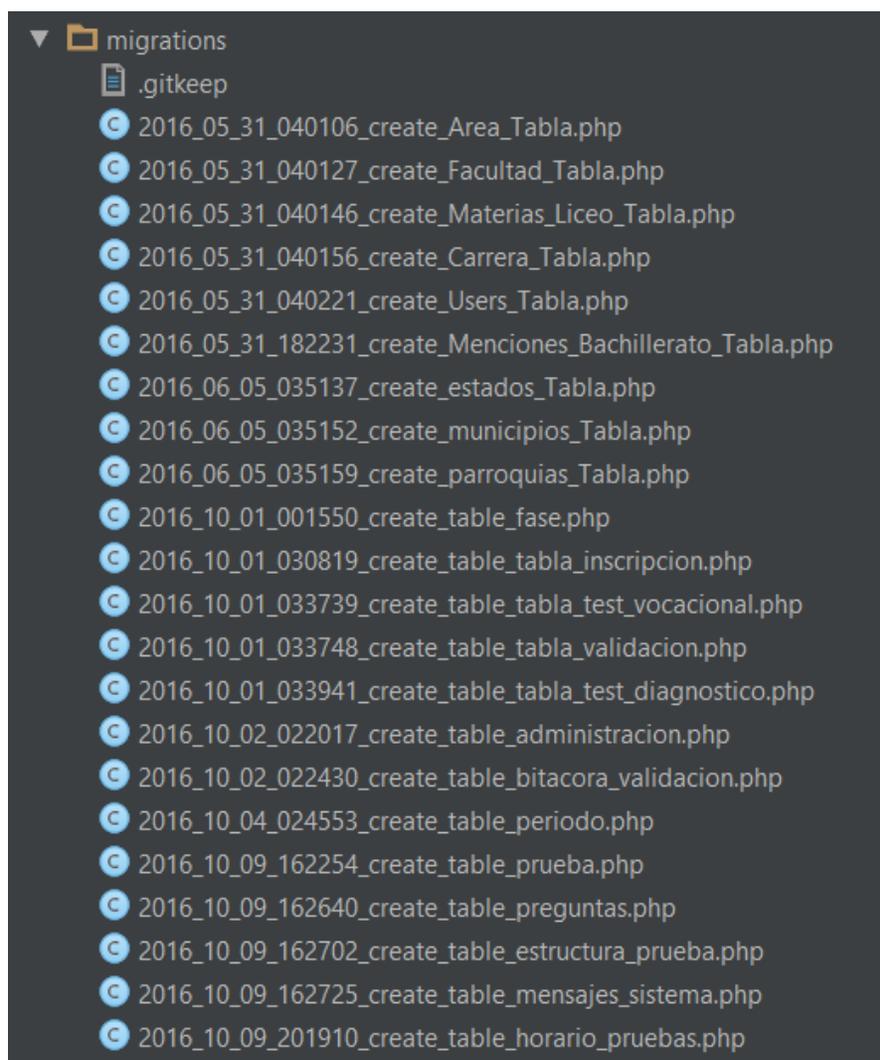


Ilustración 14: Archivos de migraciones creadas.

Cada uno de esos archivos poseen una estructura donde se establecen dos funciones, una llamada *up()* y otra llamada *down()*, la primer función es en donde se indican los campos de la respectiva tabla, mientras que la segunda función, hace lo opuesto, es decir, elimina la tabla de la base de datos. Un ejemplo de la estructura de estos archivos es el llamado “CreateUsersTabla”, mostrado en la Ilustración 15, el cual permite crear o eliminar según sea el caso, la tabla de usuarios del sistema.

```

<?php

use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateUsersTabla extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('user', function (Blueprint $table) {
            $table->increments('id');
            $table->string('user_name');
            $table->string('user_apellido');
            $table->integer('user_cedula');
            $table->string('user_phone');
            $table->date('user_fecha_nacimiento');
            $table->enum('user_genero', ['0', '1']);
            $table->integer('user_especialidad');
            $table->string('user_status');
            $table->string('user_email');
            $table->text('user_password');
            $table->string('user_estado');
            $table->string('user_municipio');
            $table->string('user_parroquia');
            $table->string('user_liceo');
            $table->string('user_liceo_estado');
            $table->string('user_liceo_municipio');
            $table->string('user_liceo_parroquia');
            $table->string('user_pdf');
            $table->text('user_notas');
            $table->string('user_opciones');
            $table->string('api_token', 255);
            $table->enum('api_token_estatus', ['0', '1']);
            $table->string('admitido')->nullable();
            $table->integer('fase_id')->unsigned();
            $table->integer('periodo_id')->unsigned();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::drop('user');
    }
}

```

Ilustración 15: Archivos de migraciones para la tabla Usuarios.

En consecuencia, se vuelve a ejecutar el comando:

```
php artisan migrate
```

Con el objetivo de crear físicamente todas las tablas definidas del modelo relacional, en la base de datos, y así finalmente, hacer las respectivas validaciones para comprobar el estado correcto en cuanto a la creación de las tablas, a fin de detectar cualquier error y solventarlo.

4.3. Iteración 3: Registro de los Aspirantes.

Esta iteración consiste en desarrollar las funcionalidades asociadas al registro del aspirante, la carga de información personal y académica del mismo, así como también el inicio de sesión en el sistema.

Durante esta iteración se mantienen reuniones con el equipo de desarrollo del proyecto en las que se presentan las versiones generadas. Asimismo, de acuerdo a lo que se plantea en esas reuniones, se consideran ciertos aspectos importantes, tales como:

- Establecer una única interfaz de inicio que permita crear una cuenta de usuario a los aspirantes que ingresan por primera vez al sistema o iniciar sesión para aquellos que ya se encuentran registrados.
- Redirigir al usuario cuando desee crear una cuenta, a la página que le solicita ingresar sus datos personales, así como su ubicación geográfica y la ubicación geográfica de la institución donde cursó bachillerato.
- En caso del aspirante ya posee una cuenta de usuario y sigue vigente el periodo de tiempo definido para el registro del mismo, redirigirlo al resumen de los datos que cargó previamente.
- Utilizar *modals* para notificarle al usuario cualquier información adicional asociada a esta fase.
- Definir las metáforas utilizadas para identificar las diferentes funciones.
- Incorporar textos de ayuda en las interfaces.

Los resultados obtenidos de esta iteración se muestran a continuación.

Funcionalidad de Inicio de Sesión

Esta funcionalidad es representada por la interfaz expuesta en la Ilustración 16. En ella se observa el nombre de la aplicación en la parte superior de la interfaz con una imagen asociada a uno de los espacios de la UCV y el logo de la universidad y el de Secretaria. Asimismo, se presenta un formulario en el que los usuarios pueden indicar los datos necesarios para acceder al sistema, crear una cuenta o recuperar su contraseña en caso de haberla olvidado.

Admisión 2016
Sistema de Ingreso por Mérito Académico y Diagnóstico

UNIVERSIDAD CENTRAL
CARACAS - VENEZUELA

SECRETARIA
UCV

Registro

Cédula del aspirante
Introduzca su cédula de aspirante

Contraseña de acceso
Introduzca su clave de acceso

Cree su cuenta Re-establecer Cuenta

Entrar

Universidad Central de Venezuela
Ciudad Universitaria, Los Chaguaramos / Caracas, Venezuela, Código Postal: 1050
RIF: G-20000062-7

Ilustración 16: Interfaz de inicio de sesión.

Si el aspirante ingresa por primera vez al sistema puede crear una cuenta de usuario a través de la opción “Cree su cuenta”, la cual le redirigirá a otra interfaz, tal como se muestra en la Ilustración 17.

Admisión 2016
Sistema de Ingreso por Mérito Académico y Diagnóstico

Ingrese sus datos personales

Datos Personales

Cédula del aspirante <input type="text" value="Cédula"/>	Correo electrónico <input type="text" value="Introduzca su correo."/>	Confirmar correo electrónico <input type="text" value="Confirme su correo."/>
Nombre del aspirante <input type="text" value="Introduzca su nombre."/>	Apellido del aspirante <input type="text" value="Introduzca su apellido."/>	Género <input type="text" value=""/>
Código <input type="text" value="0412"/>	Teléfono <input type="text" value="xxxxxxxx"/>	Fecha de Nacimiento <input type="text" value="dd/mm/yyyy"/>
Tipo de Bachillerato <input type="text" value="Lista"/>	Nombre de la Institución <input type="text" value="Introduzca el nombre de su institución."/>	

Ubicación geográfica del aspirante

Indique su estado <input type="text" value="Seleccione"/>	Municipio <input type="text" value="Seleccione"/>	Parroquia <input type="text" value="Seleccione"/>
--	--	--

Ubicación geográfica de la institución

Indique su estado <input type="text" value="Seleccione"/>	Municipio <input type="text" value="Seleccione"/>	Parroquia <input type="text" value="Seleccione"/>
--	--	--

Universidad Central de Venezuela
 Ciudad Universitaria, Los Chaguaramos / Caracas, Venezuela, Código Postal: 1050
 RIF: G-20000062-7

Ilustración 17: Interfaz para crear cuenta.

En esta interfaz el aspirante deberá llenar el formulario ingresando todos los datos que se solicitan en el formulario, los cuales son datos personales, de la ubicación geográfica del aspirante y la ubicación geográfica de la institución donde curso bachillerato. Si desea ir a la interfaz anterior (Ilustración 16) puede seleccionar la opción de “Regresar”, pero en caso contrario, debe seleccionar la opción “Crear” luego de verificar que los datos suministrados son correctos.

La opción de “Crear”, luego de superar el proceso de validación, redirige al aspirante a la interfaz de inicio de sesión y le notifica que su registro es exitoso, tal como se muestra en la Ilustración 18.

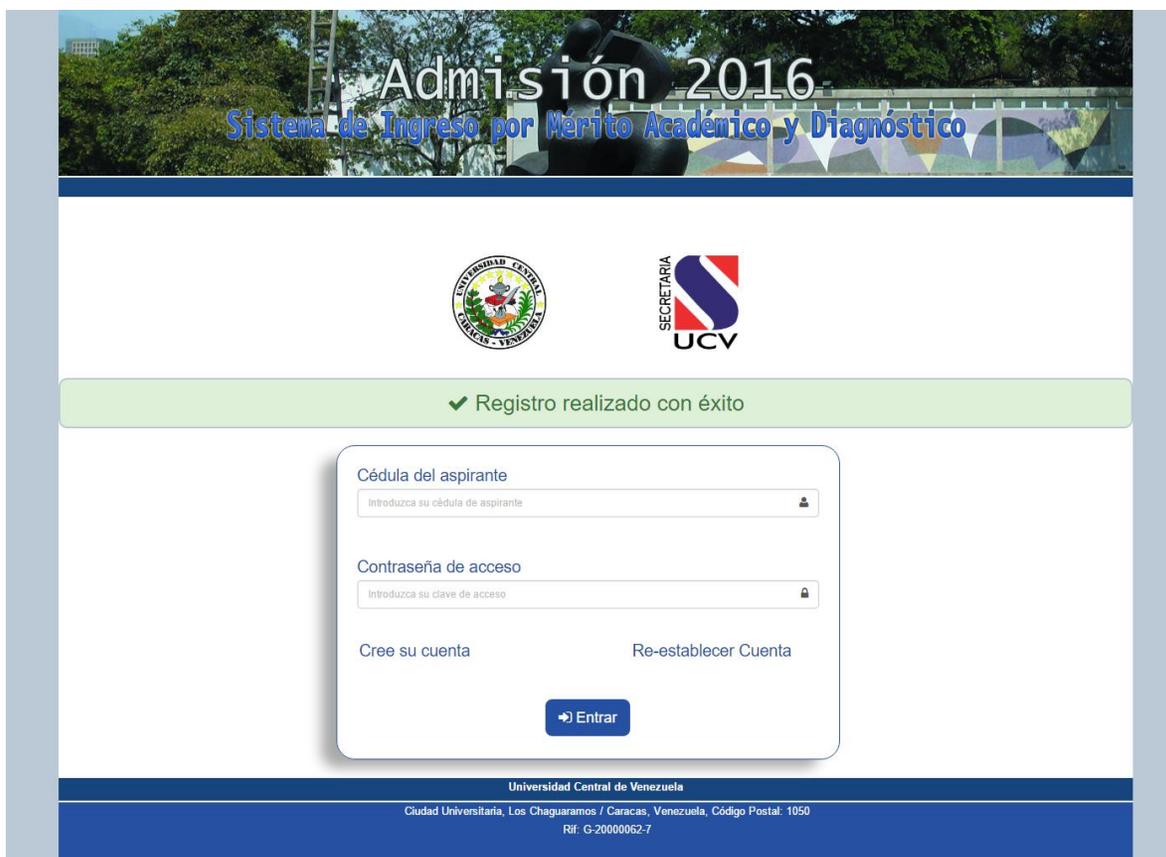


Ilustración 18: Interfaz de registro exitoso.

Al mismo tiempo que se muestra esta interfaz, se le envía un correo al aspirante para activar su registro. El correo que se envía posee la siguiente información reflejada en la Interfaz 19:

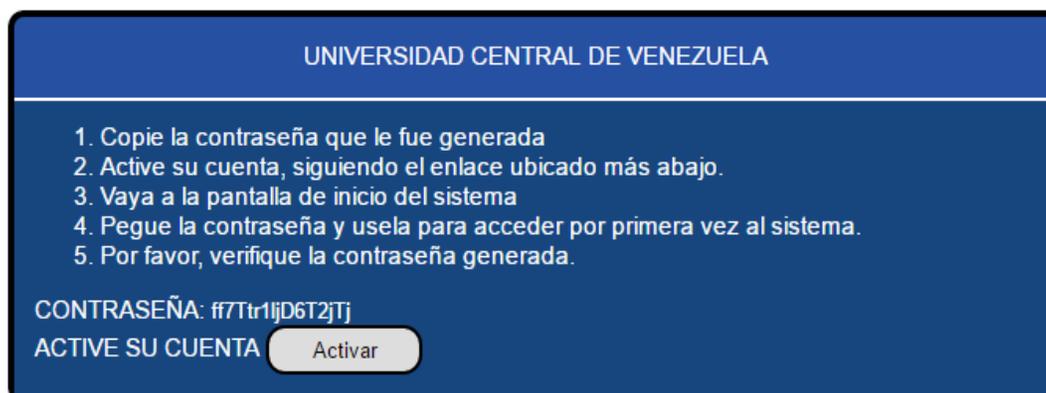


Ilustración 19: Correo enviado al aspirante para activar su registro.

