



Universidad Central de Venezuela

Facultad de Ciencias

Escuela de Computación

Centro de Ingeniería de Software y Sistemas



**Módulo de gestión de procesos para la
evaluación instrumental de idiomas que
realiza la Escuela de Idiomas Modernos**

Trabajo Especial de Grado presentado ante la ilustre

Universidad Central de Venezuela por:

Br. Daniela Colmenares

Para optar por el título de
Licenciado en Computación

Tutora:

Profa. Jossie Zambrano

Caracas, junio 2013

ACTA

Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela, para examinar el Trabajo Especial de Grado titulado: Módulo de gestión de procesos para la evaluación instrumental de idiomas que realiza la Escuela de Idiomas Modernos de la UCV, presentado por el bachiller Daniela I. Colmenares M., C.I.: 19.023.671, a los fines de optar por el título de Licenciado en Computación, dejan constancia de lo siguiente: Leído el trabajo por cada uno de los miembros del jurado, se fijó el día Lunes 29 de Julio de 2013, a las 08:30 am, para que su autor lo defendiera en forma pública en la Sala de Reuniones de Isys en la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondió a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo con la nota de _____ puntos. En fe de lo cual se levanta la presente Acta, en Caracas a los veintinueve (29) días del mes de Julio del año dos mil trece (2013).

Profa. Jossie Zambrano (Tutor)

Prof. Sergio Rivas (Jurado)

Profa. Lucía Ojeda (Jurado)

DEDICATORIA

A ustedes abuelos.

Que han sido mi inspiración, ejemplo a seguir día a día y mi mayor apoyo a lo largo de los años. Los amo con todo mi corazón.

A ustedes Mami, Sisti y Papá.

Que me dieron la motivación necesaria para superarme cada día, su amor incondicional. Que creyeron en mí sobre todas las cosas y me dieron la fuerza para superar cualquier dificultad que pude tener. Los amo muchísimo.

A mi tío Frank y tía Gladys.

Que gracias a su inspiración ayudaron a crear generaciones de Licenciados en Computación. Fueron los pioneros, Dios los bendiga hoy y siempre.

A ustedes Familia.

Papus, Iris, José Antonio, Lisbeth, Alexandra, Miguel Ángel, Jesús, Ingrid, Lilo y Lily que siempre han estado a mi lado cuando lo necesito. Que adoro y quiero infinitamente.

A ti Johnny.

Que en poco tiempo te has convertido en mi gran apoyo y sonrisa de todos los días. Te adoro.

A ustedes Jane y Karla.

Que han sido las mejores amigas que alguien puede tener. Y he contado con ustedes en cualquier momento desde que las conocí.

A ti Jossie.

Que eres una madre más, mi tutora, mi profesora, mi guía e inspiración por tu fuerza en los momentos más difíciles. Me impulsaste a conseguir esta gran meta.

A ti Sergio.

Que me ayudaste a desarrollar y formar mi conocimiento profesional, y al mismo tiempo a ser un amigo incondicional siempre que lo necesité.

AGRADECIMIENTOS

A Dios.

Por haberme guiado por el buen camino y darme las fuerzas necesarias para seguir luchando día tras día.

A ustedes abuelos.

Gracias por cuidarme, enseñarme, quererme en los momentos que lo necesité y no. Gracias por ser mi inspiración, ejemplo a seguir día a día y mi mayor apoyo a lo largo de los años.

A ustedes Mami, Sisti y Papá.

Gracias por darme el ánimo necesario para lograr todas mis metas. Gracias por siempre hacerme saber que solo yo coloco el límite para lograr lo que quiero. Gracias por darme el apoyo de lograr mi sueño.

A ustedes Familia.

Gracias a Papus, Iris, José Antonio, Lisbeth, Alexandra, Miguel Ángel, Lilo, Lily, Jesús e Ingrid por siempre estar a mi lado y ayudarme en todo lo que necesito en cualquier momento.

A ti Johnny.

Gracias por apoyarme, entretenerme, tus consejos y el cariño que me das. Gracias por ser como eres conmigo.

A ustedes Jane y Karla.

Gracias por ser mis amigas incondicionales SIEMPRE.

A ti Jossie.

Gracias por tu dedicación, tus buenos consejos, tu apoyo a lo largo de mi carrera profesional. Gracias por tratarme como una hija y amiga más. Gracias por ayudarme a lograr este sueño.

A ti Sergio.

Gracias por ayudarme, enseñarme todo lo necesario para lograr mi meta. Gracias por ser el mejor guía a lo largo de mi desarrollo.

A ustedes tíos y primos.

Gracias por ser impulsarme a conseguir mis metas.

A ustedes profesores.

Gracias a todos los que me educaron y contribuyeron a formar lo que soy el día de hoy.

A ustedes grandes amigos.

Gracias por ser quienes me acompañaron a lo largo de mi carrera universitaria, por estar presentes en todo momento, tanto personal como académico. Sin ustedes todo hubiera sido diferente, le dieron color a este camino tan difícil.

A ustedes compañeros de Infosoftware y Teradata.

Gracias por su colaboración, ayuda, consejos y sobre todo por hacerme sentir parte de su familia, muchísimas gracias.

A la UCV.

Gracias a mi casa de estudio, por prepararme, ofrecerme todos los conocimientos y formarme como profesional y persona.

RESUMEN

La Coordinación de Extensión de la Escuela de Idiomas Modernos de la Universidad Central de Venezuela, es la encargada de realizar los procesos que intervienen en el desarrollo de la evaluación instrumental, que consiste en una comprensión lectora de un idioma extranjero para los estudiantes de postgrado de dicha Universidad. El proceso a seguir para el desarrollo de la gestión de esta evaluación no se encuentra automatizado, motivo por el cual surge el planteamiento de realizar este Trabajo Especial de Grado, que tiene como objetivo general el desarrollar un módulo de gestión de procesos de inscripción, gestión administrativa y la generación de certificados y constancias para la evaluación instrumental, con el fin de reducir costos tanto en tiempo como en recursos humanos, para que puedan ser utilizados en labores de coordinación, planificación, entre otros. Para la implementación de este sistema se aplican los patrones de proceso y una instancia del método de desarrollo de software ágil llamado AgilUS, que está enfocado en el usuario final y en la incorporación de la usabilidad desde etapas tempranas del desarrollo de software.

Palabras Clave:

Evaluación, instrumental, proceso de inscripción, aplicación Web, método de desarrollo de software ágil, AgilUS, patrones de proceso.

Contactos

Daniela Colmenares

danicolmenares22@gmail.com

Jossie Zambrano

Escuela de Computación

jossie.zambrano@gmail.com

ÍNDICE GENERAL

Acta	2
Dedicatoria	3
Agradecimientos	4
Resumen.....	6
Índice General	7
Índice de Ilustraciones	9
Índice de Tablas.....	11
Introducción	12
Capítulo I Examen Instrumental.....	14
1.1 Situación problemática	14
1.1.1 Inscripción	15
1.1.2 Gestión administrativa para la evaluación instrumental	17
1.1.3 Generación de constancia o certificado	18
1.2 Planteamiento del problema	18
1.3 Objetivo general	19
1.4 Objetivos específicos.....	19
1.5 Justificación	19
1.6 Alcance	20
Capítulo II Tecnologías Web.....	22
2.1 Aplicaciones Web	22
2.1.1 Arquitectura Cliente-Servidor	23
2.2 Tecnologías del lado del Cliente.....	25
2.2.1 HyperText Markup Language, versión 5 (HTML 5).....	25
2.2.2 Cascading Style Sheets, versión 3 (CSS 3)	25
2.2.3 JavaScript.....	27
2.2.4 jQuery	27
2.2.5 jQuery User Interface (jQuery UI)	28
2.2.6 Asynchronous JavaScript And XML (AJAX)	30
2.3 Tecnologías del lado del Servidor.....	32
2.3.1 Plataforma Debian GNU/Linux.....	32
2.3.2 Apache.....	32
2.3.3 Unicorn	33
2.3.4 Patrón de Diseño MVC (Modelo Vista Controlador)	33
2.3.5 Ruby on Rails	35
2.3.6 MySQL	37
Capítulo III Ingeniería Web.....	39
3.1 Ingeniería Web	39
3.2 Método de Desarrollo de Software.....	40
3.2.1. AgilUS	41
3.3 Reusabilidad	46
3.3.1 Beneficios de la reusabilidad.....	46
3.4 Patrón de proceso	48
3.4.1 Metapatrón de procesos.....	48
3.4.2 Patrones de proceso de la evaluación instrumental	49

Capítulo IV Desarrollo de Software	56
4.1 Iteración 1 - Análisis Global	56
4.1.1 Tormenta de ideas	56
4.1.2 Requerimientos funcionales	57
4.1.3 Requerimientos no funcionales	59
4.1.4 Perfil de usuario	59
4.1.5 Modelo de casos de uso	60
4.1.6 Modelo de objetos del dominio	63
4.1.7 Patrones de interacción	64
4.2 Iteración 2 - Inscripción del curso y examen instrumental	70
4.2.1 Prototipo rápido	71
4.2.2 Prototipo ejecutable	71
4.2.3 Evaluación heurística.....	72
4.2.4 Entrega	74
4.3 Iteración 3 Gestión Administrativa	74
4.3.1 Prototipo rápido	75
4.3.2 Prototipo ejecutable	77
4.3.3 Evaluación heurística.....	87
4.3.4 Entrega	87
4.4 Iteración 4 Generación de reportes	87
4.4.1 Prototipo rápido	88
4.4.2 Prototipo ejecutable	88
4.4.3 Evaluación heurística.....	89
4.4.3 Entrega	89
4.5 Aceptación de la aplicación	90
Conclusiones y Recomendaciones	91
Referencias Bibliográficas	93