Funcionalidad de Carga de Notas Académicas y Selección de Carreras

Una vez activada la cuenta de usuario, el sistema le muestra al aspirante una interfaz donde se muestran un conjunto de pasos. El paso número uno (1) se dispone para cargar las notas académicas que obtuvo el aspirante en bachillerato, desde el 1er a 4to año. Esta carga la puede hacer de dos formas: automática o manual.

- **Carga Automática:** Para la carga de notas académicas de forma automatizada, el aspirante debe cargar un archivo con extensión PDF con todas sus notas y cuyo modelo debe ser similar a la imagen de referencia que muestra el sistema (Ilustración 20).

The screenshot shows the 'Admisión 2016' interface for the 'Sistema de Ingreso por Mérito Académico y Diagnóstico'. The user is logged in as 'Gary' (C.I. V-19398144, Aspirante). A navigation bar includes 'Inicio', 'Mi Perfil', 'Inscripción', 'Regresar', and 'Salir'. A yellow warning banner states: 'Solo dispone del día de hoy para confirmar su proceso'. The main content area features a progress indicator with 'Paso N° 1' selected, 'Paso N° 2', and 'Paso N° 3'. Below this, the user is prompted to 'Seleccione un tipo de Carga:' with two options: 'PDF' (selected) and 'MANUAL'. The 'PDF' option is further detailed with a 'Seleccionar archivo' button (showing 'Ningún archivo cargado') and a 'Cargar su notas por pdf' button. A 'Modelo de P.D.F. a cargar en el sistema' is provided as a reference, showing a sample PDF form with a barcode and various data fields. The footer contains the university's name and contact information.

Ilustración 20: Carga de notas académicas de forma automática.

En este punto el sistema recibe el archivo y lo procesa para cargar las notas académicas de acuerdo a las materias cursadas por año, dando como resultado lo presentado en la Ilustración 21.

The screenshot displays the 'Admisión 2016' system interface. At the top, there is a navigation bar with links for 'Inicio', 'Mi Perfil', 'Inscripción', 'Regresar', and 'Salir'. Below this, a user profile for 'Gary' (C.I. V-19398144) is shown as an 'Aspirante'. A yellow warning box states: 'Solo dispone del día de hoy para confirmar su proceso'. The main content area features a progress indicator with three steps: 'Paso N° 1' (active), 'Paso N° 2', and 'Paso N° 3'. Below the progress indicator, there are buttons for 'PDF' and 'MANUAL'. The user's name 'Gary Jerry Romero Bautista' and ID 'C.I.V - 19398144' are displayed. A file selection area shows 'Seleccionar archivo' with the text 'Ningún arc...ccionado' and a 'Cargar su notas por pdf' button. A table with tabs for 'Primer Año', 'Segundo Año', 'Tercero Año', and 'Cuarto Año' is shown, with a 'Confirmar' button to the right. The table lists the following subjects and grades:

Materia	Notas
CASTELLANO Y LITERATURA	15
INGLÉS	17
MATEMÁTICA	14
ESTUDIOS DE LA NATURALEZA	15
HISTORIA DE VENEZUELA	17
EDUCACIÓN FAMILIAR Y CIUDADANA	19
GEOGRAFÍA GENERAL	16
EDUCACIÓN ARTÍSTICA	18
EDUCACIÓN FÍSICA Y DEPORTE	16
EDUCACIÓN PARA EL TRABAJO	16

At the bottom of the page, the footer contains the text: 'Universidad Central de Venezuela', 'Ciudad Universitaria, Los Chaguaramos / Caracas, Venezuela, Código Postal: 1050', and 'Rif: G-20000062-7'.

Ilustración 21: Listado de las notas académicas obtenidas del archivo PDF proporcionado.

- Carga Manual:** En esta opción, el aspirante por año académico debe indicar una a una las materias y notas asociadas a ellas para agregarlas en el listado de notas que guardará el sistema (Ilustración 22). En caso de ingresar algún dato incorrecto, la aplicación le permite al aspirante eliminar la materia con la nota incorrecta y volver a agregarla de forma adecuada.

The screenshot shows the 'Admisión 2016' system for 'Sistema de Ingreso por Mérito Académico y Diagnóstico'. The user is logged in as Gary (C.I. V-19398144, Aspirante). A warning message states: 'Solo dispone del día de hoy para confirmar su proceso'. The interface is divided into three steps: 'Paso N° 1', 'Paso N° 2', and 'Paso N° 3'. Under 'Paso N° 2', the user is prompted to 'Seleccione un tipo de Carga:' with options for 'PDF' and 'MANUAL'. The 'MANUAL' option is selected. Below this, the user has selected the subject 'EDUCACIÓN FAMILIAR Y CIUDADANA' with a grade of '15'. A table shows the current list of entered grades:

Materia	Notas	Opciones
ALEMÁN	18	[Eliminar]
CIENCIAS BIOLÓGICAS	18	[Eliminar]
EDUCACIÓN FAMILIAR Y CIUDADANA	15	[Eliminar]

Buttons for '+ Agregar' and 'Confirmar' are visible. The footer contains the university's name and contact information.

Ilustración 22: Interfaz de carga de notas académicas de forma manual.

Una vez realizada cualquiera de estas dos cargas, el aspirante puede seleccionar la opción “Confirmar”, si está de acuerdo con todo. Asimismo, la aplicación por detrás procesa las notas y almacena en base de datos el promedio del aspirante, para luego avanzar al siguiente paso del sistema, donde se redirecciona la aplicación a la interfaz que comprende el paso numero dos (2) del proceso y le solicita al aspirante cuál es el área a la que desea postularse y las carreras por facultad que quiere cursar (Ilustración 23).

Admisión 2016
Sistema de Ingreso por Mérito Académico y Diagnóstico

Inicio Mi Perfil Inscrición Regresar Salir

Gary
C.I. V-19398144
Aspirante

Solo dispone del día de hoy para confirmar su proceso

Paso N° 1 Paso N° 2 Paso N° 3

Indique su área: Ciencia y Tecnología

Indique la facultad: Ciencias

Indique la carrera: Matemática

+ Agregar Limpiar

Opción	Área	Facultad	Carrera	Opciones
1	Ciencia y Tecnología	Ciencias	Computación	↓ ☰
2	Ciencia y Tecnología	Ciencias	Física	↓ ☰
3	Ciencia y Tecnología	Ciencias	Matemática	↓ ☰

Confirmar

Universidad Central de Venezuela
Ciudad Universitaria, Los Chaguaramos / Caracas, Venezuela, Código Postal: 1050
RIF: G-2000062-7

Ilustración 23: Ingreso de las carreras a las se postula el aspirante.

En este interfaz el aspirante puede seleccionar solo una de las cuatro (4) áreas de conocimiento existentes y tres (3) carreras que pueden pertenecer a cualquier facultad que integre el área seleccionada. Luego de esto, el aspirante puede escoger la opción de “Continuar”; opción que redirige al usuario a la interfaz del paso número tres (3). El paso número tres (3), le muestra al usuario una interfaz con una vista previa de todos los datos proporcionados anteriormente, tal como se observa en la Ilustración 24.

Admisión 2016
Sistema de Ingreso por Mérito Académico y Diagnóstico

Salir

Gary Romero
 C.I. V-19398144

Datos personales en el sistema

- C.I.V. -19398144
- Aspirante: Gary Romero
- garyromero@gmail.com
- 0414-1536807
- Masculino
- 1988-04-30

Datos de ubicación en el sistema

Ubicación del aspirante

- Distrito_Capital
- Libertador
- La_Pastora

Ubicación de la institución

- Distrito_Capital
- Libertador
- La_Pastora

Carreras seleccionadas

Código	Carrera
10001	Física
10002	Matemática
10011	Computación

Notas cargadas

Primer Año Segundo Año Tercero Año Cuarto Año

Materia	Notas
CASTELLANO Y LITERATURA	15
INGLÉS	17
MATEMÁTICA	14
ESTUDIOS DE LA NATURALEZA	15
HISTORIA DE VENEZUELA	17
EDUCACIÓN FAMILIAR Y CIUDADANA	19
GEOGRAFÍA GENERAL	16
EDUCACIÓN ARTÍSTICA	18
EDUCACIÓN FÍSICA Y DEPORTE	16
EDUCACIÓN PARA EL TRABAJO	16

Universidad Central de Venezuela
 Ciudad Universitaria, Los Chaguaramos / Caracas, Venezuela, Código Postal: 1050
 Rf: G-2000062-7

Ilustración 24: Resumen de los datos proporcionados por el aspirante.

Además, hay que resaltar que, si el aspirante ya se encontraba registrado en el sistema y también ya había cargado sus notas académicas, la interfaz anterior (Ilustración 24), es lo que visualiza al acceder, siempre y cuando, permanezca en vigencia la fase de registro.

4.4. Iteración 4: Prueba Vocacional con el Cálculo y Publicación de los Resultados Obtenidos.

En esta iteración se desarrollan las funcionalidades asociadas a la presentación de una prueba vocacional y la selección de los aspirantes de acuerdo a los resultados obtenidos de la prueba y el promedio de las notas académicas, calculado previamente en la iteración anterior.

Esta prueba es una herramienta psicológica que busca identificar las potencialidades de una persona, por tanto, permite obtener un perfil aproximado de los intereses, habilidades destacadas y áreas ocupacionales que más se relacionan con el aspirante, a fin de conocer que tan acorde está con las carreras que seleccionó cuando se registró. Si el resultado obtenido en esta prueba está relacionado a las carreras que eligió, entonces el sistema le asigna una ponderación al aspirante, en caso de no estarlo, entonces no se le asigna ninguna ponderación.

De igual forma, durante esta iteración también se siguen manteniendo reuniones con el equipo de desarrollo del proyecto en las que se discuten aspectos importantes a desarrollar, siendo algunos de ellos, los siguientes:

- El modelo de prueba vocacional debe ser realizado por una persona capacitada para ello, por tal motivo, se le solicita a un profesional en psicología certificado por la universidad para crear dicho modelo.
- Las preguntas son mostradas al aspirante en bloque de cinco (5), ya que esto evita que quien responda la prueba se sobrecargue con tantas preguntas.
- Las respuestas son comprendidas por una selección simple, donde el aspirante debe contestar a la pregunta si le gusta, no le gusta, le es indiferente o tiene dudas y no conozco la actividad o profesión.
- En base a la respuesta que arroja como resultado la prueba, el sistema la contrasta contra las carreras seleccionadas por el aspirante y guardan una relación entonces se asigna una ponderación al aspirante.

- Procesar la información suministrada por el aspirante en la prueba.
- Evaluar la ponderación acumulada entre el promedio del aspirante y lo obtenido en la prueba para seleccionar a los más destacados.
- Publicar los resultados indicando si el estudiante es seleccionado o no para la siguiente fase (fase de verificación).
- Redirigir al usuario cuando acceda al sistema, a la interfaz de inicio para comenzar a ejecutar la prueba.
- Mantener la consistencia visual en todas las interfaces.
- La definición de las metáforas utilizadas para identificar las diferentes funciones.
- Incorporar textos de ayuda en las interfaces.

Para esta iteración los resultados obtenidos se muestran a continuación.

Funcionalidad de Prueba Vocacional

En esta fase, el sistema le muestra al aspirante una interfaz inicial en donde se le informa lo que debe tener en cuenta para presentar adecuadamente la prueba vocacional, en conjunto con la opción “Ir a Test” que debe seleccionar el aspirante para comenzar dicha prueba (Ilustración 25).

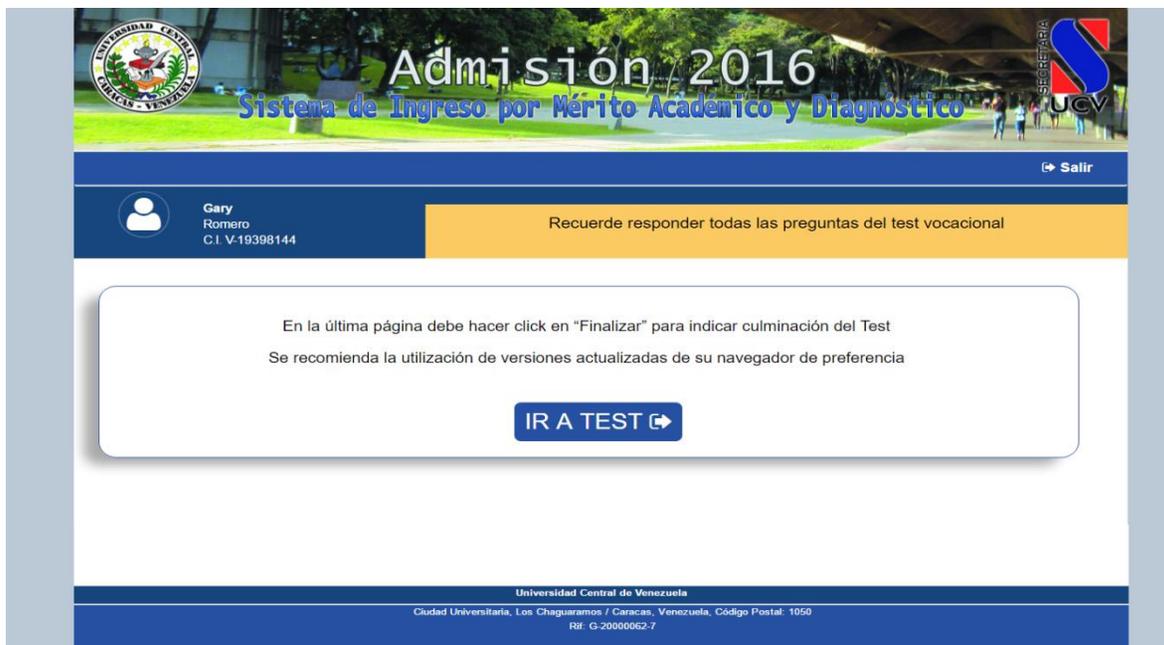


Ilustración 25: Interfaz de inicio para la prueba vocacional.

Luego de eso, la aplicación le presenta al aspirante una serie de preguntas en bloques de cinco (5), las cuales debe contestar a través de una selección simple (Ilustración 26). Esta prueba consta de un número finito de preguntas y la duración de ejecución de la misma, lo determina el tiempo de vigencia que se define para esta fase. Además, las preguntas reflejadas no son las mismas para todos los aspirantes, ya que el sistema va seleccionando aleatoriamente, una a una las preguntas a mostrar entre un gran número de preguntas que dispone el sistema en base de datos.

Admisión 2016
Sistema de Ingreso por Mérito Académico y Diagnóstico

Salir

Gary
Romero
C.I. V-1

A - "ME GUSTA"
B - "NO ME GUSTA"
C - "ME ES INDIFFERENTE O TENGO DUDAS"
D - "NO CONOZCO LA ACTIVIDAD O LA PROFESIÓN"

1.- Investigar y experimentar en el campo de la mecánica, la óptica, la física nuclear, etc
A B C D

2.- Estudiar la composición y estructura de la atmósfera y los astros (planetas, satélites, etc)
A B C D

3.- Trabajar como analista clínico (Bionalista)
A B C D

4.- Proyectar y dirigir la construcción de edificios. Preparar los planos de edificios o zonas completas de la ciudad (barrio, parques, zonas comerciales, etc.).
A B C D

5.- Ser arquitecto/a
A B C D

• Pagina: 1 de 3

Anterior Siguiete

Universidad Central de Venezuela
Ciudad Universitaria, Los Chaguaramos / Caracas, Venezuela, Código Postal: 1050
Rif: G-20000062-7

Ilustración 26: Interfaz de prueba vocacional.

Después de contestar todas las preguntas, el aspirante debe escoger la opción de “Finalizar” para que todos los datos sean procesados y cargados en base de datos, según los criterios de evaluación definidos para esta prueba, seguido por un mensaje que la aplicación le muestra al aspirante notificándole que sus resultados están disponibles a partir de una fecha establecida previamente.

Funcionalidad de Selección de Aspirantes

Partiendo de lo anterior y llegada la fecha que indicó previamente el sistema al aspirante, la aplicación comienza a sumar por cada uno de ellos, el promedio de notas y la ponderación obtenida en la prueba vocacional, para posteriormente ordenar descendientemente todos esos resultados, producto de la suma, y así seleccionar los N aspirantes que continúan con el proceso de ingreso a la universidad.

Al finalizar ese proceso de selección, la aplicación presenta una interfaz como la que se observa en la Ilustración 27, para que el aspirante ingrese su número de cédula de identidad y conozca si fue o no, seleccionado para continuar con el proceso.

UNIVERSIDAD CENTRAL DE VENEZUELA

Admisión 2016

Sistema de Ingreso por Mérito Académico y Diagnóstico

SECRETARIA UCV

RESULTADOS DE LA ADMISION

Cédula del aspirante

Pasos para verificar ADMISION

1. Ingrese su cédula de identidad
2. Haga click en el boton Buscar

INFORMACIÓN IMPORTANTE

Si usted fue Admitido, dirigase al siguiente enlace
!!INFORMACIÓN!!

Universidad Central de Venezuela
Ciudad Universitaria, Los Chaguaramos / Caracas, Venezuela, Código Postal: 1050
RIF: G-2000062-7

Ilustración 27: Interfaz de publicación de los resultados de los aspirantes seleccionados.

No obstante, si el aspirante ha sido seleccionado debe ingresar al enlace que muestra la misma Ilustración 27 para conocer los recaudos necesarios que debe consignar en la próxima fase.

4.5. Iteración 5: Validación de Recaudos y Publicación de la Selección de los Aspirantes.

En esta iteración, se implementan las funcionalidades referentes a la verificación de recaudos que debe presentar el aspirante en la universidad, ante un operador asociado a la facultad de la carrera que desea cursar para la cual fue seleccionado en la fase anterior.

Como se acaba de hacer referencia, para esta validación interviene un operador. Este operador recibe los recaudos en físico del aspirante y accede al sistema de ingreso para validar que toda la información que el aspirante suministro, es correcta y válida con la finalidad de registrar sus observaciones en el sistema. Sin embargo, luego de lo anterior, interviene un supervisor, quien ingresa al sistema también y evalúa todas las observaciones realizadas por el operador a fin de escoger los aspirantes que cumplen con todas las condiciones establecidas por la universidad y pasan a la próxima fase.

Los aspectos más resaltantes para esta fase son:

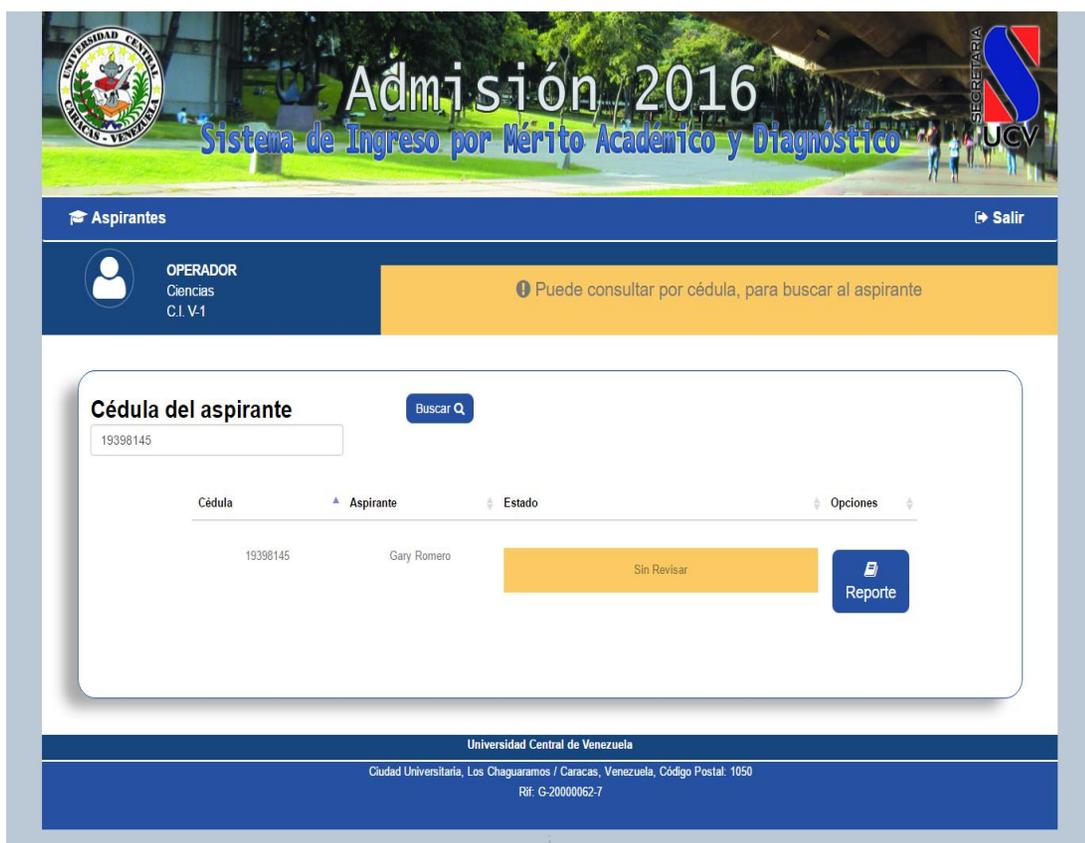
- Permitir que en el sistema participe personal administrativo de la universidad cumpliendo el rol de operador y supervisor respectivamente.
- Permitir que el operador y supervisor busquen a un aspirante de acuerdo a su número de cédula de identidad.
- Mostrar toda la información asociada al aspirante que se está verificando, es decir, datos personales y notas académicas.
- Permitir que el operador ingrese sus observaciones y consideraciones resultantes de la verificación de recaudos en el sistema.
- Permitir que el supervisor evalúe las observaciones o consideraciones proporcionadas por el operador hacia un aspirante.
- Mantener la consistencia visual en todas las interfaces.
- Publicar los resultados indicando si el estudiante es seleccionado o no para la siguiente fase (fase de prueba diagnóstica).
- La definición de las metáforas utilizadas para identificar las diferentes funciones.
- Incorporar textos de ayuda en las interfaces.

Los resultados obtenidos de esta iteración se muestran a continuación.

Funcionalidad de Verificación

Las actividades que comprenden esta funcionalidad son llevadas a cabo básicamente por dos actores perteneciente a una determinada facultad. Estos actores son: el operador y el supervisor.

- **Operador:** Ingresa al sistema y recibe los documentos que le lleva el aspirante. A través de la aplicación, el operador coloca el número de cédula de identidad del aspirante y selecciona la opción de “Buscar” para que el sistema realice la búsqueda de los datos pertenecientes a dicho aspirante seleccionado en la fase anterior.



The screenshot displays the 'Admisión 2016' interface for the 'Sistema de Ingreso por Mérito Académico y Diagnóstico' at the Universidad Central de Venezuela (UCV). The header includes the UCV logo and the text 'SECRETARIA UCV'. Below the header, there is a navigation bar with 'Aspirantes' and 'Salir' options. The main content area shows the user role as 'OPERADOR' in the 'Ciencias' faculty with ID 'C.I. V-1'. A yellow notification bar states: 'Puede consultar por cédula, para buscar al aspirante'. The search section, titled 'Cédula del aspirante', features a search box containing '19398145' and a 'Buscar' button. Below this is a table with columns for 'Cédula', 'Aspirante', 'Estado', and 'Opciones'. The table contains one entry for '19398145' and 'Gary Romero', with a 'Sin Revisar' status and a 'Reporte' button. The footer contains the university's name and contact information: 'Universidad Central de Venezuela, Ciudad Universitaria, Los Chaguaramos / Caracas, Venezuela, Código Postal: 1050, RIF: G-20000062-7'.

Ilustración 28: Buscador de aspirantes.

El sistema encuentra el registro del aspirante y el operador debe seleccionar la opción “Reporte” para visualizar todos los datos suministrados por dicho aspirante en la fase de registro, tal como lo refleja la Ilustración 29, con el objetivo de validar la información que tiene el sistema con la reflejada en los recaudos entregados en físico.

Admisión 2016
Sistema de Ingreso por Mérito Académico y Diagnóstico

Aspirantes Salir

OPERADOR
Ciencias
C.I. V-1

Puede consultar por cédula, para buscar al aspirante

Datos Personales

Cédula: 19398145 | Correo: garyrb@gmail.com

Nombre del aspirante: Gary | Apellido del aspirante: Romero

Género: M | Fecha de Nacimiento: 2004-10-27 | Código: 0414 | Teléfono: 1538807

Tipo de Bachillerato: CIENCIAS | Nombre de la Institución: Santos Michelena

Ubicación geográfica del aspirante

Indique su estado: Distrito Capital | Municipio: Libertador | Parroquia: La Pastora

Ubicación geográfica de la institución

Indique su estado: Distrito Capital | Municipio: Libertador | Parroquia: La Pastora

Carreras seleccionadas

Código	Carrera
10002	Matemática
10011	Computación
10240	Ingeniería Civil (Cagua)

Notas cargadas

Materia	Notas
CASTELLANO Y LITERATURA	11
CIENCIAS BIOLÓGICAS	15
MATEMÁTICA	20

Observaciones

Cédula
 Notas certificadas
 Constancia de 5to año
 Flotante
 OPSU

Universidad Central de Venezuela
 Ciudad Universitaria, Los Chaguaramos / Caracas, Venezuela, Código Postal: 1050
 Ref: G-20000062-7

Ilustración 29: Interfaz del reporte con los datos registrados del aspirante seleccionado.

Luego de revisar los datos reflejados en el reporte, éste le proporciona al operador la posibilidad de agregar alguna observación que haya surgido con base en la verificación de dichos datos. Si todos los datos están bien, entonces el operador selecciona la opción de “Correcto”, en caso contrario, selecciona la opción “Incorrecto”.

Partiendo de lo anterior, el sistema internamente coloca el registro del aspirante en estado de “Revisado” y redirecciona al operador a la interfaz que comprende al buscador de aspirantes para que realice la búsqueda de un nuevo aspirante sin mostrar los datos del aspirante anterior porque ya fue verificado (Ilustración 30).