ÍNDICE DE ILUSTRACIONES

Ilustración N° 1.1 Inscripción para presentar el examen.....	16
Ilustración N° 1.2 Inscripción para el curso.....	17
Ilustración N° 2.1 Arquitectura Cliente - Servidor.....	24
Ilustración N° 2.2 Funcionamiento de una aplicación Web tradicional.....	31
Ilustración N° 2.3 Funcionamiento de una aplicación Web con AJAX.	31
Ilustración N° 2.4 Patrón de diseño MVC.....	34
Ilustración N° 3.1 Etapas del método AgilUS.	43
Ilustración N° 3.2 Catálogo de Patrones de Proceso.....	49
Ilustración N° 3.3 Catálogo de patrón de proceso para la inscripción del curso y evaluación instrumental.....	55
Ilustración N° 4.1 Nivel 0 del diagrama de Casos de Uso.....	60
Ilustración N° 4.2 Nivel 1 del diagrama de Casos de Uso.....	61
Ilustración N° 4.3 Nivel 2 del diagrama de Casos de uso.	63
Ilustración N° 4.4 Nivel 3 del diagrama de Casos de Uso.....	63
Ilustración N° 4.5 Modelo de objetos del dominio.	64
Ilustración N° 4.6 Taxonomía de los patrones de interacción para la evaluación instrumental.	70
Ilustración N° 4.7 Prototipo rápido para las inscripciones.....	71
Ilustración N° 4.8 Prototipo ejecutable de la inscripción del examen.	72
Ilustración N° 4.9 Prototipo ejecutable de la inscripción del curso instrumental.	72
Ilustración N° 4.10 Prototipo rápido para el almacenamiento de exámenes.....	75
Ilustración N° 4.11 Prototipo rápido para el manejo de fechas de presentación de los exámenes.	76
Ilustración N° 4.12 Prototipo rápido para la configuración general.	76
Ilustración N° 4.13 Gestionar fechas para exámenes.	77
Ilustración N° 4.14 Funcionalidad para modificar fecha de presentación.	78
Ilustración N° 4.15 Funcionalidad para calificar.....	78
Ilustración N° 4.16 Funcionalidad para enviar correos.	79
Ilustración N° 4.17 Repositorio de los exámenes.....	79
Ilustración N° 4.18 Gestión del curso instrumental.	80
Ilustración N° 4.19 Funcionalidad para agregar un nuevo estudiante.....	80
Ilustración N° 4.20 Funcionalidad para consultar curso activo.....	81
Ilustración N° 4.21 Funcionalidad para cerrar el curso activo.	81
Ilustración N° 4.22 Funcionalidad para consultar estudiantes del curso activo.	82
Ilustración N° 4.23 Funcionalidad para consultar cursos anteriores.	82
Ilustración N° 4.24 Funcionalidad para cerrar la inscripción del examen activo.	83
Ilustración N° 4.25 Funcionalidad para consultar examen activo.....	83
Ilustración N° 4.26 Funcionalidad para suspender inscripciones.	84
Ilustración N° 4.27 Funcionalidad para consultar próximas fechas de presentación.	84
Ilustración N° 4.28 Funcionalidad para agregar nueva fecha de presentación.	85

Ilustración N° 4.29 Funcionalidad para eliminar una fecha de presentación.	85
Ilustración N° 4.30 Funcionalidad para consultar una fecha de presentación específica.	86
Ilustración N° 4.31 Funcionalidad para consultar un estudiante.	86
Ilustración N° 4.32 Prototipo rápido para generar reportes.	88
Ilustración N° 4.33 Solicitud de constancia.	88
Ilustración N° 4.34 Descarga del reporte.	89

ÍNDICE DE TABLAS

Tabla N° 3.1 Metapatrón de Proceso.....	49
Tabla N° 3.2 Patrón de proceso de gestión para la inscripción de la Evaluación Instrumental.	50
Tabla N° 3.3 Patrón de proceso Solicitar inscripción.	51
Tabla N° 3.4 Patrón de proceso Suministrar datos para la inscripción.	51
Tabla N° 3.5 Patrón de proceso Seleccionar tipo de examen.	51
Tabla N° 3.6 Patrón de proceso Gestionar el examen y curso instrumental.	52
Tabla N° 3.7 Patrón de proceso Ofertar exámenes y cursos.	52
Tabla N° 3.8 Patrón de proceso Confirmar.	53
Tabla N° 3.9 Patrón de proceso Consultar.	53
Tabla N° 3.10 Patrón de proceso Gestionar noticias.	54
Tabla 3.11 Patrón de proceso de Calificar.....	54
Tabla 3.12 Patrón de proceso Enviar correos.	54
Tabla 3.13 Patrón de proceso Descargar reportes.....	55
Tabla N° 4.1 Requerimiento para Visualizar noticias.	58
Tabla N° 4.2 Requerimiento para Administrar el examen instrumental.....	58
Tabla N° 4.3 Requerimiento para realizar las Inscripciones del curso o prueba instrumental.	58
Tabla N° 4.4 Requerimiento para tener un Mecanismo que maneje correos electrónicos.	58
Tabla N° 4.5 Requerimiento para generar constancias y certificados.	59
Tabla N° 4.6 Requerimiento para Administrar la inscripción del curso y evaluación instrumental.	59
Tabla N° 4.7 Requerimiento para Almacenar exámenes.	59
Tabla N° 4.8 Patrón de dominio.	64
Tabla N° 4.9 Patrón de usuario Aspirante.	65
Tabla N° 4.10 Patrón de usuario Administrador.	65
Tabla N° 4.11 Patrón de sistema.	66
Tabla N° 4.12 Patrón de tarea Registro.....	66
Tabla N° 4.13 Patrón de tarea Formulario.	67
Tabla N° 4.14 Patrón de elemento compuesto Acordeón.	68
Tabla N° 4.15 Patrón de elemento compuesto Menú desplegable.	69
Tabla N° 4.16 Patrón de elementos simples Acceso directo de selección.....	69
Tabla N° 4.16 Tabla de evaluación heurística iteración 2.	74
Tabla N° 4.17 Tabla de evaluación heurística para la iteración 3.	87
Tabla N° 4.18 Tabla de evaluación heurística para la iteración 4.	89
Tabla N° 4.32 Lista de chequeo.....	90
Tabla N° 4.33 Resultado de la Lista de Comprobación aplicada a 10 usuarios.....	90

INTRODUCCIÓN

En la actualidad, las organizaciones son concebidas como entidades procesadoras de información, independientemente de su actividad, ya que usualmente tienen necesidad de obtener y analizar sobre mercados, costos, ventas y procesos de producción a todo nivel. Esta información procede de fuentes internas o externas para ser procesada y utilizada. Los sistemas de software que son diseñados para la automatización de procesos, son cada vez más demandados para el control de actividades dentro de cada organización. Además, gracias a que el internet ha tenido un gran auge en las últimas décadas, se ha provocado que se sustituyan los programas de escritorio por aplicaciones Web. Permitiendo de esta manera, que los usuarios finales se vean favorecidos por estos servicios.

Una de las organizaciones que se ha sumado a la implantación de aplicaciones Web, es la Escuela de Idiomas Modernos de la Facultad de Humanidades y Educación de la Universidad Central de Venezuela (UCV). Los procesos de inscripción y administración de los cursos de idiomas impartidos por esta Escuela, son algunos de los procedimientos que fueron implementados bajo este tipo de aplicación. No obstante, aún existen procesos que se realizan de forma manual y con poco uso de tecnología.

Este Trabajo se enfoca en el desarrollo de una aplicación Web que automatice las actividades relacionadas con el proceso de inscripción y gestión administrativa para la aplicación de la prueba instrumental en un idioma extranjero y las respectivas actividades logísticas para la generación de certificados, constancias y cursos que ofrece la Escuela de Idiomas Modernos a los distintos postgrados de las facultades de la UCV, con el fin de reducir costos en tiempo y en recursos humanos, para que puedan ser utilizados en labores de coordinación y planificación. Además, es imprescindible implementar y hacer uso de un conjunto de métodos y técnicas que garanticen la realización de un software de calidad. En este Trabajo el método a seguir para el desarrollo del software es AgilUs, propuesto por la profesora Alecia Eleonora Acosta del Centro de Ingeniería de Software y Sistemas (ISYS) de la Escuela de Computación de la Facultad de Ciencias de la UCV, que divide el desarrollo de un sistema de software en cuatro (4) etapas con distintas técnicas y artefactos para cada una, que enfoca el desarrollo en el usuario final y toma en cuenta la usabilidad desde etapas tempranas del desarrollo de software.

Este Trabajo Especial de Grado se estructura tal como se describe a continuación:

El Capítulo I titulado **Examen Instrumental** describe la problemática, donde se realiza el análisis de la gestión de la evaluación instrumental, en donde se detallan todos los procesos que se están relacionados tales como: la inscripción, la gestión administrativa y la generación de reportes, seguidamente se presentan el objetivo general y los objetivos específicos, la importancia y justificación de automatizar los procesos involucrados en el problema, mediante el desarrollo de una aplicación Web y los beneficios que generaría a la comunidad involucrada. Posteriormente, se lleva a cabo el alcance de la aplicación que expone las funcionalidades que el sistema está en la capacidad de ofrecer.

El Capítulo II titulado **Tecnologías Web** que representan las bases fundamentales del desarrollo de este trabajo. Se describen las aplicaciones Web Cliente/Servidor, su funcionamiento, arquitectura y características principales. Posteriormente, se describen las herramientas tecnológicas de software libre utilizadas para la realización de esta aplicación Web, agrupadas como del lado del cliente y del lado del servidor, incluyendo el Patrón de Diseño Modelo-Vista-Controlador (MVC) y los sistemas manejadores de bases datos.