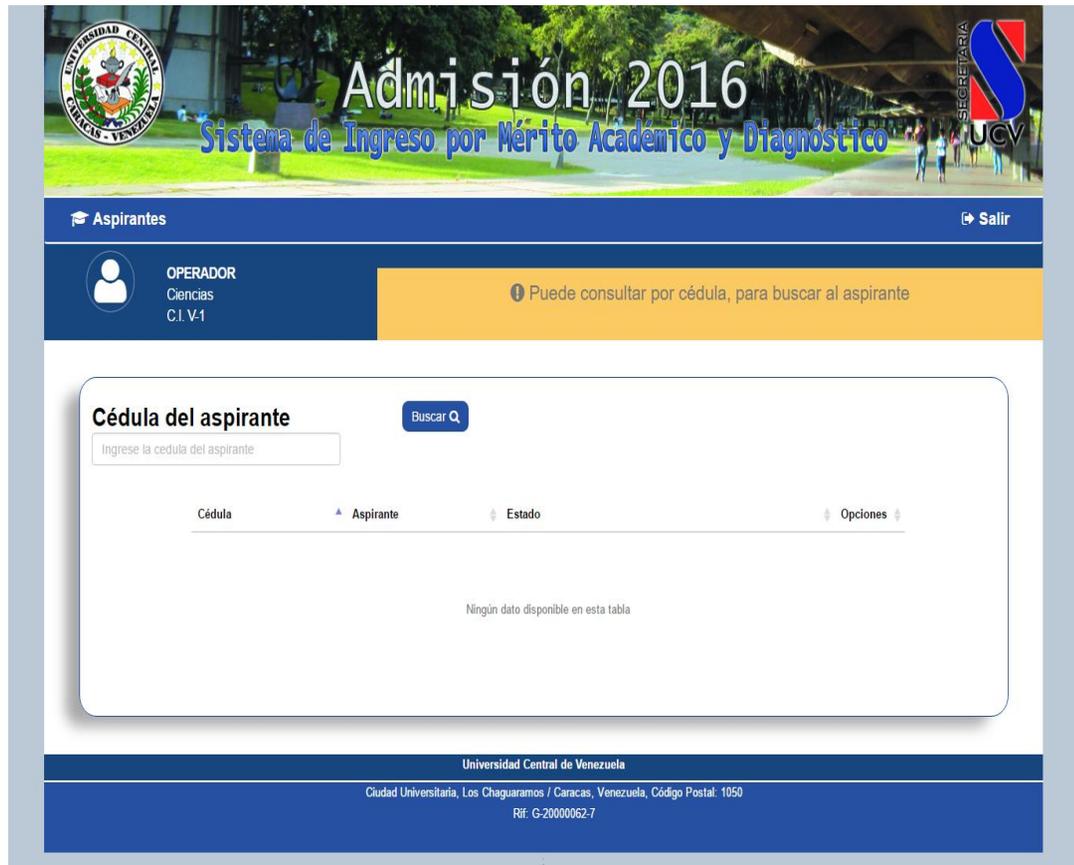


Ilustración 30: Interfaz limpia para buscar un nuevo aspirante.

Sin embargo, en caso de que el operador busque un aspirante que ya fue verificado, el sistema le indica a dicho operador, que el estado del registro del aspirante se encuentra en revisado y no lo puede modificar.

- **Supervisor:** Después de que el operador revise los datos de los aspirantes y coloque su observación, el supervisor ingresa al sistema para evaluar el estado del registro de todos los aspirantes seleccionados en la fase previa tal como se muestra en la Ilustración 31.

Admisión 2016
Sistema de Ingreso por Mérito Académico y Diagnóstico

Aspirantes Procesados Salir

COORDINADOR
Ciencias
C.I. V-2

Puede consultar por cédula, para buscar al aspirante

Buscar

Cédula	Aspirante	Estado	Opciones
19398144	Gary Romero	Correcto	Reporte Aprobar Rechazar
19398145	Gary Romero	Incorrecto	Reporte Aprobar Rechazar
19398146	Gary Romero	Sin Revisar	Reporte Aprobar Rechazar

Universidad Central de Venezuela
Ciudad Universitaria, Los Chaguaramos / Caracas, Venezuela, Código Postal: 1050
RIF: G-20000062-7

Ilustración 31: Listado de los registros de los aspirantes seleccionados en la fase anterior por estado.

En esta interfaz (Ilustración 31), el sistema le proporciona al supervisor, por cada registro listado, las opciones de “Reporte”, “Aprobar” y “Rechazar”. Si selecciona la opción “Reporte” de algún aspirante, el sistema presenta la interfaz que muestra todos los datos registrados de ese aspirante, con el objetivo de que los vea y evalúe las observaciones realizadas por el operador en caso de tenerlas, así como lo refleja la Ilustración 32.



Aspirantes Procesados Salir


COORDINADOR
 Ciencias
 C.I. V-2

Puede consultar por cédula, para buscar al aspirante

Datos Personales

Cédula: 19398145 | Correo: garyrb@gmail.com

Nombre del aspirante: Gary | Apellido del aspirante: Romero

Género: M | Fecha de Nacimiento: 2004-10-27 | Teléfono: 0414-1536807 | Tipo de Bachillerato: CIENCIAS

Nombre de la Institución: Santos Michelena

Ubicación geográfica del aspirante

Indique su estado: Distrito_Capital | Municipio: Libertador | Parroquia: La_Pastora

Ubicación geográfica de la institución

Indique su estado: Distrito_Capital | Municipio: Libertador | Parroquia: La_Pastora

Carreras seleccionadas

Código	Carrera
10002	Matemática
10011	Computación
10240	Ingeniería Civil (Cagua)

Notas cargadas

Primer Año		Segundo Año		Tercero Año		Cuarto Año	
Materia	Notas						
CASTELLANO Y LITERATURA	11						
CIENCIAS BIOLÓGICAS	15						
MATEMÁTICA	20						

Observaciones: esto es un reporte
**FALTA NOTAS CERTIFICADAS

Ilustración 32: Reporte con datos del registro de un aspirante y observaciones hechas por el operador.

Luego de evaluar las observaciones, el sistema le permite al supervisor indicar si aprueba o no al aspirante, mediante el uso de las opciones de “Aprobar” / “Rechazar” que se muestran tanto en la interfaz que carga el reporte (Ilustración 32), como en la interfaz con el listado de registros de los aspirantes por estado (Ilustración 31). Asimismo, después de la decisión tomada por el supervisor, la aplicación le muestra un listado final (Ilustración 33) con los aspirantes que han sido aprobados ó no para la siguiente fase (prueba diagnóstica).

The screenshot displays the 'Admisión 2016' system interface. At the top, there is a banner with the university logo and the text 'Admisión 2016 Sistema de Ingreso por Mérito Académico y Diagnóstico'. Below the banner is a navigation bar with 'Aspirantes' and 'Procesados' tabs, and a 'Salir' button. A user profile section shows 'COORDINADOR Ciencias C.I. V-2'. A search bar contains the text 'Puede consultar por cédula, para buscar al aspirante'. The main content area features a table with the following data:

Cédula	Nombre	Apellido	Estado
19398145	Gary	Romero	APROBADO
19398146	Gary	Romero	RECHAZADO

At the bottom of the interface, there is a footer with the text: 'Universidad Central de Venezuela Ciudad Universitaria, Los Chaguaramos / Caracas, Venezuela, Código Postal: 1050 Rif: G-20000062-7'.

Ilustración 33: Listado de aspirantes aprobados o no para la siguiente fase.

Funcionalidad de Selección de Aspirantes

Una vez culminada la fase de verificación, el sistema se habilita para presentarle al aspirante una interfaz como la que se observa en la Ilustración 34, para que ingrese su número de cédula de identidad y conozca si fue seleccionado ó no, para seguir adelante con la siguiente fase (prueba diagnóstica).



Ilustración 34: Interfaz de publicación de los resultados de los aspirantes seleccionados.

4.6. Iteración 6: Prueba Diagnóstica con el Cálculo y Publicación de los Resultados Obtenidos.

En esta iteración se desarrollan las funcionalidades asociadas a la presentación de una prueba diagnóstica dividida en dos partes (verbal y lógica matemática), y la selección de los aspirantes de acuerdo a los resultados obtenidos en dicha prueba.

La prueba diagnóstica busca identificar las potencialidades de una persona en base a su capacidad verbal y lógica matemática. Durante esta iteración se llevan a cabo un conjunto de reuniones con el equipo de desarrollo del proyecto en las que se discuten los aspectos importantes a implementar en el sistema para esta fase, siendo algunos de ellos, los siguientes:

- Los modelos de pruebas diagnósticas, tanto verbales como lógicos matemáticos son realizados por una persona capacitada para ello, por tal motivo, se le solicita a un profesional certificado por la universidad para crear dichos modelos.
- Cada prueba (verbal y lógica matemática), tiene una duración de 45 minutos.
- Cada prueba (verbal y lógica matemática), posee varios modelos cargados en base de datos.
- Establecer un algoritmo que permita aleatoriamente seleccionar un modelo y preguntas de forma aleatoria con el objetivo de evitar que varios estudiantes presenten el mismo examen.
- Desplegarle al usuario un reloj que contabilice el tiempo que le queda al aspirante para responder toda la prueba.
- Las respuestas son comprendidas por una selección simple.
- Definir varios bloques de horas para la presentación de esta prueba.
- Procesar las respuestas suministradas por el aspirante en la prueba.
- Llevar la bitácora de actividades que realiza el aspirante en el sistema durante la prueba.
- Evaluar la ponderación obtenida por todos los aspirantes en la prueba diagnóstica, para seleccionar a los más destacados.
- Publicar los resultados indicando si el estudiante es seleccionado o no, para proceder a inscribirse e ingresar a la universidad.
- Redirigir al aspirante cuando acceda al sistema, a la interfaz de inicio para comenzar a ejecutar la prueba.
- Mantener la consistencia visual en todas las interfaces.
- La definición de las metáforas utilizadas para identificar las diferentes funciones.
- Incorporar textos de ayuda en las interfaces.

Para esta iteración los resultados obtenidos se muestran a continuación.

Funcionalidad de Prueba Diagnóstica

En esta fase, el sistema le muestra al aspirante una interfaz inicial en donde se le informa lo que debe tener en cuenta para presentar adecuadamente la prueba, tanto verbal como lógica matemática, en conjunto con la opción “Ir a Test” que debe seleccionar para comenzar a presentar dicha prueba (Ilustración 35).

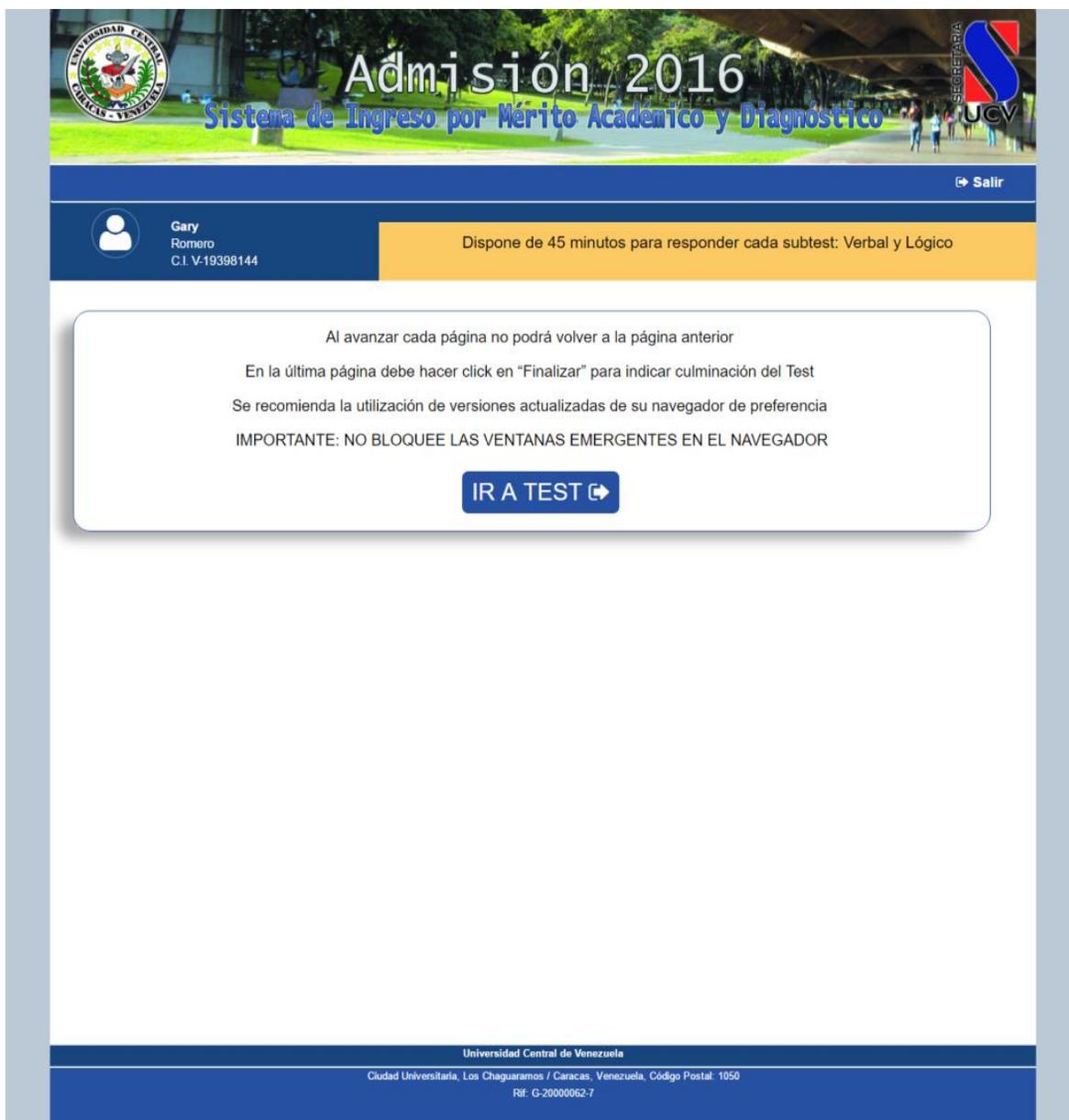


Ilustración 35: Interfaz de inicio para la prueba diagnóstica.

Luego de eso, la aplicación le presenta al aspirante una serie de preguntas. Primero, preguntas asociadas a la prueba verbal cuya duración comprende 45 minutos (Ilustración 36), y luego, preguntas de la prueba lógica matemática que también tiene una duración de 45 minutos (Ilustración 37). Estas dos pruebas que conforman toda la prueba diagnóstica, constan de un número finito de preguntas las cuales no son iguales para todos los aspirantes, ya que el sistema va seleccionando aleatoriamente un modelo de prueba, y de ese modelo, va seleccionando una a una las preguntas a mostrar entre un gran número de preguntas que dispone el sistema en base de datos.



Admisión 2016

Sistema de Ingreso por Mérito Académico y Diagnóstico



Inicio

Salir



Gary
Romero
C.I. V-19398144

TIEMPO 44:44

Test Verbal

TEXTO 3: El iPhone está siendo utilizado para realizar ataques de Pharming

El lanzamiento en cientos de países del iPhone, el popular teléfono móvil de Apple, está siendo utilizado por los ciberdelincuentes como cebo con el que atraer a los usuarios para infectarlos con malware.

El pharming es una versión sofisticada del phishing. Consiste en cambiar los contenidos del DNS (Domain Name Server o Servidor de Nombres de Dominio) a través de la configuración del protocolo TCP/IP o del archivo host. Los DNS almacenan la dirección numérica o IP (62.14.63.187., por ejemplo) asociada a cada nombre de dominio o URL. El fraude consiste en que cuando se teclea el nombre de una página web, el servidor reenviará al usuario a otra dirección numérica, es decir, a otra IP que será una página fraudulenta, similar a la original.

Mientras engaña al usuario con esa página, el troyano modifica el fichero host redirigiendo las URLs de algunos bancos y empresas a una página falsa. De este modo, cuando un usuario intente visitar alguna de esas entidades tecleando su dirección en el navegador o accediendo a ellas desde una búsqueda en Internet, será redirigido a la página fraudulenta. En ella se le pedirán datos privados (número de cuenta bancaria, contraseña para operaciones bancarias online, etc.) que, de ser dados, irán a parar a manos del ciberdelincuente.

La modificación no provoca ningún comportamiento raro en el computador. Además el fraude se lleva a cabo sin que el usuario tenga que realizar ningún comportamiento extraño, puesto que basta con que teclee la dirección de su banco en el navegador para convertirse en víctima. Esto hace que el ataque sea aún más peligroso.

"Obviamente, el propósito de los ciberdelincuentes es utilizar esa información para saquear las cuentas de los usuarios y hacerse con su dinero", explica Luis Corrons, director de PandaLabs, que añade: "el iPhone actúa en este caso como cebo para atraer a los usuarios e invitarles a ejecutar el archivo que contiene el código malicioso".

Glenda González

23.-De acuerdo al texto, podría inferirse que la autora:

- a) compraría un iPhone.
- b) no utiliza celular.
- c) compraría un Nokia.
- d) no usaría un iPhone.

24.-Si el pharming ha contaminado un iPhone, lo más probable es que cuando el usuario ingrese en la página del Banco "El Buen Pastor", sus datos sean transferidos a:

- a) un malware
- b) un ciberdelincuente
- c) otro teléfono.
- d) otro banco

25.-La expresión "sin que el usuario tenga que realizar ningún comportamiento extraño", quiere decir, el usuario:

- a) realiza acciones no convencionales.
- b) actúa de manera habitual.
- c) se comporta de manera rara.
- d) es una persona normal.

26.-¿Qué relación se establece entre la autora y Luis Corrons?

- a) representan a Apple.
- b) los dos son periodistas
- c) él es su punto de apoyo.
- d) representan a PandaLab.

27.-El troyano es el:

- a) fichero
- b) usuario.
- c) ciberdelincuente
- d) pharming

28.-El propósito de la autora es:

- a) denunciar a Apple por código malicioso.
- b) dar instrucciones para ingresar en cuentas.
- c) prevenir sobre el manejo del celular.
- d) promocionar el uso del iPhone.

29.-Según el texto, puede deducirse que:

- a) algunas personas han sido estafadas.
- b) Apple mejorará el iPhone.
- c) el iPhone será sacado del mercado.
- d) PandaLab es competencia de Apple.

30.-La expresión "página fraudulenta" se asocia al concepto de página:

- a) web
- b) porno
- c) fantasma
- d) paralela.

Iniciar Test Lógico-numérico

Universidad Central de Venezuela

Ciudad Universitaria, Los Chaguaramos / Caracas, Venezuela, Código Postal: 1050
Rif: G.20000062.7

Ilustración 36: Interfaz de prueba verbal.

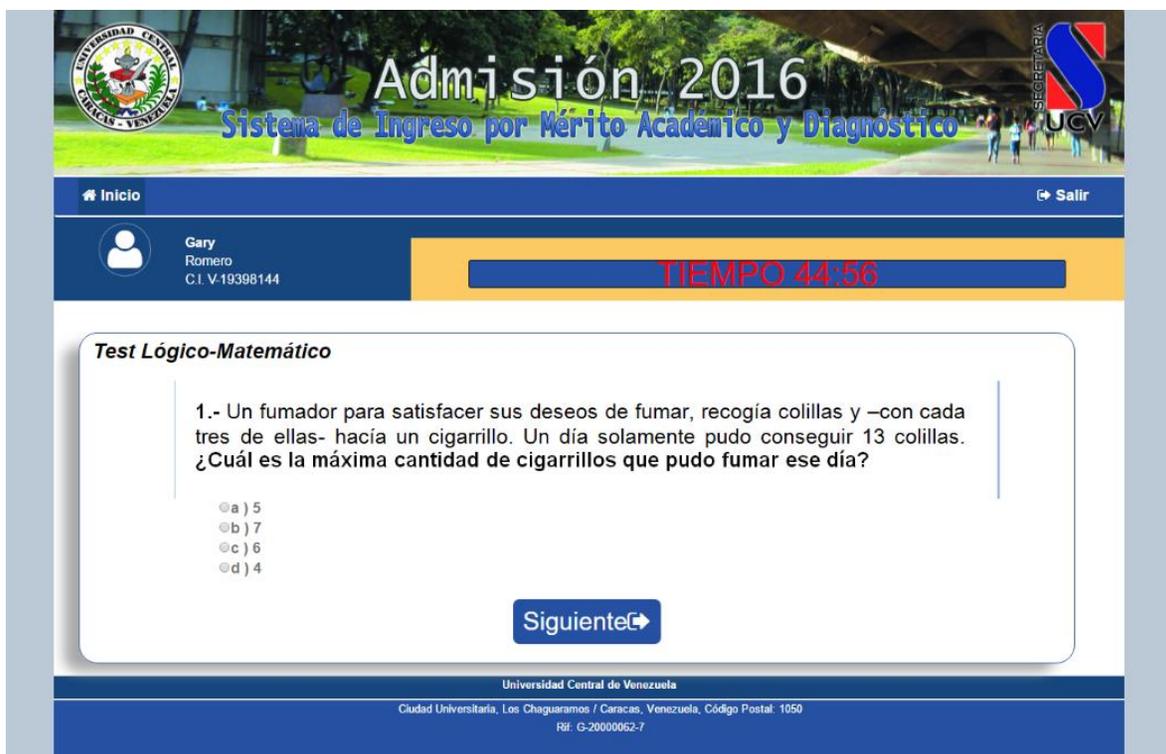


Ilustración 37: Interfaz de prueba lógica matemática.

Después de contestar todas las preguntas, el aspirante debe escoger la opción de “Finalizar” para que todos los datos sean procesados y cargados en base de datos, según los criterios de evaluación definidos para esta prueba, seguido por un mensaje que la aplicación le muestra al aspirante notificándole que sus resultados están disponibles a partir de una fecha determinada.

Funcionalidad de Selección de Aspirantes

Partiendo de lo anterior, y llegada la fecha que indicó previamente el sistema al aspirante, la aplicación comienza a ordenar descendentemente la ponderación obtenida en la prueba diagnóstica por los aspirantes, para seleccionar los N aspirantes con mejor ponderación ya que éstos serán los que pueden ingresar a la universidad.

Al culminar ese proceso de selección, la aplicación presenta una interfaz como la que se visualiza en la Ilustración 38, para que el aspirante ingrese su número de cédula de identidad y conozca si fue ó no, admitido en la universidad para estudiar la carrera que desea.



Ilustración 38: Interfaz de publicación de los resultados de los aspirantes seleccionados.

4.7. Iteración 7: Módulo de Administración.

Para el módulo de administración se desarrollan las funcionalidades asociadas a la gestión del sistema de ingreso en donde participa como actor del sistema, uno o varios administradores. Para ello, se mantienen reuniones con el equipo de desarrollo del proyecto en las que se presentan las versiones generadas durante esta iteración, a fin de comprobar y verificar los aspectos importantes que el sistema para esta parte, debe llevar a cabalidad.

Entre los aspectos más resaltantes para este módulo están:

- Crear periodos académicos para la ejecución de todo el proceso de ingreso mediante el sistema.
- Permitir que el administrador defina el tiempo por cada fase que el sistema abarca.
- Cargar y eliminar modelos de prueba diagnóstica.
- Mostrarle al administrador un conjunto de reportes basados en los datos obtenidos y proporcionados por los aspirantes.
- Permitir la visualización de la bitácora con el historial de actividades que realizan los operadores y supervisores dentro de la aplicación.
- Establecer la cantidad de cupos disponibles por carrera para la selección de los aspirantes.

- Gestionar la relación entre las materias de bachillerato y las diversas carreras que oferta la universidad.
- Mantener la consistencia visual en todas las interfaces.
- La definición de las metáforas utilizadas para identificar las diferentes funciones.

Los resultados obtenidos de esta iteración se muestran a continuación.

Funcionalidad de Nuevo Periodo

El sistema le proporciona al administrador una interfaz para que registre cada vez que se va a aplicar un nuevo proceso de ingreso, un periodo asociado a éste. Esta interfaz opera como una funcionalidad que permite volver a usar el sistema, con la finalidad de que todas las fases que comprenden el proceso se asocien a un período. Asimismo, el sistema garantiza mantener siempre los datos de los períodos anteriores.

Para realizar ese registro, la aplicación muestra la siguiente interfaz (Ilustración 39), donde el administrador debe ingresar los datos requeridos y luego seleccionar la opción “Agregar”.

Ilustración 39: Crear un nuevo periodo.

Funcionalidad de Gestión de Fases

Esta funcionalidad le permite al administrador gestionar el tiempo de vigencia de cada una de las fases que componen el proceso de ingreso, indicando la fecha de inicio y la fecha fin, dispuestas para cada fase.

Asimismo, de acuerdo a la línea de tiempo de ejecución de cada fase, el administrador puede llevar a cabo ciertas actividades, por ejemplo, luego de finalizar el tiempo de ejecución de la fase dos (2) de prueba vocacional, se pueden descargar los resultados de la evaluación para calcular los datos y luego ingresar ese procesamiento indicando los porcentajes que se le aplicaran al promedio de las notas. En la Ilustración 40, se puede visualizar la interfaz asociada a dicha funcionalidad.

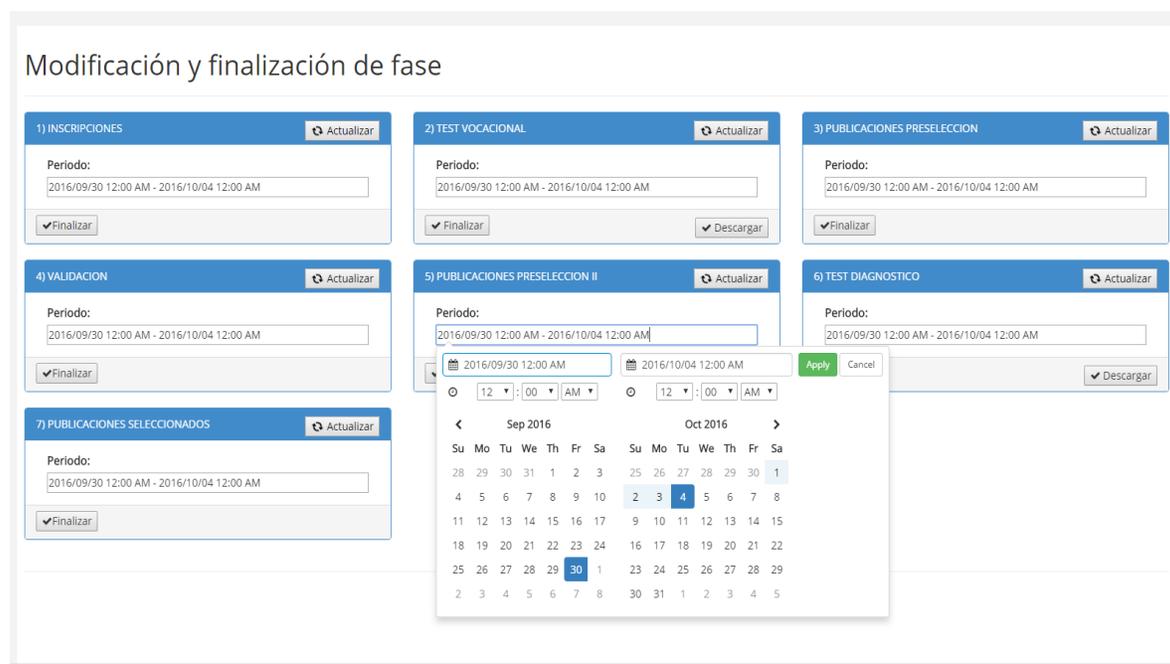


Ilustración 40: Modificación y finalización de las fases del sistema.

Funcionalidad de Materias por Carrera

Por medio de esta funcionalidad, el administrador puede indicar por cada una de las materias que se cursan en bachillerato, cuáles son las que competen en cada carrera que oferta la universidad para realizar el cálculo del promedio específico de cada opción seleccionada. Para ello, el sistema presenta la interfaz que se muestra en la Ilustración 41.