El Capítulo III **Ingeniería Web** describe la importancia de utilizar un método de desarrollo de software para la creación de una aplicación Web. Además, se presentan las ventajas de los métodos ágiles, para así desarrollar el método AgilUs. Y gracias a esto, se puede hacer referencia a la importancia de la reusabilidad en la creación del software. Para concluir, se presentan los patrones de procesos que aportan una solución a problemas recurrentes, y serán desarrollados para la gestión de la inscripción del curso preparatorio y evaluación instrumental que lleva a cabo la Coordinación de Extensión de la Escuela de Idiomas Modernos.

El Capítulo IV detalla la **Implementación del Sistema**, se planifican y se llevan a cabo cuatro (4) iteraciones, la primera llamada Análisis Global del Sistema donde se encuentran las fases de requisitos y análisis de toda la aplicación, posteriormente la iteración Inscripción del curso y examen instrumental donde se encuentra el desarrollo de un módulo para esta funcionalidad, luego la Gestión Administrativa donde se lleva a cabo todas las tareas por parte del administrador del sistema, y por último, la Generación Reportes constituidos por constancias o certificados. Estas tres (3) últimas iteraciones serán constituidas por las fases de Prototipaje y Entrega. Además se realizan unas pruebas de aceptación para medir el grado de satisfacción de los usuarios. Finalmente, se exponen las conclusiones, recomendaciones y se listan las referencias bibliográficas que brindan soporte a esta investigación.

CAPÍTULO I EXAMEN INSTRUMENTAL

Este capítulo tiene como finalidad presentar el contexto del problema, relacionado con la gestión de la evaluación instrumental de la Escuela de Idiomas Modernos de la UCV, la cual consiste en la comprensión lectora de un idioma extranjero, para estudiantes de postgrado. Igualmente, se destaca el objetivo general, los objetivos específicos de esta investigación, importancia y justificación de automatizar los procesos involucrados, mediante el desarrollo de una aplicación Web. Finalmente, se presenta el alcance de la aplicación que expone las funcionalidades que el sistema estaría en la capacidad de ofrecer.

1.1 Situación problemática

La Escuela de Idiomas Modernos de la Facultad de Humanidades y Educación de la UCV, se encuentra automatizando diferentes procesos para obtener resultados óptimos sobre las actividades llevadas a cabo por ella. De las cuales se puede señalar, la inscripción y administración de los cursos de idiomas extranjeros suministrados por esta Escuela. No obstante, aún existen actividades que se siguen realizando de forma manual, con poco o ningún uso de la tecnología, como es el caso de los procesos que intervienen en el desarrollo y administración de la evaluación instrumental, los cuales son ejecutados semanalmente y las actividades relacionadas al curso instrumental se realizan cada siete (7) semanas, lo que conlleva a que exista un gran porcentaje de error humano, un cuantioso tiempo de respuesta y costos elevados, lo cual origina grandes desventajas a la organización.

Esta evaluación y curso instrumental se aplica a estudiantes de postgrado (en todas las áreas) de la Universidad Central de Venezuela, en un idioma extranjero, los cuales incluyen: inglés, alemán, francés, italiano y portugués. Cada aspirante que opta para aprobar dicho examen puede seleccionar el idioma de su preferencia, pero el único que está disponible para todas las áreas es el inglés. Además, el curso sólo se dicta en inglés.

Se podría decir que el primer inconveniente que posee la Coordinación de Extensión es la falta de medios para comunicar la información sobre los cursos y el examen instrumental. Debido a esto una cantidad considerable de personas llaman a diario para obtener información sobre: el costo, los requisitos, las fechas para realizar tanto el curso como el examen, entre otros.

Entre las actividades que se llevan a cabo en la Coordinación de Extensión se encuentran las siguientes:

1. Inscripción del curso.
2. Inscripción del examen.
3. Gestión administrativa para la evaluación instrumental.
4. Generación de constancia o certificado.

A continuación se describe de cada una de las tareas mencionadas que intervienen en la gestión de la evaluación instrumental de la Escuela de Idiomas Modernos:

1.1.1 Inscripción

Los aspirantes que desean realizar el examen instrumental deben llamar por teléfono a las instalaciones de la Coordinación de Extensión de la Escuela de Idiomas Modernos, y posteriormente se realizará el Proceso de Registro de los datos personales tales como: nombre, cédula, número de teléfono, área de estudio de postgrado que va a realizar el examen y el idioma deseado (solo si es diferente al inglés y si está disponible para el área de estudio de postgrado). De igual forma, se le indica el costo del examen, el número de cuenta a depositar, la fecha del próximo examen y se le pide confirmación para la fecha que desee presentar (solo se presentan los días sábados). En esta confirmación se debe realizar el registro del nombre de la persona que va a realizar el examen, número del depósito bancario y corroborar el área de estudio de postgrado que desee llevar a cabo, luego se le indicará la fecha, hora y lugar donde se efectuará prueba, además se le indican los recaudos y sugerencias, entre los que se encuentran: constancia de depósito bancario en físico, documento de identidad y un diccionario. Este proceso se realiza de lunes a jueves, en horario de oficina (8:00 am - 5:00 pm). A continuación en la Ilustración N° 1.1 se puede apreciar el proceso gráficamente.

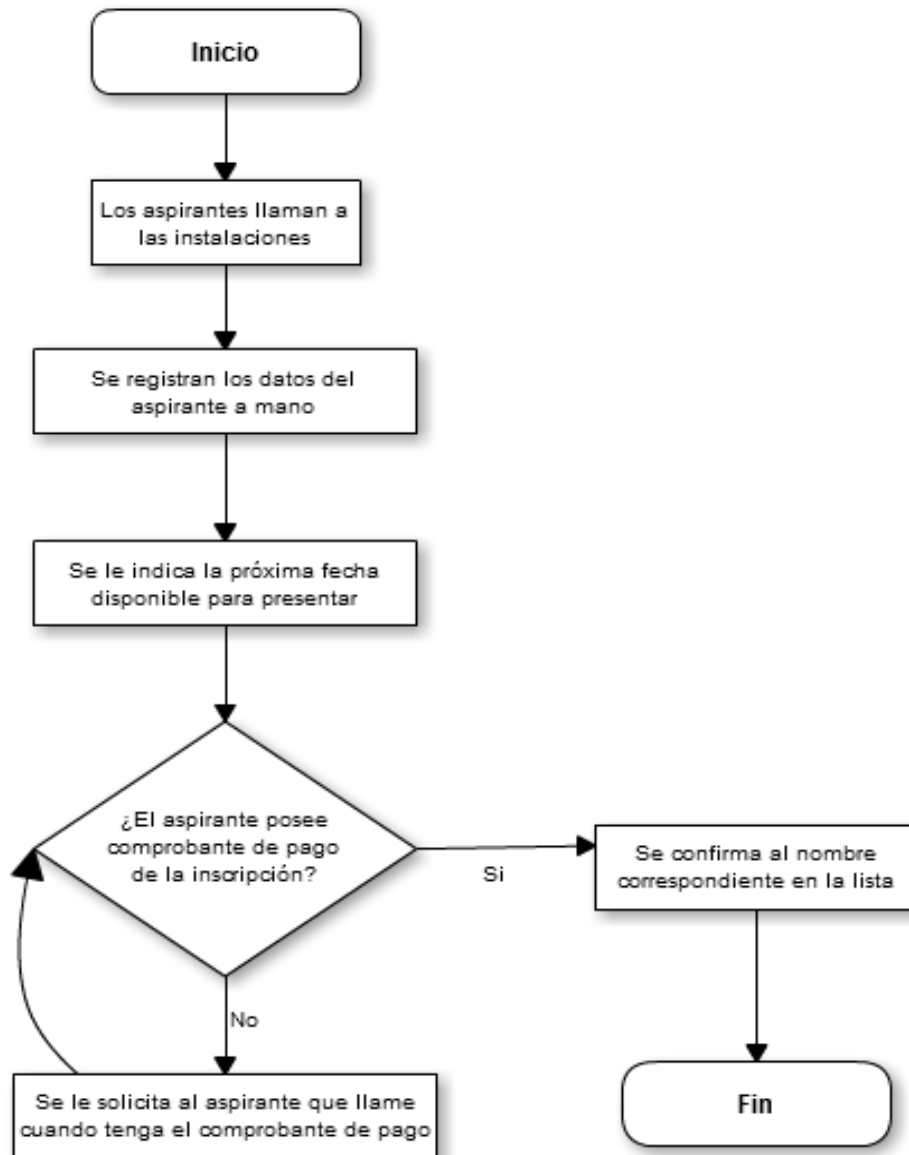


Ilustración N° 1.1 Inscripción para presentar el examen.

El proceso de inscripción para los cursos (mostrado en la Ilustración N° 1.2) es similar, los aspirantes deben contactar vía telefónica a las instalaciones de la Coordinación de Extensión de la Escuela de Idiomas Modernos y se registran los datos de identidad como nombre, número de cédula, número telefónico, correo electrónico y área de estudio de postgrado (las únicas áreas disponibles en la actualidad para presentar el curso son Odontología, Ciencias Económicas y Sociales, Medicina, y Derecho). Estos aspirantes se apuntan en una lista de espera, y cuando la Directiva decide abrir el curso se le informará por diferentes medios como correo electrónico,

mensajes de texto o llamadas telefónicas, especificándole la fecha de inicio, lugar, horario, pago y datos de la cuenta a depositar como institución financiera, monto, número y tipo de cuenta.