Agregar materia a carrera

Carreras

- Química
- Física
- Matemática
- Biología
- Ingeniería Geofísica
- Ingeniería Metalúrgica
- Ingeniería de Minas
- Ingeniería Geológica
- Ingeniería Geodésica
- Ingeniería Hidrometeorológica

Materias

- MATEMÁTICA
- CIENCIAS BIOLÓGICAS
- QUÍMICA
- FÍSICA
- CASTELLANO Y LITERATURA
- EDUCACIÓN ARTÍSTICA
- HISTORIA DE VENEZUELA
- HISTORIA UNIVERSAL
- HISTORIA CONTEMPORÁNEA DE VENEZUELA
- FILOSOFÍA

Condiciones Existentes

Mostrar registros Buscar:

Carrera	Materia
Administración	MATEMÁTICA
Administración	CASTELLANO Y LITERATURA
Administración	HISTORIA DE VENEZUELA
Administración	HISTORIA UNIVERSAL
Administración	HISTORIA DE VENEZUELA (CÁTEDRA BOLIVARIANA)
Administración	HISTORIA CONTEMPORÁNEA DE VENEZUELA
Antropología	MATEMÁTICA
Antropología	CASTELLANO Y LITERATURA
Antropología	HISTORIA DE VENEZUELA
Antropología	HISTORIA UNIVERSAL

Mostrando registros del 1 al 10 de un total de 242 registros

Anterior 1 2 3 4 5 ... 25 Siguiente

Ilustración 41: Selección de materias de bachillerato por carrera ofertada por la universidad.

Funcionalidad de Cupos

Con esta funcionalidad el sistema le permite al administrador indicar la cantidad de cupos disponibles por carrera, con el fin de que el sistema conozca cuál será la cantidad de aspirantes que puede seleccionar al finalizar cada fase de prueba realizada.

En la Ilustración 42, se puede visualizar la interfaz que le presenta el sistema al administrador.

Cupos de Carreras

Cupos por carrera

Mostrar registros Buscar:

Carrera	Cupos	
Administración	# <input type="text" value="0"/>	<input type="button" value="Actualizar"/>
Antropología	# <input type="text" value="0"/>	<input type="button" value="Actualizar"/>
Arquitectura	# <input type="text" value="0"/>	<input type="button" value="Actualizar"/>
Arquitectura (EUS) (Barquisimeto)	# <input type="text" value="0"/>	<input type="button" value="Actualizar"/>
Artes	# <input type="text" value="0"/>	<input type="button" value="Actualizar"/>
Bibliotecología y Archivología	# <input type="text" value="0"/>	<input type="button" value="Actualizar"/>
Bioanálisis	# <input type="text" value="0"/>	<input type="button" value="Actualizar"/>
Biología	# <input type="text" value="0"/>	<input type="button" value="Actualizar"/>
Ciencias Actuariales	# <input type="text" value="0"/>	<input type="button" value="Actualizar"/>
Ciencias Estadísticas	# <input type="text" value="0"/>	<input type="button" value="Actualizar"/>

Mostrando registros del 1 al 10 de un total de 76 registros

...

Ilustración 42: Carga de cupos por carrera.

Funcionalidad de Bitácora

Esta funcionalidad permite que el sistema le muestre al administrador, las actividades que realizaron los operadores y supervisores en el sistema, con la finalidad de reflejar si esos actores están cumpliendo a cabalidad sus responsabilidades.

En la Ilustración 43 se puede observar lo que refleja el sistema.

Bitacora de Validaciones

Condiciones Existentes

Mostrar registros Buscar:

Administrador	Aspirante	Facultad	Actividad	Fecha
Ningún dato disponible en esta tabla				

Mostrando registros del 0 al 0 de un total de 0 registros

Ilustración 43: Bitácora de validaciones.

No obstante, el sistema también proporciona otras funcionalidades, tales como:

Funcionalidad de Cargar Pruebas

Esta le permite al administrador cargar nuevos modelos de pruebas especificando su nombre, tipo (vocacional, diagnóstica verbal ó diagnóstica de lógica matemática) y subir el archivo con el formato de dicha prueba.

Por otro lado, el administrador también puede eliminar uno o varios de los modelos previamente cargados, si desea que el sistema no lo(s) siga manteniendo almacenado(s).

Funcionalidad de Horarios de Pruebas

Para esta funcionalidad el administrador indica a través del sistema, el horario disponible para que los aspirantes puedan presentar. Para ello, selecciona un modelo de prueba diagnóstica verbal y otro modelo de prueba diagnóstica lógica matemática para asociarlos a un horario y finalmente, agregar esa información al sistema.

Funcionalidad de Roles

Esta funcionalidad le permite al administrador cargar y eliminar aquellos usuarios (personal administrativo de la universidad autorizado) que pueden usar el sistema para la fase de validación de recaudos, ejerciendo los roles de supervisores u operadores respectivamente.

Funcionalidad de Reportes

Mediante esta funcionalidad, el administrador puede visualizar un conjunto de reportes útiles que reflejan la información procesada de los datos que dispone el sistema en un periodo determinado, permitiendo a su vez, poder descargar dichos reportes en un archivo con formato .xlsx para su impresión o uso. Estos reportes están predefinidos en el sistema y su información varía de acuerdo al periodo actual.

Algunos de los reportes que se pueden visualizar en esta funcionalidad son: listas enumerada con los datos de los de aspirantes por fases, registrados por carrera, facultad y estado, listados de preselección y admisión final igualmente por carrera, facultad y estado. A través de estos, el sistema le proporciona al administrador un acceso rápido y directo a la información almacenada, sin necesidad de realizar consultas directamente sobre la base de datos.

CAPÍTULO V

RESULTADOS

En este capítulo se exponen los resultados obtenidos durante el proceso de ejecución de la aplicación *web* propuesta en el presente Trabajo Especial de Grado, de manera que se pueda apreciar la escala de la data y población manejada durante el mismo, tomando como herramienta de visualización, gráficos de barra y de tarta, mediante los cuales se presenta información útil y oportuna para la universidad ya que con ella se puede mejorar y/u optimizar el proceso de ingreso.

Por otra parte, desde un punto de vista más amplio, se puede ver que, tras la ejecución de todas las fases pertenecientes al desarrollo de este sistema, el número total de aspirantes participantes fue de 9.880, de los cuales, sólo 2.624 de ellos, fueron admitidos en una de las carreras por las que optó ingresar (Ilustración 44).



Ilustración 44: Gráfico de aspirantes inscritos.

Los aspirantes inscritos en el proceso, fueron en su mayoría pertenecientes a la región capital del país, no obstante, aunque en otros estados no se destaca una población de aspirantes tan significativa (Ilustración 45), de igual forma, se puede visualizar el mismo comportamiento para el caso de los aspirantes admitidos (Ilustración 46).

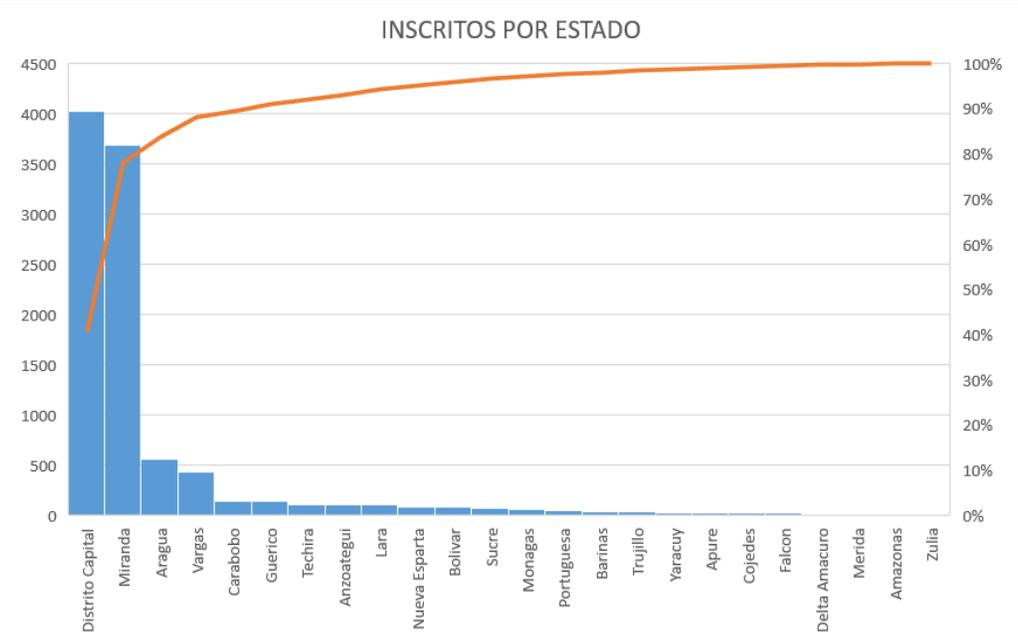


Ilustración 45: Gráfico de aspirantes inscritos por estados.

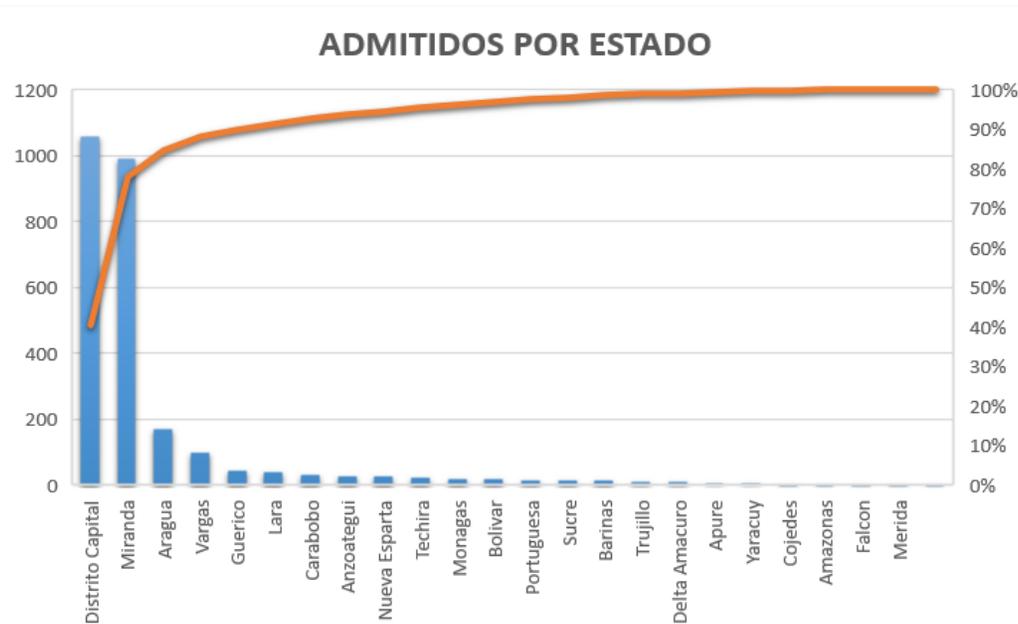


Ilustración 46: Gráfico de aspirantes admitidos por estados.

Sin embargo, tomando los valores expuesto anteriormente, se realiza una comparación en la cual se hace notar el nivel de selección resultante, visualizando así, en la Ilustración 47, una reducción del casi 35% de los aspirantes de todos los estados, con respecto a la población inicial, es decir, al total de aspirantes que se registraron en la aplicación.

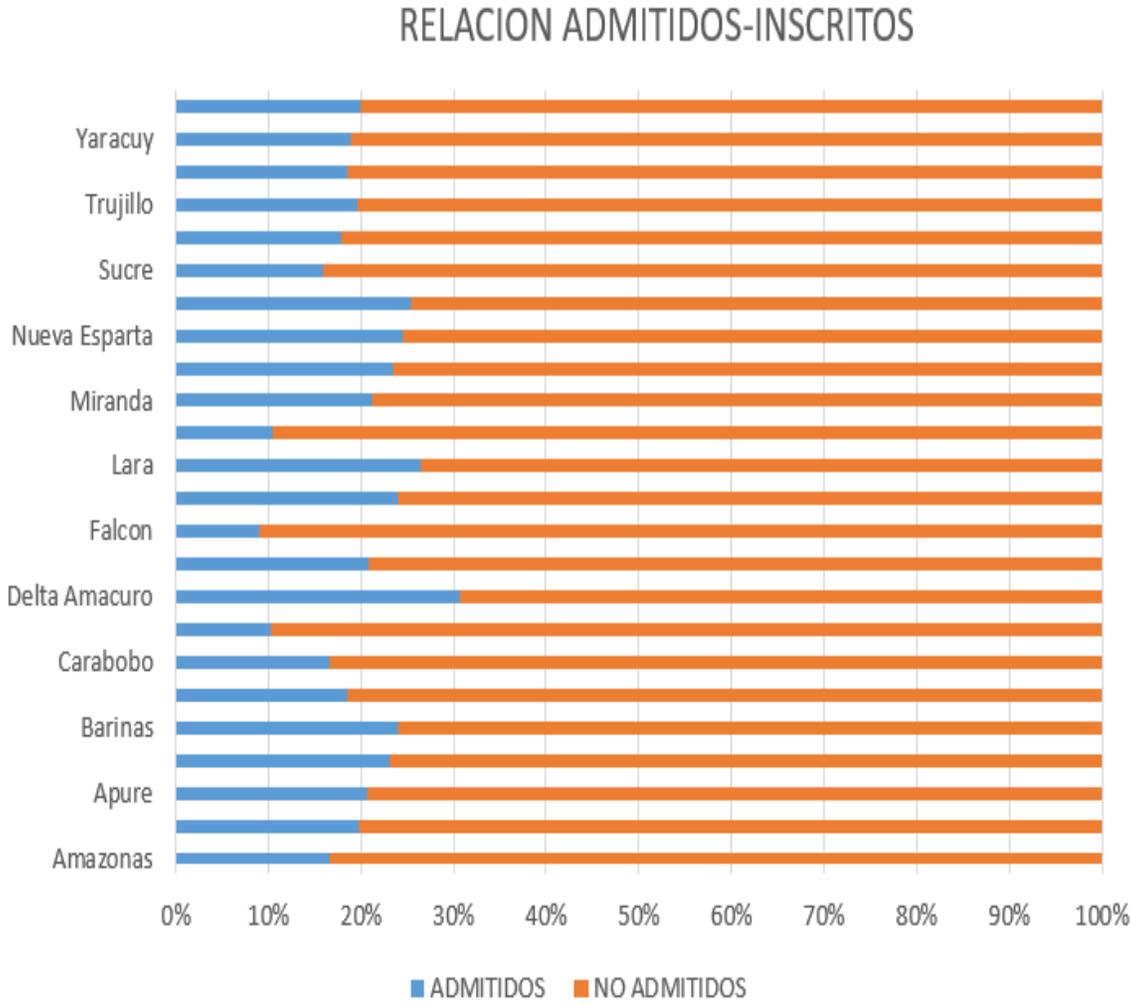


Ilustración 47: Gráfico de relación admitidos e inscritos.