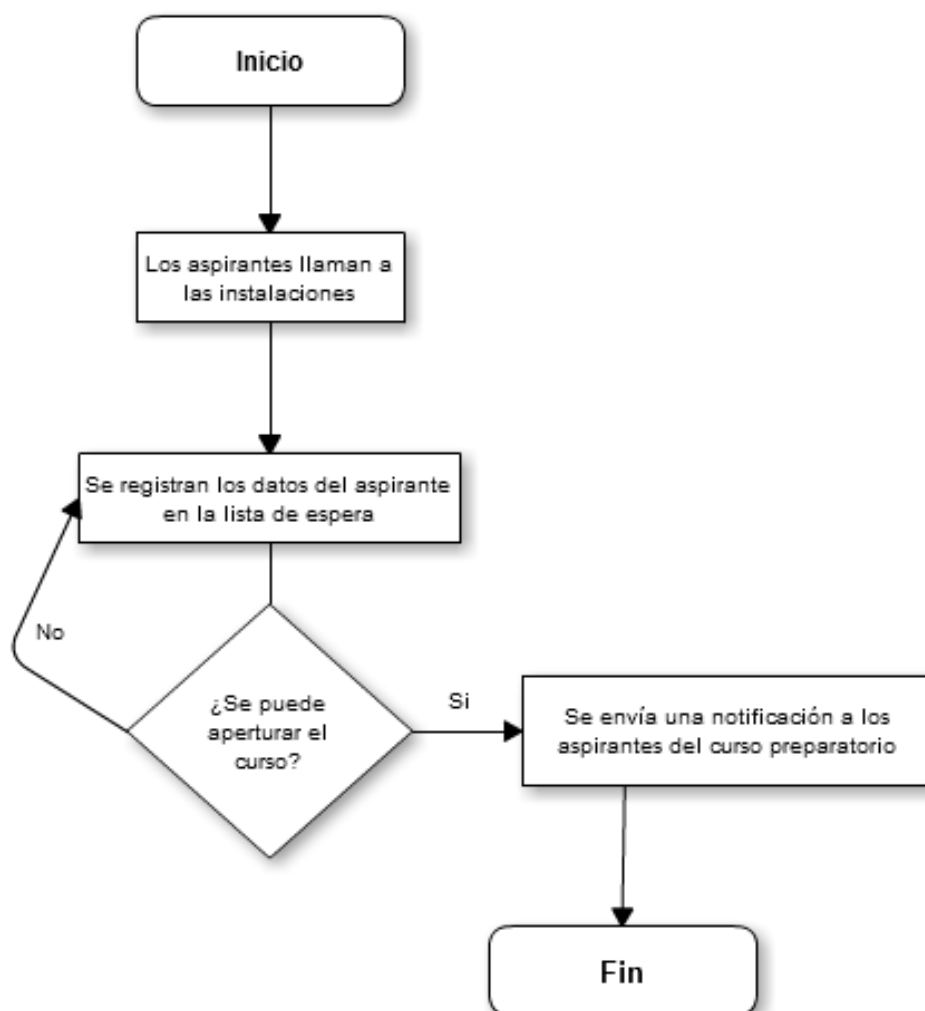


Ilustración N° 1.2 Inscripción para el curso.

1.1.2 Gestión administrativa para la evaluación instrumental

El personal directivo y administrativo de la Coordinación de Extensión, son los encargados de planificar y realizar toda la gestión del curso y examen instrumental. Entre sus actividades se encuentran:

- Calificar los exámenes y cursos para cada aspirante que presente.
- Coordinar las fechas de presentaciones de los exámenes.

- Organizar por postgrado a los aspirantes inscritos por cada fecha de presentación de examen.
- Informar a los aspirantes de los detalles en la inscripción tanto del curso como del examen instrumental.
- Enviar mensajes de texto, correos electrónicos o realizar llamadas para informar a los aspirantes sobre cambios en la fecha o lugar de presentación del examen, así como para informar la calificación de su evaluación.

1.1.3 Generación de constancia o certificado

Culminado el curso o examen instrumental los aspirantes que participaron en estos pueden solicitar al personal de la Coordinación de Extensión de la Escuela de Idiomas Modernos un certificado (para el curso) o una constancia (para el curso o la prueba). El certificado es realizado por los pasantes de la Escuela, según un formato establecido. Por otro lado, las constancias son realizadas por el Director en un software procesador de textos, el cual posee una plantilla y solo se cambian los datos de la persona que lo solicitó.

1.2 Planteamiento del problema

Cada una de las actividades, descritas en la sección anterior, se llevan a cabo de forma manual, siendo este el principal problema que se presenta. Asimismo, todos los exámenes que poseen se encuentran en físico. Esto, además de aumentar los costos en cuanto a recursos físicos, también incrementa el número de errores en el flujo de trabajo de las tareas, y los tiempos de respuestas. Otro problema que se presenta es que los procesos duran más tiempo en su ejecución y en la mayoría de las ocasiones la cantidad de material utilizado es sobreestimado. En este sentido se busca optimizar los procedimientos y recursos.

Al momento de la presentación del examen instrumental, no se lleva a cabo una logística adecuada debido a la falta de organización y gestión en este proceso. Adicionalmente, uno de los primordiales problemas para la Coordinación de Extensión de la Escuela de Idiomas Modernos es la comunicación poco efectiva que permite informar acerca de las características principales del curso y evaluación instrumental.

En síntesis, se puede observar que todos estos procedimientos se realizan de forma manual por diferentes miembros de la Coordinación de Extensión de la Escuela de Idiomas Modernos

presentando diferentes inconvenientes a lo largo de toda su ejecución, por lo que se propone desarrollar una aplicación que automatice y estandarice dichas actividades permitiendo así facilitar y agilizar dichos procesos.

1.3 Objetivo general

El Objetivo de este Trabajo Especial de Grado es desarrollar un módulo de gestión de procesos de inscripción, gestión administrativa y la generación de certificados y constancias para la evaluación instrumental que realiza la Escuela de Idiomas Modernos para los estudiantes de postgrado de las facultades de la UCV.

1.4 Objetivos específicos

- Diseñar las interfaces de usuarios correspondientes a los procesos de gestión de inscripción del taller y la evaluación instrumental, tomando en cuenta el perfil de usuario, a fin de proveer un producto usable con alto grado de satisfacción.
- Aplicar el método AgilUS, para el desarrollo de una aplicación Web usable que lleve a cabo los procesos inherentes a la evaluación instrumental y curso preparatorio.
- Diseñar e implementar la Base de Datos que permita almacenar la información relacionada con la gestión de la evaluación instrumental.
- Desarrollar un módulo para las inscripciones tanto del curso como de la evaluación instrumental para los estudiantes de postgrado de la UCV.
- Desarrollar un módulo para la gestión administrativa del curso y examen instrumental.
- Implementar un módulo para la generación de constancias y certificados del curso y la evaluación instrumental.
- Desarrollar una funcionalidad dentro de una aplicación Web que le permita a la Coordinación de Extensión de la Escuela de Idiomas de la UCV comunicarse con los usuarios, mostrando entre otras cosas, información del curso y la evaluación instrumental.

1.5 Justificación

La Coordinación de Extensión de la Escuela de Idiomas Modernos de la UCV es la encargada de desarrollar los cursos para aplicar la evaluación instrumental, así que efectúa todas las actividades

concernientes a los mismos. Además mantiene una comunicación constante con los estudiantes de postgrado para información o algún requerimiento en especial. Sin embargo, como estos procesos son totalmente manuales, pueden generar errores e inconvenientes, por lo tanto acarrear una gran cantidad de documentos impresos.

Se presenta un conjunto de soluciones a estas dificultades, con el objeto de aumentar el grado de satisfacción en los diferentes usuarios, tales como: aspirantes que optan por esta evaluación o curso, docentes, personal administrativo y directivo. Y de esta manera minimizar el porcentaje de error que pudiese existir a la hora de ejecutar estas tareas y/o actividades, apoyándonos en las bondades que ofrecen las aplicaciones Web.

Entre los beneficios que aporta la automatización de estos procesos, se pueden mencionar:

- Facilitar el trabajo y generar un enlace de comunicación efectivo entre el público y la Coordinación de Extensión de la Escuela de Idiomas Modernos.
- Mejorar los tiempos de respuesta y validar los datos.
- Disminuir el esfuerzo y el tiempo invertido.
- Disminuir los costos en recursos físicos, como el material impreso.
- Incrementar el grado de satisfacción de todas las personas involucradas en el proceso de la evaluación instrumental.

Además, se llevará a cabo el método de desarrollo de software AgilUS, debido a que esta aplicación Web estará centrada en los usuarios que la manejen, como el personal de la Coordinación de Extensión de la Escuela de Idiomas Modernos y los estudiantes de postgrado interesados en la evaluación o curso instrumental de la UCV. Y por lo tanto se requiere una de las cualidades que proporciona este método, usabilidad en el manejo de dicha aplicación.

1.6 Alcance

El alcance de esta investigación es el desarrollo de una aplicación Web de un módulo para implementar las funcionalidades concernientes a la gestión del proceso de evaluación instrumental, detalladas a continuación:

- Implementar funcionalidades que conllevan la inscripción tanto para el examen como para el curso instrumental.
- Implementar un módulo para la gestión administrativa del curso y examen instrumental.

- Implementar una herramienta que apoye a los profesores y/o personal administrativo como medio de comunicación que permita informar a los estudiantes acerca de los diferentes eventos durante el proceso de evaluación o curso instrumental.
- Implementar una solución eficiente para la generación de constancias y certificados.

En este capítulo se presentan los conceptos que sirven de fundamento teórico para el desarrollo de esta aplicación Web. El mismo detalla el concepto de Aplicaciones Web y las tecnologías utilizadas en la arquitectura Cliente/Servidor.

2.1 Aplicaciones Web

Se denominan aplicaciones Web aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor Web a través de Internet o de una Intranet mediante un navegador. En otras palabras, es una aplicación de software que se codifica en un lenguaje soportado por los navegadores Web en la que se confía la ejecución al navegador (*Aplicaciones Web, 2012*).

Una ventaja significativa es que las aplicaciones Web deberían funcionar de la misma manera, independientemente de la versión del sistema operativo instalado en el cliente. En vez de crear clientes para Windows, Mac OS, GNU/Linux y otros sistemas operativos, la aplicación Web se escribe una vez y se ejecuta igual en todas partes. Sin embargo, hay aplicaciones inconsistentes escritas con *HTML*, *CSS*, y otras especificaciones para navegadores Web que pueden causar problemas en el desarrollo y soporte de estas aplicaciones, principalmente debido a la falta de adicción de los navegadores a dichos estándares Web (especialmente versiones de Internet Explorer anteriores a la 7.0). Otras de las ventajas que poseen las aplicaciones Web se listan a continuación:

- Inmediatez de acceso: las aplicaciones Web no necesitan ser descargadas, instaladas y configuradas.
- Menos Bugs: las aplicaciones Web deberán ser menos propensas a colgarse y crear problemas técnicos debido a software o conflictos de hardware con otras aplicaciones existentes, protocolos o software personal interno. Con aplicaciones basadas en Web, todos utilizan la misma versión, y todos los bugs pueden ser corregidos tan pronto como son descubiertos.
- Múltiples usuarios concurrentes: las aplicaciones basadas en Web pueden realmente ser utilizada por múltiples usuarios al mismo tiempo.

Las aplicaciones Web funcionan bajo una arquitectura llamada Cliente-Servidor, la cual es abarcada a continuación.

2.1.1 Arquitectura Cliente-Servidor

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, que le da respuesta.

Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras (*Arquitectura Cliente/Servidor, 2012*).

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un solo programa. Los tipos específicos de servidores incluyen los servidores Web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

La arquitectura cliente-servidor sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico.

La red cliente-servidor es aquella red de comunicaciones en la que todos los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta; y que los pone a disposición de los clientes cada vez que estos son solicitados. Esto significa que todas las gestiones que se realizan se concentran en el servidor, de manera que en él se disponen los requerimientos provenientes de los clientes que tienen prioridad, los archivos que son de uso público y los que son de uso restringido, los archivos que son de solo lectura y los que, por el contrario, pueden ser modificados, etc. Este tipo de red puede utilizarse conjuntamente en caso de que se esté utilizando en una red mixta.

El funcionamiento de la arquitectura cliente- servidor se aprecia en la Ilustración N° 2.1 tomada de (*Arquitectura Cliente-Servidor, 2012*).

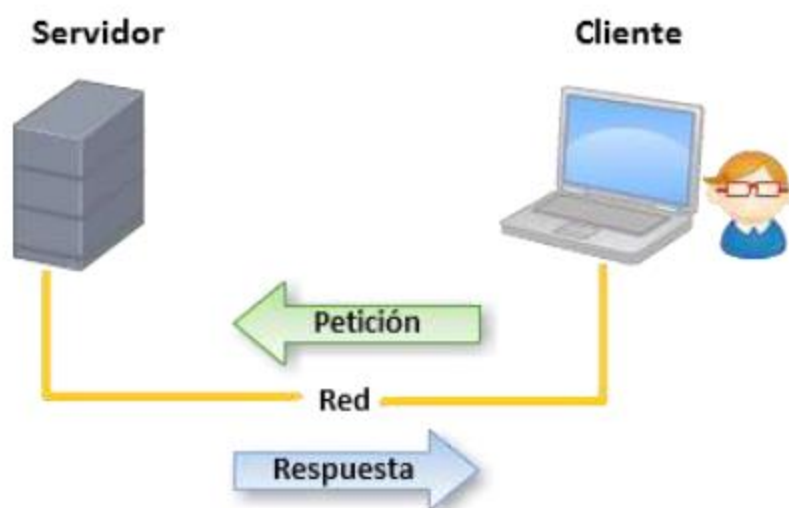


Ilustración N° 2.1 Arquitectura Cliente - Servidor.

Esta arquitectura se basa en el paradigma de separar el código de presentación, de procesamiento y de almacenamiento con el fin de hacer modular el trabajo del desarrollador Web.

En términos generales, la capa de presentación proporciona la interfaz necesaria para presentar información y reunir datos. La capa de procesamiento responde a peticiones del usuario para ejecutar una tarea en específico, interactuando con los datos que están almacenados. La capa de almacenamiento representa las fuentes de datos finales y está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento y reciben solicitudes de recuperación de información desde la capa de procesamiento.

En los primeros tiempos de la computación cliente-servidor, cada aplicación tenía su propio programa cliente y su interfaz de usuario, estos tenían que ser instalados separadamente en cada estación de trabajo de los usuarios. Una mejora al servidor, como parte de la aplicación, requería típicamente una mejora de los clientes instalados en cada una de las estaciones de trabajo, añadiendo un costo de soporte técnico y disminuyendo la eficiencia del personal.

En contraste, las aplicaciones Web generan dinámicamente una serie de páginas en un formato estándar, soportado por navegadores Web comunes, como *HTML* o *XHTML*, y actualmente *HTML 5* la cual es la quinta revisión importante del lenguaje básico de la Red Informática Mundial o

World Wide Web (WWW), *HTML*. Adicionalmente se utilizan lenguajes interpretados del lado del cliente, tales como *JavaScript*, para añadir elementos dinámicos a la interfaz de usuario. También se han desarrollado tecnologías para coordinar estos lenguajes con tecnologías del lado del servidor, como por ejemplo *PHP*, *Ruby on Rails*, *JSP*, entre otros.

2.2 Tecnologías del lado del Cliente

Las tecnologías del lado del cliente, son aquellas que nos permiten procesar o interpretar a través de un navegador las peticiones que se hacen del lado del cliente. Estas tecnologías se detallan a continuación:

2.2.1 HyperText Markup Language, versión 5 (HTML 5)

HTML 5 es una colección de estándares para el diseño y desarrollo de páginas Web. Esta colección representa la manera en que se presenta la información en el navegador de internet y la manera de interactuar con ella. *HTML 5* es una nueva versión de diversas especificaciones como *HTML 4* y *XHTML 1* (Hogan, 2012).

A la par, *HTML 5* pretende proporcionar una plataforma para desarrollar aplicaciones Web más parecidas a las aplicaciones de escritorio, donde su ejecución dentro de un navegador no implique falta de recursos o facilidades para resolver las necesidades reales de los desarrolladores.

HTML 5 incluye novedades significativas en diversos ámbitos:

- Estructura del cuerpo: la mayoría de los sitios Web tienen un formato común, formado por elementos como cabecera, pie, navegadores, entre otros. *HTML 5* permite agrupar todas estas partes de una Web en nuevas etiquetas que representarán cada uno de las partes típicas de una página.
- Etiquetas para contenido específico: se utilizaba una única etiqueta para incorporar diversos tipos de contenido enriquecido, como animaciones Flash o video. Ahora se utilizan etiquetas específicas para cada tipo de contenido en particular, como audio, video, entre otros.
- Canvas: es un nuevo componente permite dibujar, por medio de las funciones de un API para mejorar el funcionamiento nativo de HTML5, en la página todo tipo de formas, que podrán estar animadas y responder a interacción del usuario y sin la necesidad de tener instalado ningún complemento en el navegador.

- Bases de datos locales: el navegador permite el uso de una base de datos local, con la que se podrá trabajar en una página Web por medio del cliente y a través de un API. Es algo análogo a las Cookies, pero pensadas para almacenar grandes cantidades de información, lo que permitirá la creación de aplicaciones Web que funcionen sin necesidad de estar conectados a Internet.
- Web Workers: son procesos que requieren bastante tiempo de procesamiento por parte del navegador, pero que se podrán realizar en un segundo plano, para que el usuario no tenga que esperar que se terminen para empezar a usar la página.
- Aplicaciones Web Offline: existe otro API para el trabajo con aplicaciones Web, que se podrán desarrollar de modo que funcionen también en local y sin estar conectados a Internet.
- Geo localización: las páginas Web se pueden localizar geográficamente.
- Fin de las etiquetas de presentación: todas las etiquetas que tienen que ver con la presentación del documento, que modifican estilos de la página, son eliminadas, por ejemplo las etiquetas <center>, <left> y <right> para posicionar algún elemento de la página. La responsabilidad de definir el aspecto de una Web está a cargo únicamente de las hojas de estilo en cascada CSS o CSS 3, de lo cual se hace referencia a continuación.

2.2.2 Cascading Style Sheets, versión 3 (CSS 3)

Las hojas de estilo en cascada se crearon para separar el contenido de la forma, a la vez permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas. Una de las características principales de CSS es su flexibilidad y las diferentes opciones que ofrece para realizar una misma tarea (*Eguíluz, 2012 a*).

Desde que CSS comenzó han pasado muchos años y ya se encuentra por la especificación de CSS 3, la cual incorpora una serie de novedades cuyo objetivo principal es que los desarrolladores tengan un mayor control sobre el estilo con el que se muestran los elementos de las páginas.

Las ventajas principales en esta nueva versión son la inclusión de nuevas propiedades especialmente en cuanto al aspecto gráfico. Incluye bordes redondeados, textos con sombras, la capacidad de asignar múltiples fondos, un mejor manejo de tabla, entre otros. Con las nuevas propiedades, la carga de la página disminuye pues el hecho de que muchos efectos estén bajo el control del navegador hace que diversos recursos visuales e imágenes usadas anteriormente ya no tengan razón de seguir siendo utilizados (*Hogan, 2012 a*).

A continuación se hará referencia a un lenguaje de programación para realizar páginas Web dinámicas mediante la interacción con las tecnologías ya descritas *HTML 5* y *CSS 3*.