Por último, se puede observar que, en la división del proceso por fases de selección, se fue reduciendo el número de aspirantes en dichas fases, debido a los procesos de evaluación y selección que se establecieron a fin de garantizar que la población resultante a cada fase sea la más capacitada para ingresar a la universidad (Ilustración 48).



Ilustración 48: Gráfico de aspirantes por fase del sistema.

Por otro lado, también el sistema proporciona información sobre las carreras que ofrece la universidad por facultad, tal como se refleja en la Ilustración 49, ya que a través de esta información se puede conocer la cantidad de campos de conocimientos que brinda esta casa de estudio por cada una de las facultades que la componen y con ello tener una visión de cual facultad tendrá una mayor demanda.

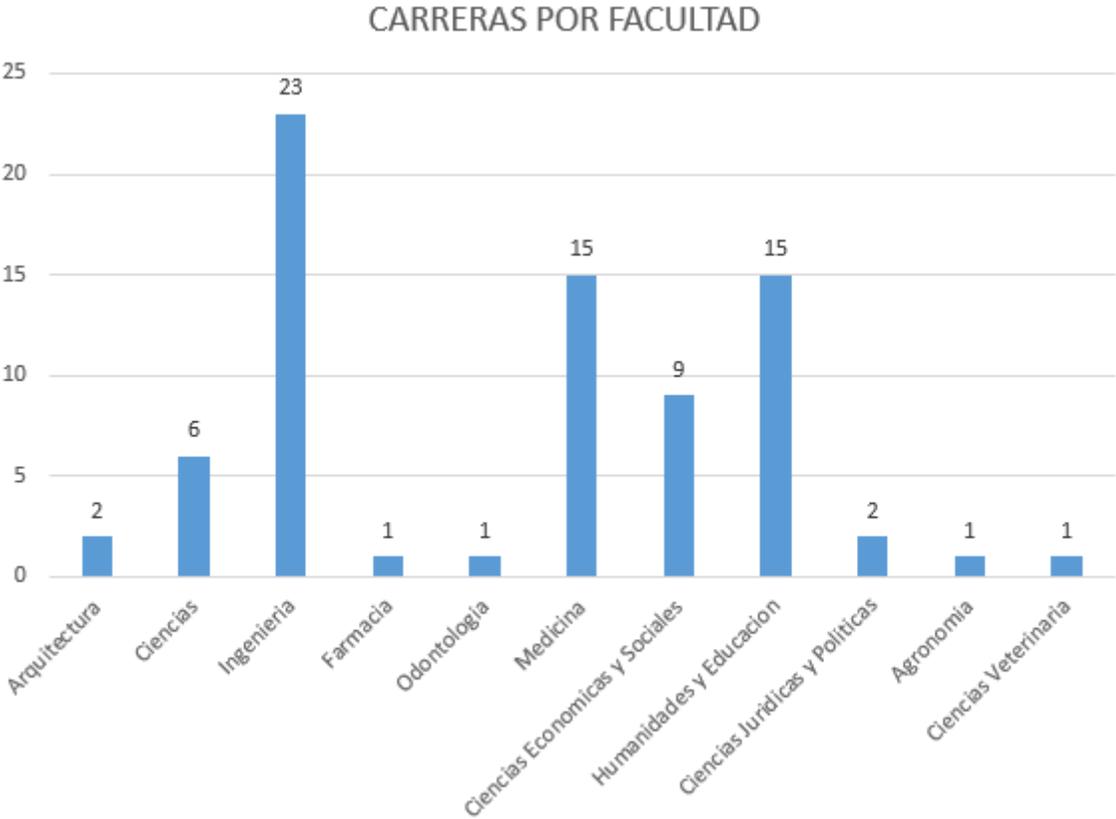


Ilustración 49: Gráfico de carreras por facultad.

Al mismo tiempo, se puede visualizar la cantidad de cupos permitidos por facultades, ya que esto señala las capacidades que tiene cada facultad para recibir nuevos estudiantes, destacando que, este numero depende mucho de la capacidad física que dispone dicha facultad para recibir nuevos ingresos, ya que cada una de las carreras poseen la modalidad de educación presencial (Ilustración 50).

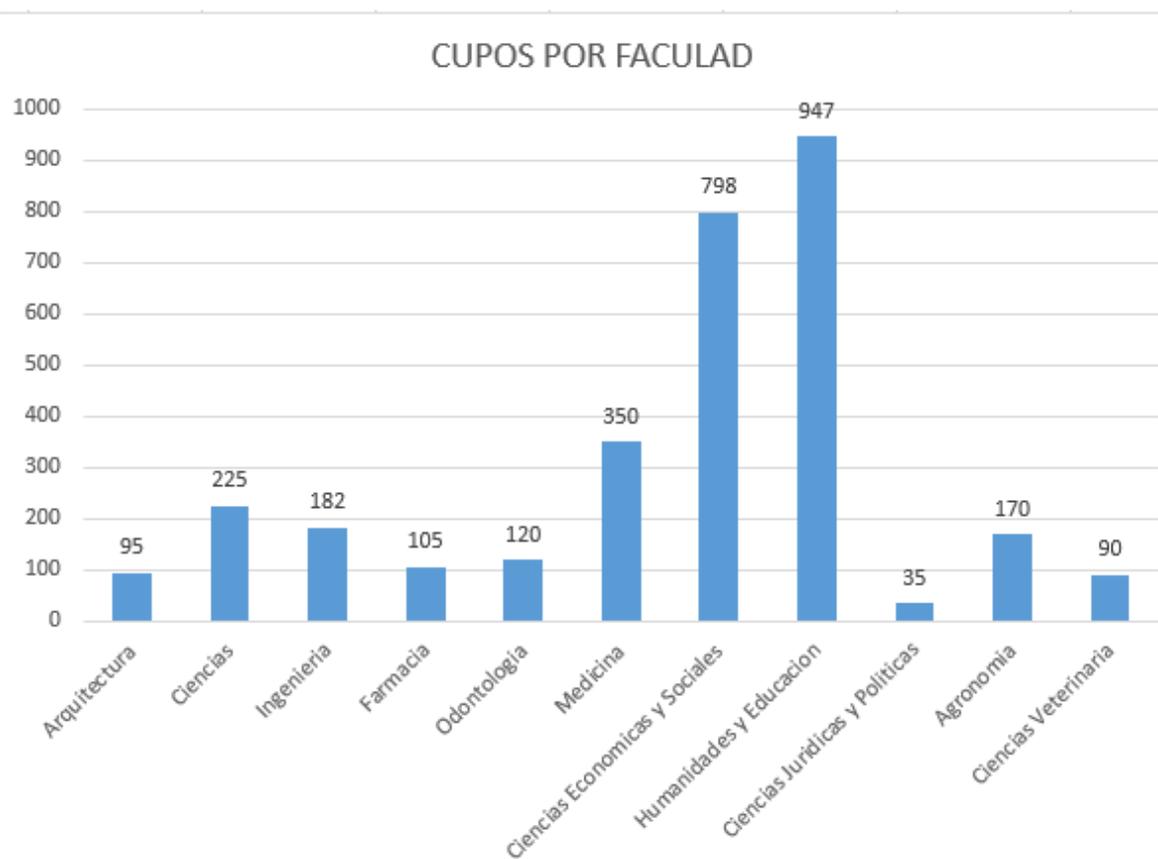


Ilustración 50: Gráfico de cupos por facultad.

CONCLUSIONES

Se cumplió satisfactoriamente el objetivo del presente Trabajo Especial de Grado, el cual consistió en el desarrollo de una aplicación *web* para la UCV, llamada “Sistema de Ingreso por Mérito Académico y Diagnóstico Integral”, que permite la gestión, organización, evaluación y selección de los aspirantes a ingresar en ella por cada Facultad, a fin de automatizar estos procesos cumpliendo con cada una de las etapas que se llevan a cabo de forma manual, y así, mejorar las actividades de la universidad en cuanto al proceso de ingreso e incentivar a los miembros de la misma. Haciendo todo esto posible, gracias a la investigación previamente realizada sobre el tópico de investigación.

Se logró la obtención de un producto que cumple con los requerimientos generales planteados al principio del desarrollo del mismo, la definición e implementación de una arquitectura tecnológica y la integración de las fases que componen el sistema de forma satisfactoria.

El uso del método incremental y creciente permitió mediante un conjunto de iteraciones un desarrollo de forma ágil, incremental, iterativo y estructurado a lo largo del proyecto, lo cual facilitó la construcción del sistema.

Se diseñó e implementó en el Sistema Manejador de Base de Datos MySQL, partiendo de un análisis previo, un modelo relacional sólido que soporta todo el proceso de ingreso y permite proporcionar de forma rápida y efectiva un acceso conveniente a la información que se requiere para la obtención de resultados precisos.

Adicionalmente, se utilizó el *framework* de desarrollo de aplicaciones *web* Laravel, en conjunto con otras tecnologías (HTML5, CSS, JS, JSON, AJAX, entre otros) y herramientas (Git, TortoiseGit) para este tipo de aplicaciones, con el objetivo de implantar las funcionalidades definidas para el sistema, y garantizar un desarrollo y funcionamiento óptimo del mismo.

Durante el desarrollo de esta aplicación, se elaboraron y ejecutaron pruebas, tomando como criterio las observaciones dadas por el equipo de trabajo, con la finalidad de garantizar que el sistema cumpla con las expectativas deseadas.

Finalmente, con la implantación del nuevo sistema la Universidad Central de Venezuela, obtendrá grandes beneficios, ya que la automatización del proceso de ingreso de estudiantes a la universidad permite que dicho proceso capture la información de los aspirantes y analice los resultados de una manera más rápida y oportuna; logrando que el almacenamiento y procesamiento de los datos sea íntegro, consistente y apoye el proceso de toma de decisiones. Con la sistematización de esta experiencia la universidad cuenta con una memoria que permite la marca y crecimiento de este proceso facilitando la toma de decisiones para la continuidad del sistema de ingreso por mérito académico y diagnóstico integral.

Recomendaciones

Con la finalidad de extender las funcionalidades creadas en esta aplicación *web* y mejorar la usabilidad y experiencia del mismo, se proponen las siguientes recomendaciones para trabajos futuros:

- Incorporar un módulo de gestión de pruebas que permita la elaboración de modelos de pruebas de manera dinámica, facilitando así, que el contenido de dichas pruebas sea variado y escalable en el tiempo, sin la necesidad de cargar un archivo al sistema, ya que este puede resultar tedioso porque debe cumplir con un formato determinado.
- Desarrollar un módulo de gestión de análisis en el cual se realice la implementación de una Solución de Inteligencia de Negocio, que permita obtener los datos almacenados del sistema y transformarlos en información confiable y oportuna por medio de reportes, indicadores y cuadros de mando, que le sirva de utilidad a la universidad para la toma de decisiones significativas en pro del proceso de ingreso.
- Realizar un estudio a nivel de UX (Experiencia de Usuario) sobre la población que actualmente se encuentra postulando a la universidad, debido a que las tecnologías, simbologías y patrones de comportamiento a los que están acostumbrados, generan pequeños problemas de adaptación y orientación dentro del sistema.

De igual forma, con base en la logística implementada por la universidad para la ejecución del proceso de ingreso, se recomienda:

- Plantear medidas de contingencia en caso de presentarse algún inconveniente con el proceso, como, por ejemplo: disponer de salas capacitadas por cada facultad para que los aspirantes puedan acudir a realizar el proceso de manera asistida.
- Utilizar las redes sociales en una escala mayor con la finalidad de que los estudiantes puedan informarse y notificarse acerca del proceso. Asimismo, el uso de dichas redes no sólo permitirá compartir información, sino que a futuro podría contemplarse como una fuente de información para la universidad permitiéndole medir su alcance social, debido a que estas redes disponen de herramientas que permiten obtener datos cuantificables sobre su alcance e interacción social.
- Realizar una capacitación, mediante la utilización de pequeñas campañas publicitarias que instruyan al aspirante acerca del funcionamiento del sistema, permitiendo que éste no se sienta abrumado o desorientado durante la utilización del mismo.