2.2.3 JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas Web dinámicas. Una página Web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario (Eguíluz, 2012 b).

Técnicamente, *JavaScript* es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con *JavaScript* se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Este lenguaje se define como orientado a objetos.

Entre las acciones típicas que se pueden realizar en *JavaScript* tenemos dos vertientes. Por un lado los efectos especiales sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, *JavaScript* nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

A pesar de su nombre, *JavaScript* no guarda ninguna relación directa con el lenguaje de programación Java, y actualmente se encuentra en su versión 1.8.5.

En la actualidad existen diversas bibliotecas que extienden *JavaScript*; una de ellas es *jQuery* y es presentada a continuación.

2.2.4 jQuery

jQuery es una biblioteca de *JavaScript*, actualmente en su versión 1.7.2, que permite a los usuarios aplicar funcionalidades dinámicas a las páginas Web con gran facilidad. *jQuery* ofrece varias características de gran alcance, incluyendo la posibilidad de acceder a una parte de una red la página, añadir animaciones, y mucho mas (Harwani, 2012).

Esta biblioteca es rápida, concisa y simplifica la manera de interactuar con los documentos *HTML*, así como también el manejo de eventos, animación, y las interacciones para *JavaScript* asíncrono y

XML para el desarrollo Web rápido. *jQuery* consiste en un único fichero *JavaScript* que contiene las funcionalidades comunes de manejo de los elementos del *Modelo de Objetos del Documento* o *Document Object Model (DOM)*, eventos y efectos. Adicionalmente una de las características consideradas más importante de *jQuery* es que es no intrusivo, es decir, que está totalmente separado de la información y estructura sintáctica del sitio. *jQuery* está diseñado para cambiar la forma en que se escribe *JavaScript*.

La forma de interactuar con la página es mediante la función `$()`, un alias de *jQuery()*, que recibe como parámetro una expresión *CSS* o el nombre de una etiqueta del documento *HTML* y devuelve todos los nodos (elementos) que concuerden con la expresión.

A continuación se listan de forma concreta las características de esta biblioteca:

- Selección de elementos, interactividad y modificaciones del árbol *DOM*.
- Manejo de eventos.
- Manipulación de la hoja de estilos *CSS*.
- Efectos y animaciones.
- Animaciones personalizadas.
- Funcionalidades de interacciones para *JavaScript* asíncrono y *XML*.
- Soporta extensiones.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones como `trim()` (elimina los espacios en blanco del principio y final de una cadena de caracteres), entre otros.
- Compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+.

A partir de *jQuery* surge otra biblioteca como complemento de esta, dedicada principalmente a efectos visuales como *jQuery User Interface* descrita a continuación.

2.2.5 jQuery User Interface (jQuery UI)

jQuery UI es una biblioteca de componentes para *jQuery* que le añaden un conjunto de complementos, widgets y efectos visuales para la creación de aplicaciones Web, actualmente en su versión 1.8.20 (*jQuery UI*, 2012).

La biblioteca se divide en cuatro módulos:

- Núcleo: contiene las funciones básicas para el resto de módulos como lo son:
- Core: requerido para todas las interacciones y widgets.
- Widget: la fábrica de widget, base para todos los widgets.
- Mouse: el widget de ratón, una clase base para todas las interacciones y widgets con la interacción a través del ratón.
- Position: una extensión de utilidad para los elementos de posicionamiento en relación con otros elementos.
- Interacciones: añade comportamientos complejos a los elementos, por ejemplo:
 - Draggable: permite que cualquier elemento de la página se pueda arrastrar.
 - Droppable: habilita elementos donde los elementos que fueron arrastrados son liberados.
 - Resizable: permite que a cualquier elemento de la página se le pueda modificar su tamaño.
 - Selectable: crea una lista de elementos seleccionables a través del ratón arrastrando un cuadro o haciendo clic en ellos.
 - Sortable: hace una lista de elementos los cuales se pueden ordenar.
- Widgets: es un conjunto completo de controles de interfaz. Cada control tiene un conjunto de opciones configurables y se les puede aplicar estilos CSS, por ejemplo:
 - Accordion: crea un widget de navegación en forma de acordeón.
 - Autocomplete: crea un widget de autocompletado.
 - Button: crea un widget de botón.
 - Dialog: abre un cuadro de diálogo que puede arrastrarse y redimensionarse.
 - Slider: crea un control deslizable con rangos y accesible a través del teclado.
- Tabs: transforma un conjunto de contenedores en una estructura de pestañas.
- Datepicker: un selector de fechas que puede ser activado a través de un elemento de entrada.
- Progressbar: un indicador de estado que puede ser utilizado para un estado de carga.
- Efectos: una API para añadir transiciones animadas y facilidades para interacciones.

Una vez descrito *JavaScript* y la biblioteca *jQuery*, pasemos ahora a conocer una técnica de desarrollo que utiliza ambas tecnologías.

2.2.6 Asynchronous JavaScript And XML (AJAX)

AJAX significa *JavaScript* asíncrono y *XML*. Aunque en sentido estricto *AJAX* no es en sí una tecnología, mezcla técnicas de programación bien conocidas de una manera poco común para que los desarrolladores Web creen aplicaciones con interfaces de usuario mucho más atractivas (Ballard & Moncur, 2012).

Es una técnica de desarrollo Web para crear aplicaciones interactivas mediante la combinación de tres tecnologías ya existentes: *HTML* y Hojas de Estilo en Cascada (*CSS*) para presentar la información; *JavaScript*, para interactuar dinámicamente con los datos, y lenguaje de marcado extensible o *eXtensible Markup Language (XML)* y para intercambiar y manipular datos con un servidor Web (aunque las aplicaciones que usan *AJAX* pueden usar otro tipo de formato, incluyendo texto plano, para realizar esta labor).

AJAX es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. *JavaScript* es el lenguaje interpretado en el que normalmente se efectúan las funciones de llamada de *AJAX* mientras que el acceso a los datos se realiza mediante *XMLHttpRequest*.

En las aplicaciones Web tradicionales los usuarios interactúan mediante formularios, que al enviarse, realizan una petición al servidor Web. El servidor se comporta según lo enviado en el formulario y contesta enviando una nueva página Web. Se desperdicia mucho ancho de banda, ya que gran parte del *HTML* enviado en la segunda página Web, ya estaba presente en la primera.

Además, de esta manera no es posible crear aplicaciones con un alto grado de interacción. En la Ilustración N° 2.2 tomada de (Ballard & Moncur, 2012) se ve el funcionamiento de una aplicación Web tradicional.

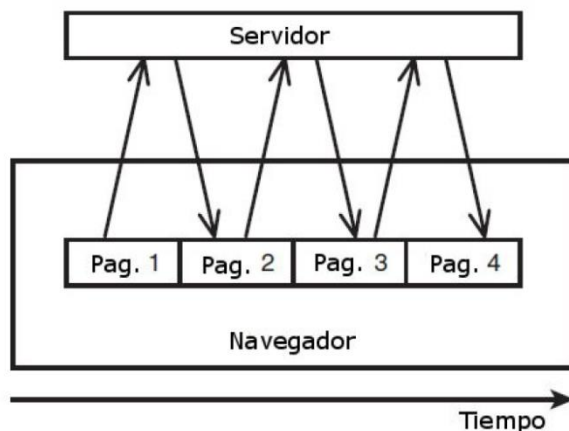


Ilustración N° 2.2 Funcionamiento de una aplicación Web tradicional.

En aplicaciones *AJAX* se pueden enviar peticiones al servidor Web para obtener únicamente la información necesaria, empleando *Simple Object Access Protocol (SOAP)* o algún otro protocolo para servicios Web basado en *XML*, y usando *JavaScript* en el cliente para procesar la respuesta del servidor Web. Esto resulta en una mayor interacción gracias a la reducción de información intercambiada entre servidor y cliente, además de que parte del proceso lo hace el propio cliente, liberando al servidor de ese trabajo. Lo negativo de este proceso es que la descarga inicial de la página es más lenta al tener que obtener todo el código *JavaScript*. En la Ilustración N° 2.3 tomada de (Ballard & Moncur, 2012) se ve el funcionamiento de una aplicación Web con *AJAX*.

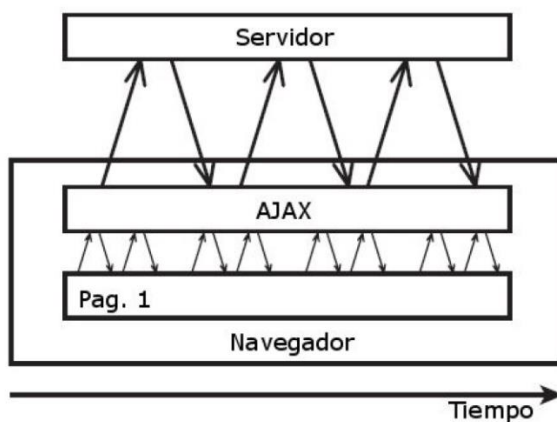


Ilustración N° 2.3 Funcionamiento de una aplicación Web con AJAX.

2.3 Tecnologías del lado del Servidor

Para complementar las tecnologías que usaremos, debemos mencionar las del lado del servidor. Que poseen la habilidad de correr programas que interactúan con los clientes. Estos pequeños programas residen en los servidores Web y permiten tener interactividad en los sitios Web. Por ejemplo, si en tu sitio Web necesitas que todos los usuarios entren con contraseña, la base de datos que contiene los nombres de usuario y contraseñas para acceder al sitio Web, y el programa que ofrece la autenticación de acceso se encontrará en el servidor Web.

2.3.1 Plataforma Debian GNU/Linux

Para empezar con estas tecnologías debemos hablar del sistema operativo, que en este caso se desarrolló gracias al Proyecto Debian, el cual es una asociación de personas que han hecho causa común para crear un sistema operativo libre. Y se llama Debian GNU/Linux (*Debian, 2012*).

Un sistema operativo es un conjunto de programas y utilidades básicas que hacen que su computadora funcione. El centro de un sistema operativo es el núcleo. El núcleo es el programa más importante en la computadora, realiza todo el trabajo básico y le permite ejecutar otros programas.

Los sistemas Debian actualmente usan el núcleo de Linux. El cual es una pieza de software creada en un principio por Linus Torvalds y soportada por miles de programadores a lo largo del mundo.

Debian se caracteriza por:

- La disponibilidad en varias arquitecturas
- Una amplia colección de software disponible.
- Un grupo de herramientas para facilitar el proceso de instalación y actualización del software (APT, Aptitude, Dpkg, Synaptic, Dselect, entre otras.) Todas ellas obtienen información de donde descargar software, desde la ruta `/etc/apt/sources.list`, que contiene los repositorios.
- Su compromiso con los principios y valores involucrados en el movimiento del Software Libre.

2.3.2 Apache

El software servidor que se va a utilizar en el sistema operativo antes mencionado será el *HTTP* Apache de código abierto para plataformas Unix, Windows, Macintosh y otras, que implementa el

protocolo *HTTP* y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA *HTTPd* 1.3, pero más tarde fue reescrito por completo (Apache, 2012).

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. La arquitectura del servidor Apache es modular, lo cual implica que puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la Interfaz de Programación de Aplicaciones (API) necesaria. El servidor consta de una sección núcleo y diversos módulos que aportan muchas funcionalidades que podrían considerarse básicas para un servidor Web.

Entre las ventajas de utilizar Apache se encuentran que es multiplataforma, gratuito, extensible, fácil para conseguir ayuda y/o documentación.

2.3.3 Unicorn

Otro servidor que se va a utilizar es Unicorn, el cual es un servidor *HTTP* para aplicaciones Rack diseñado para servir rápidamente a los clientes en baja latencia, las conexiones de gran ancho de banda y aprovechar las características de Unix. Los clientes lentos solo deben ser atendidos por la colocación de un Proxy inverso capaz de amortiguar tanto la solicitud como la respuesta en medio de Unicorn y los clientes lentos (Unicorn, 2012).

Esta aplicación estará basada en el patrón de diseño MVC (Modelo Vista Controlador) debido a su simplicidad y sencillez en su implementación.

2.3.4 Patrón de Diseño MVC (Modelo Vista Controlador)

La arquitectura de tres capas se refleja en el patrón de arquitectura *MVC* (Modelo-Vista-Controlador), desarrollado con el propósito reducir el acoplamiento, o grado de dependencia entre módulos, entre la lógica de negocios y la de presentación.

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. Este patrón se ve frecuentemente en aplicaciones Web, donde la *vista* es la página *HTML* y el código que provee de datos dinámicos a la página. El *modelo* es el encargado de la gestión de base de datos y la lógica de negocio, y el *controlador* es el responsable de recibir los datos de entrada desde la *vista* (MVC, 2012).

La finalidad de este modelo es mejorar la reusabilidad por medio del desacoplo entre la *vista* y el *modelo*. Los elementos del patrón son *modelo*, *vista* y *controlador*. El *modelo* es la representación específica de la información con la cual el sistema opera. La *vista* presenta el *modelo* en un formato adecuado para interactuar, usualmente la interfaz de usuario. Finalmente el *controlador* responde a eventos, usualmente acciones del usuario, e invoca peticiones al *modelo* y, probablemente, a la *vista*. Dicha interacción se puede ver reflejada en la Ilustración N° 2.4 tomada de (MVC, 2012).

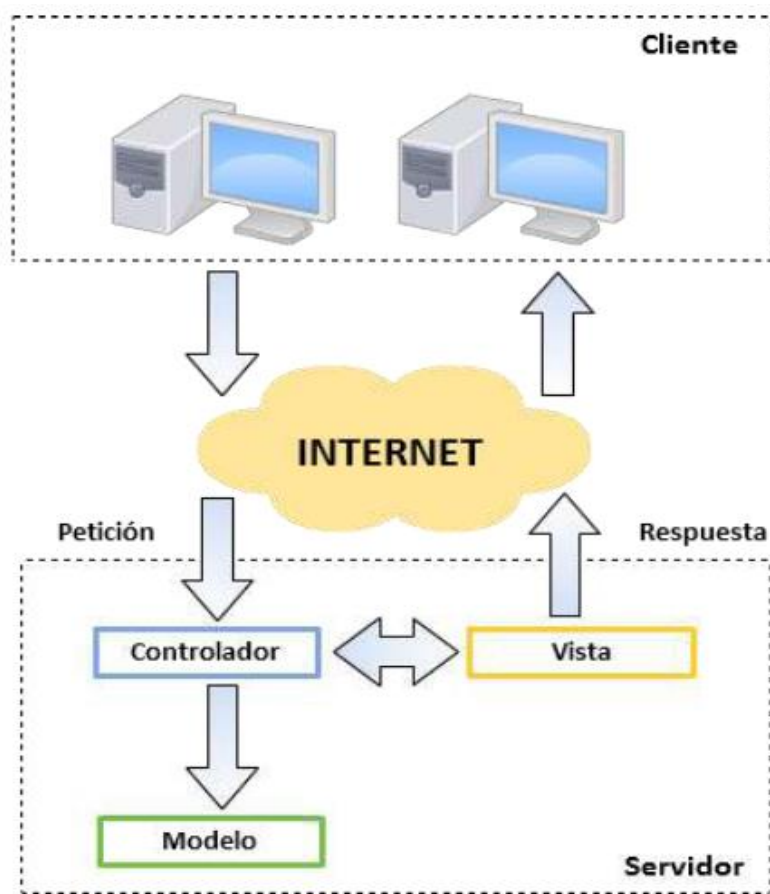


Ilustración N° 2.4 Patrón de diseño MVC.

Este modelo de arquitectura presenta varias ventajas:

- Hay una separación entre los componentes de un programa; lo cual nos permite implementarlos por separado.
- Al incorporar el modelo de arquitectura MVC a un diseño, los módulos de un programa se pueden construir por separado y luego unirlos en tiempo de ejecución.

- Facilidad de desarrollo y acortamiento del tiempo de respuesta gracias a las tareas paralelizadas.
- Aumenta en gran medida el nivel de reusabilidad de código. Facilita una evolución continua de los sistemas, sin puntos de ruptura, ya que un cambio en un sistema afectará a uno o más componentes pero nunca al núcleo de la aplicación de manera significativa.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

- El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace).
- El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra).
- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Para implementar la aplicación Web, se hará uso de un framework llamado Ruby on Rails. Unicorn es compatible con Ruby versión 1.8 y 1.9.

2.3.5 Ruby on Rails

Este framework permite facilidad en el desarrollo, implementación y en el mantenimiento de la misma. Durante los meses que siguieron a su lanzamiento inicial, Rails paso de ser un juguete desconocido a ser un fenómeno en todo el mundo, y más importante, se ha convertido en el

marco de elección para la aplicación de una amplia gama de las llamadas aplicaciones Web 2.0 (*Ruby & Thomas & Hansson, 2012*).

Lo antes mencionado se debe a que en primer lugar, un gran número de desarrolladores se sienten frustrados con las tecnologías que se utilizan para crear aplicaciones Web. No parece importar si se utiliza Java, PHP, o. NET, hubo una creciente sensación de que trabajarlas es una tarea muy difícil. Y luego, de repente, llegó Rails, y todo fue más fácil. Sin embargo, fácil de por sí no es suficiente. Los desarrolladores querían sentir que las aplicaciones que estaban desarrollando pasarían la prueba del tiempo en que se han diseñado e implementado, utilizando técnicas modernas y profesionales.

Todas las aplicaciones Rails se implementan utilizando el Modelo-Vista-Controlador (MVC). Sin embargo, cuando se desarrolla en Rails, se comienza con una solicitud de trabajo, hay un lugar de cada pieza de código, y todas las partes de la aplicación interactúan en una forma estándar. Los programadores profesionales pueden escribir pruebas. A medida que se agregan funcionalidades al código, Rails crea automáticamente talones de prueba para esa funcionalidad.

Las aplicaciones de Rails están escritas en Ruby, un lenguaje moderno y orientado a objetos de secuencias de comandos. Ruby es conciso, puede expresar las ideas de manera natural. Esto conduce a que los programas sean fáciles de escribir y fáciles de leer meses después. Rails tiene Ruby para el límite, que se extiende en nuevas formas que hacen que la vida de programador sea más sencilla. Lo que implica que los programas puedan ser más cortos y más legibles.

Dos fundamentos filosóficos para mantener el código de Rails corto y legible: *DRY* y la convención sobre la configuración. *DRY* se refiere a que no te repitas: cada pieza de conocimiento en un sistema se debe expresar en un solo lugar. Rails utiliza la potencia de Ruby para traer esto a la vida. Es poco común encontrar una duplicación en una aplicación Rails. Para los programadores que utilizan otros entornos Web, donde un simple cambio en el esquema podría involucrarlos en una media docena o más cambios en el código se creó lo que se llama convención sobre la configuración. Esto quiere decir que Rails tiene parámetros por defecto para casi todos los aspectos de género de punto, junto a su solicitud.

Si se siguen las convenciones, se puede escribir una aplicación Rails con menos código que una típica aplicación Java Web, la cual utiliza una configuración de XML. Rails permite facilidad para los desarrolladores al momento de integrar características tales como Ajax y REST en su código, ya

que el apoyo está construido adentro. Además brinda simplicidad y sencillez al usar un patrón de diseño Modelo - Vista – Controlador.