BIBLIOGRAFÍA

- Alarcón, V. F. (s.f.). Desarrollo de sistemas de información: una metodología basada en el modelado.
- Alcaraz, F., Espín, A., Martínez, A., & Alarcón, M. (1 de Octubre de 2006). *Revista Clínica Médica Familiar*. Recuperado el 15 de Junio de 2015, de Diseño de Cuestionarios para la recogida de información: metodología y limitaciones:
<http://www.revclinmedfam.com/PDFs/06409663226af2f3114485aa4e0a23b4.pdf>
- Almaraz Hernández, J. M., Campos Cantero, P., & Castelo Delgado, T. (2011). *Docplayer*. Obtenido de Desarrollo de una aplicación Web para la gestión de Entornos Virtuales:
<http://docplayer.es/3232988-Desarrollo-de-una-aplicacion-web-para-la-gestion-de-entornos-virtuales.html>
- Alter, S. (1996). *Information Systems: A Management Perspective. 2nd Edition.* . California: The Benjamin / Cummings Publishing Company.
- Bascón Pantoja, E. (2004). *Universidad Católica Boliviana "San Pablo" Unidad Académica Cochabamba*. Obtenido de El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing:
<http://www.ucbca.edu.bo/Publicaciones/revistas/actanova/documentos/v2n4/v2.n4.bascon.pdf>
- Beltrán Jaramillo, J. M. (2006). *Indicadores de Gestión: Herramientas para lograr la competitividad. 2da Edición.* 3R Editores.
- Blanco, P. (21 de marzo de 2013). *Desarrollando Webs Dinámicas*. Obtenido de ¿Qué es Laravel?:
<http://desarrollandowebsdinamicas.blogspot.com/2013/03/que-es-laravel.html>
- Bonnefoy, J. C. (Marzo de 2006). *Indicadores de Desempeño*. Obtenido de CEPAL (Comisión Económica para América Latina y el Caribe):
<http://www.cepal.org/ilpes/noticias/paginas/2/23992/Indicadores%20de%20Desempe%C3%B1o.pdf>
- Cano. (2007). *Business Intelligence: Competir con Información*.
- Caracheo, E. (28 de Enero de 2015). *Metodología del desarrollo para sistemas de información basados en Web*. Obtenido de Repositorio Institucional de la Universidad Autónoma de Queretaro: <http://ri.uaq.mx/bitstream/123456789/2394/1/RI001928.pdf>
- Christiansen , R., & Gaete, E. (2014). *Sistemas de tipo OLTP*.
- Colombia, S. d. (2014). *SIRAC*. Recuperado el 11 de Junio de 2015, de Indicadores:
<http://www.sirac.info/hospitales/html/indicadores.asp>
- Colomina, E. (28 de Septiembre de 1998). *Universidad de Alicante*. Recuperado el 22 de Mayo de 2015, de Adopción de sistemas de información en las PYME: teoría y evidencia empírica:
<http://rua.ua.es/dspace/bitstream/10045/3393/1/Colomina%20Climent%2c%20Evaristo.pdf>

- Comunidad Rails. (2007). *Introducción a Ruby on Rails*. Recuperado el 17 de Abril de 2015, de https://librosweb.es/libro/introduccion_rails/
- Damián. (2010). *¿Qué es CSS3?* Recuperado el 24 de Marzo de 2015, de <http://html5.dwebapps.com/que-es-css3/>
- Definición ABC. (25 de septiembre de 2015). *Definición ABC*. Obtenido de Definición de Universidad; Concepto en Definición ABC: <http://www.definicionabc.com/general/universidad.php>
- EduRed. (8 de Abril de 2012). Obtenido de Metodología de desarrollo iterativo y creciente: https://www.ecured.cu/Metodolog%C3%ADa_de_desarrollo_iterativo_y_creciente
- EduRed. (2013). *Modelo Espiral - EduRed*. Obtenido de Modelo Espiral: https://www.ecured.cu/Modelo_Espiral
- Estrada, L. (8 de Junio de 2012). *El Desempeño Docente*. Obtenido de https://upload.wikimedia.org/wikipedia/commons/d/dd/IMPORTANCIA_DEL_DESEMPE%C3%91O_DOCENTE.pdf
- Figus LTDA. (2014). *Figus*. Obtenido de Git - Manual de usuario versión 1.: http://blog.figus.com/wp-content/uploads/2014/05/git_manual.pdf
- Flores, M. (2014). *HTML5Fácil*. Obtenido de Introducción a AngularJS: <http://html5facil.com/tutoriales/introduccion-angularjs/>
- Grant, N. (2003). El liderazgo de los estudiantes hoy. *New Horizon For Learning*.
- Gravitar. (12 de Octubre de 2014). *Gravitar - Información sin límites*. Recuperado el 22 de julio de 2015, de Pentaho : <http://gravitar.biz/pentaho/>
- Hernández, G. (11 de Septiembre de 2011). *Slideshare*. Recuperado el 16 de Junio de 2015, de Escala de Likert: http://es.slideshare.net/gabriela_hernandez/escala-de-likert-9182198
- Hernández, M., García , S., Abejón, N., & Zazo, M. (19 de Noviembre de 2010). *Universidad Autónoma de Madrid*. Recuperado el 11 de Junio de 2015, de Estudio de Encuestas: https://www.uam.es/personal_pdi/stmaria/jmurillo/InvestigacionEE/Presentaciones/Curso_10/ENCUESTA_Trabajo.pdf
- Imhoff, C., & Gallemmo, N. (2003). *Mastering DataWarehouse Desing: Relational and Dimensional Techniques*. New York: John Wiley & Sons.
- Inmon, W. H. (1996). *Building the Data Warehouse*. New York: John Wiley & Sons.
- José, N. (2010). *Usabilidad en metodologías ágiles (tesis de master)*. Obtenido de Facultad de Informática, Universidad Politécnica de Madrid, España: https://www.fi.upm.es/catedra-ibmrational/sites/www.fi.upm.es.catedra-ibmrational/files/Tesis_MarcoAgilTrabajo.pdf
- jQuery. (2015). *jQuery write less, do more*. Recuperado el 24 de Abril de 2015, de <http://jquery.com/>
- JSON. (10 de julio de 2016). *JSON*. Obtenido de Introducción a JSON: <http://www.json.org/json-es.html>

- Kimball. (2002). *The Data Warehouse Toolkit*. Wiley Computer Publishing.
- Kimball, & Caserta. (2008). *The Data Warehouse ETL Toolkit* (2da Ed.).
- Kimball, R. (1998). *The Data Warehouse Lifecycle Toolkit*. New York: John Wiley & Sons.
- Kimball, R. (2008). *The Data Warehouse Lifecycle Toolkit. 2nd Edition*. New York: John Wiley & Sons.
- Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit*. New York: John Wiley & Sons.
- Laudon, K., & Laudon, J. (2004). *Sistemas de Información Gerencial (Versión en Español)*. México: Prentice Hall.
- Luján Mora, S. (2001). *Programación en Internet: Cliente Web*. Club Universitario.
- Márquez Avendaño, B. M., & Zulaica Rugarcía, J. M. (12 de enero de 2004). *Colección de Tesis Digitales - Universidad de las Américas Puebla*. Obtenido de Implementación de un reconocedor de voz gratuito a el sistema de ayuda a invidentes Dos-Vox:
http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/portada.html
- Méndez del Río, L. (s.f.). Más allá del Business Intelligence: 16 experiencias de éxito.
- Méndez K., E. E. (2010). *A Quantitative Framework for the Evaluation of Agile Methodologies*. Obtenido de Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, Bahía Blanca, Argentina:
<http://journal.info.unlp.edu.ar/journal/journal28/papers/JCST-Jun10-4.pdf>
- Metodología Kimball*. (30 de enero de 2014). Obtenido de Inteligencia de Negocio:
<http://inteligenciadenegociosval.blogspot.com/2014/01/metodologia-de-kimball.html>
- México, A. N. (2004). *Asociación Nacional de Universidades e Instituciones de Educación Superior de México*. México: Colección Biblioteca de la Educación Superior. Serie Investigaciones.
- México, S. d. (20 de Septiembre de 2011). *SHCP*. Recuperado el 11 de Junio de 2015, de Indicadores de Desempeño:
http://www.shcp.gob.mx/EGRESOS/sitio_pbr/progra_presupuestacion/Paginas/indicadores_des.aspx
- MySQL. (2011). *MySQL*. Obtenido de MySQL 5.6 Reference Manual.:
<http://downloads.mysql.com/docs/refman-5.6-en.pdf>
- Oracle. (s.f.). *Creating a Repository Using the Oracle Business Intelligence Administration Tool*. Obtenido de Oracle:
http://www.oracle.com/webfolder/technetwork/tutorials/obe/fmw/bi/bi1113/biadmin11g_01/biadmin11g.htm#t7
- Oracle. (s.f.). *Oracle Database 11g Standard Edition One*. Obtenido de Oracle:
<http://www.oracle.com/us/products/database/standard-edition-one/overview/index.html>
- Pacheco, J., Castañeda, W., & Caicedo, H. (2002). *Indicadores integrales de gestión*. Colombia: McGraw-Hill.

- Padilla Sierra, G., & Ramos Tejeda, M. (s.f.). *Psicología del aprendizaje*.
- Palacio, J. (16 de Octubre de 2006). *El Modelo Scrum*.
- Pentaho. (2012). *PENTAHO*. Recuperado el 12 de julio de 2015, de <http://www.pentaho.com/>
- Peña. (2006). *Tecnologías de la Información*. Mexico.
- Peñafiel, M. (15 de abril de 2015). *Scribd*. Obtenido de Aplicaciones Web: <https://es.scribd.com/doc/136052164/APLICACIONES-WEB-pdf>
- Pérez Lugo, J. (2002). *Monografías*. Recuperado el 11 de Junio de 2015, de Importancia del Liderazgo Directivo en el Desempeño Docente en la I y II Etapa de Educación Básica: <http://www.monografias.com/trabajos13/lider/lider.shtml>
- PHP Org. . (01 de octubre de 2016). *PHP*. Obtenido de Manual de PHP : Manual de PHP
- Pressman, R. (2002). Ingeniería del Software. En R. Pressman, *Ingeniería del Software* (Quinta ed.). MacGraw Hill.
- Real Academia Española. (2016). *Diccionario de la lengua española - Edición del Tricentenario*. Obtenido de DLE - Diccionario de la lengua española - Edición del Tricentenario: <http://dle.rae.es/?id=Lb03XBB>
- Real Academia Española. (2016). *Real Academia Española*. Obtenido de DLE: proceso - Diccionario de la lengua española - Edición del Tricentenario: <http://dle.rae.es/?w=diccionario>
- Red Iberoamericana para la Acreditación de la Calidad de la Educación Superior. (2004). *Glosario internacional RIACES de evaluación de la calidad y acreditación: documento*. Obtenido de <http://www.saidem.org.ar/docs/Glosario/RIACES.%20T%E9rminos%20para%20la%20Evaluaci%F3n%20de%20la%20calidad%20y%20acreditaci%F3n.pdf>
- Rivadera, G. (2010). *Ucasal*. Recuperado el 02 de Junio de 2015, de La metodología de Kimball para el diseño de almacenes de datos: <http://www.ucasal.edu.ar/htm/ingenieria/cuadernos/archivos/5-p56-rivadera-formateado.pdf>
- Rivadera, G. R. (s.f.). *La metodología de Kimball para el diseño de almacenes de* . Obtenido de <http://www.ucasal.edu.ar/htm/ingenieria/cuadernos/archivos/5-p56-rivadera-formateado.pdf>
- Robbins, S. (2004). *Comportamiento Organizacional*. México: Prentice Hall.
- Robbins, S., & DeCenzo, D. (2008). *Supervisión. Quinta Edición*. México: Prentice Hall.
- Robles, J. (8 de Julio de 2010). *Generador de Encuestas*. Obtenido de <https://prezi.com/ayachz5agskd/generador-de-encuestas/>
- Ruby Org. (2015). *Sitio en español de la organización Ruby*. Recuperado el 22 de Marzo de 2015, de <https://www.ruby-lang.org/es/>
- Sánchez, J. (2003). *Manual de referencia de JavaScript*. Obtenido de <http://www.jorgesanchez.net/web/javascript.pdf>
- Sánchez, J. (2012). *Servidores de Aplicaciones Web*. Obtenido de Implantación de Aplicaciones Web en Sistemas Informaticos de Red: <http://www.jorgesanchez.net/web/iaw/iaw1.pdf>

- Scrum. (27 de Julio de 2015). *Scrum*. Obtenido de Acerca de Scrum: <https://www.scrum.org/about>
- Siguan, M. (1973). La selección para el ingreso en la Universidad. *CONVIVIUM*.
- Sinnexus. (2012). *Bases de datos OLTP y OLAP*. Recuperado el 25 de Mayo de 2015, de http://www.sinnexus.com/business_intelligence/olap_vs_oltp.aspx
- Solórzano, B. (2004). Planeación y Desarrollo de Web Site. *Revista del Centro de Investigación*. Vol. 6. Nro. 21, 78.
- Tahuiton Mora, J. (agosto de 2011). *Centro de Investigacion y de Estudios Avanzados del Instituto Politecnico Nacional*. Obtenido de Arquitectura de software para aplicaciones Web: <https://www.cs.cinvestav.mx/TesisGraduados/2011/tesisJuanTahuiton.pdf>
- TortoiseGit. (11 de julio de 2015). *TortoiseGit*. Obtenido de TortoiseGit Manual: <https://tortoisegit.org/docs/tortoisegit/tgit-introduction.html#tgit-intro-about>
- UCV. (2009). *Universidad Central de Venezuela*. Obtenido de <http://www.ucv.ve/>
- UNA. (1990). *Técnicas de Documentación e Investigación (III)*. Caracas.
- Universidad Central de Venezuela, E. (10 de Diciembre de 2011). *Centro de Estudiantes de Computación*. Obtenido de Normativa de Evaluaciones de la Escuela de Computación: <https://cecucv.wordpress.com/compilacion-legislativa/normativa-de-evaluaciones-de-la-escuela-de-computacion/>
- Universidad de Champagnat. (16 de Julio de 2002). *Gestiopolis*. Recuperado el 12 de Junio de 2015, de Encuesta, Cuestionario y Tipos de Pregunta: <http://www.gestiopolis.com/encuesta-cuestionario-y-tipos-de-preguntas/>
- Villarreal Fuentes, C. A. (2013). *Northware*. Obtenido de ¿Qué es jQuery?: http://www.northware.mx/wp-content/uploads/2013/11/Noviembre_Que_es_jQuery__Northware.pdf
- W3C. (22 de marzo de 2012). *W3C*. Obtenido de Guía Breve de CSS: <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>
- W3Schools. (2012). *W3Schools*. Obtenido de HTML5 Tag Reference: http://www.w3schools.com/html5/html5_reference.asp
- W3Schools. (2015). Obtenido de AJAX Introduction: http://www.w3schools.com/Ajax/ajax_intro.asp
- Walker, G. (14 de junio de 2011). *IBM*. Obtenido de Fundamentos HTML5, Parte 1: <http://www.ibm.com/developerworks/ssa/web/library/wa-html5fundamentals/>