Debido a que la aplicación Web va a manejar grandes y complejos volúmenes de datos, es necesario hacer uso de los servidores de bases de datos, para así poder compartir la información con un conjunto de clientes de manera segura. Ante este enfoque, un Sistema Manejador de Base de Datos deberá ofrecer soluciones de forma fiable, rentable y de alto rendimiento.

2.3.6 MySQL

El sistema manejador de bases de datos relacional (SMBDR) llamado MySQL, es multihilo y multiusuario. MySQL AB es una compañía comercial de software libre, que distribuye, soporta y además lo desarrolla bajo un esquema de licencia dual. Por un lado se tiene la licencia GNU GPL (*General Public License GNU*) y por otro, aquellas empresas que desean utilizar este sistema en productos privados, pueden comprar la licencia que les permita su uso.

Actualmente MySQL es el sistema de gestión de bases de datos más popular del mundo con más de 6 millones de instalaciones alrededor del mundo, con aproximadamente 40 mil descargas del software. Es ampliamente utilizado en aplicaciones Web, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores. Su popularidad en las aplicaciones Web está muy ligada a PHP, que a menudo aparece en combinación con MySQL, sin embargo existen varias APIs que permiten a las aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, como son C, C++, C#, Pascal, Delphi (vía dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), entre otros.

Algunas características de MySQL son las siguientes (*MySQL, 2012*):

- Interioridad y portabilidad.
 - Escrito en C y C++.
 - Es multiplataforma. Funciona en plataformas como AIX, BSD, FreeBSD, HP - UX, GNU/Linux, Mac OS X, NetBSD, Windows 95/98/NT/2000/XP/Vista y otras versiones de Windows.
 - Es multihilo. Utiliza hilos del kernel, aprovechando así la ventaja de usar multiprocesamiento con varios procesadores.
 - Proporciona sistemas de almacenamiento transaccional y no transaccional.

- Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, Memory/heap, entre otros), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.
- Tipos de Datos.
 - Posee diversos tipos de datos: enteros con/sin signo de 1, 2, 3, 4, y 8 bytes de longitud, float, double, char, varchar, text, blob, date, time, datetime, timestamp, year, set, enum, y tipos espaciales OpenGIS.
 - Tiene registros de longitud fija y longitud variable.
- Sentencias y funciones.
 - Soporte completo para operadores y funciones en las cláusulas de consultas Select y Where.
 - Soporte completo para las cláusulas sql group by y order by. Soporte de funciones de agrupación (count(), count(distinct ...), avg(), std(), sum(), max(), min(), y group_concat()).
 - Puede mezclar tablas de distintas bases de datos en la misma consulta.
- Seguridad.
 - Tiene un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el Host. Las contraseñas son seguras porque todo el tráfico de contraseñas esta encriptado cuando se conecta con un servidor.

Este capítulo tiene como finalidad analizar la importancia de la Ingeniería Web en la creación de una aplicación Web, así como el propósito de llevar a cabo un método de desarrollo de software ágil que permita interactuar con el cliente y desarrollar incrementalmente el software, mostrando versiones parcialmente funcionales al mismo en intervalos cortos de tiempo. Y con esto se pueda evaluar y sugerir cambios en el producto según se va desarrollando. Asimismo, se explica la reusabilidad y su importancia.

Adicionalmente, se presentan los patrones de procesos, apoyados por la reusabilidad, obtenidos del desarrollo de la inscripción para el curso preparatorio y la evaluación instrumental que lleva a cabo la Coordinación de Extensión de la Escuela de Idiomas Modernos.

3.1 Ingeniería Web

Internet ha introducido a la población en general en el mundo de la informática. Se compra fondos de inversión colectivos y acciones, se descarga música, se obtiene asesoramiento médico, se recibe cursos universitarios, es decir, en el mundo virtual se puede hacer todo lo que se necesite. Se puede decir que Internet y la Web son los avances más importantes en la historia de la informática.

Estas tecnologías nos han llevado a todos nosotros a la era de la informática. Durante los primeros años del siglo veintiuno estas tecnologías han llegado casi a formar parte de nuestra vida diaria (*Ingeniería, 2012*).

Con objeto de evitar una Web enmarañada y lograr un mayor éxito en el desarrollo y aplicación de sistemas basados en Web complejos y a gran escala, existe una necesidad apremiante de enfoques de ingeniería Web disciplinada, métodos y herramientas nuevas para el desarrollo, empleo, evaluación de sistemas y aplicaciones basados en dicho concepto. Tales enfoques y técnicas deberán tener en cuenta las características especiales en el medio nuevo, en los entornos, escenarios operativos, y en la multiplicidad de perfiles de usuario implicando todo ello en un reto adicional para el desarrollo de aplicaciones.

La Ingeniería Web está relacionada con el establecimiento y utilización de principios científicos, de ingeniería, de gestión, con enfoques sistemáticos y disciplinados del desarrollo, empleo y

mantenimiento de aplicaciones basadas en Web de alta calidad. Por lo que al crear y desarrollar un sistema Web es fundamental llevar a cabo una metodología de desarrollo de software.

3.2 Método de Desarrollo de Software

Los Métodos de Desarrollo de Software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto software (*Métodos, 2012*).

Estos pretenden guiar a los desarrolladores al crear un nuevo software, pero los requisitos de un software a otro son tan variados y cambiantes, que ha dado lugar a que exista una gran variedad de métodos para el desarrollo del software. Se podrían clasificar en dos grandes grupos:

- Las metodologías orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán. Estas metodologías son llamadas Pesadas.
- Las metodologías orientadas a la interacción con el cliente y el desarrollo incremental del software, mostrando versiones parcialmente funcionales del software al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando. Son llamadas metodologías ligeras/ágiles.

Las metodologías pesadas son las más tradicionales, se centran en la definición detallada de los procesos, tareas a realizar, herramientas a utilizar y requiere una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodologías son más eficaces y necesarias cuando mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear, donde una gran organización es requerida.

Como una reacción a estas metodologías, un nuevo grupo ha surgido en los últimos años. Durante algún tiempo se conocían como ligeras, pero el término aceptado ahora es ágil. Para mucha gente el encanto de ellas es su reacción ante la burocracia de las metodologías monumentales o pesadas.

En consecuencia, se crean los métodos ágiles, los cuales cambian significativamente algunos de los énfasis de los métodos pesados. La diferencia inmediata es que son menos orientados al documento, exigiendo una cantidad más pequeña de documentación para una tarea dada.

Los métodos ágiles son adaptables en lugar de predictivos. Los pesados tienden a intentar planear una parte grande del proceso del software en gran detalle para un plazo largo de tiempo, y funciona bien hasta que las cosas cambian. Así que su naturaleza es resistirse al cambio. Para los ágiles, no obstante, el cambio es bienvenido. Intentan ser procesos que se adaptan y crecen en el cambio, incluso al punto de cambiarse ellos mismos.

Los métodos ágiles son orientados a los usuarios y no orientados al proceso. La meta de los métodos pesados es definir un proceso que funcionara bien con cualquier usuario que lo use. Al contrario de los ágiles, ya que afirman que ningún proceso podrá nunca maquillar las habilidades del equipo de desarrollo, de modo que el papel del proceso es apoyar al equipo de desarrollo en su trabajo. Explícitamente puntualizan el trabajar a favor de la naturaleza humana en lugar de en su contra y enfatizan que el desarrollo de software debe ser una actividad agradable.

En consecuencia, a continuación se va a presentar un método de desarrollo de software ágil, por los beneficios que brinda al perfil de usuario del sistema ya que se centra en satisfacer sus necesidades.

3.2.1. AgilUS

Es un método de desarrollo ágil ya que se caracteriza por ser interactivo e incremental, y fácilmente adaptable a los cambios, involucrando al usuario en el desarrollo (*Acosta, 2012*).

AgilUS es un método de desarrollo ágil, desarrollado en el Centro de Ingeniería de Software y Sistemas (ISYS) de la Escuela de Computación, Universidad Central de Venezuela.

Se basa en el concepto de usabilidad, en la necesidad de desarrollar software usables. Se fundamenta en el análisis centrado en el usuario y en la participación de especialistas, con el objetivo de evolucionar el software, a fin de que este alcance el mayor grado de usabilidad una vez culminado su desarrollo. AgilUS es un método iterativo e incremental que pone el mayor peso del desarrollo en la consecución de la usabilidad. Se centra en que la construcción y desarrollo de las interfaces de usuario no debe ser una adición estética que se da al final del desarrollo del sistema sino, muy por el contrario, se debe guiar bajo las decisiones en Ingeniería de Software. En AgilUs son los usuarios, y no el cliente ni los programadores quienes guían el desarrollo del proyecto.

Las fases de AgilUs hacen énfasis en la importancia del usuario y sus evaluaciones. Está basado en el desarrollo iterativo e incremental de prototipos de alta fidelidad hasta que se convierten en el producto final para entrega.

La Ilustración N° 3.1 tomada de (*Acosta, 2012*) muestra un diagrama de la relación entre cada una de las etapas del ciclo de vida de AgilUs, con las actividades que se realizan y artefactos que se generan en cada etapa. A continuación se describen las etapas de este método:

- **Requisitos:** se realiza el 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 en el desarrollo del software, 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 modelo 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 la 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 evaluaciones de usabilidad apropiadas a esta etapa: las evaluaciones heurísticas y las listas de comprobación.
- **Entrega:** se aplican las pruebas al sistema para certificar que la aplicación desarrollada sea un software usable y sin errores, finalmente se pone en producción la aplicación.

En cada etapa del desarrollo se incluyen actividades para la construcción de la usabilidad. Se busca proporcionar una manera de proceder de manera organizada para construir la usabilidad durante el desarrollo de un producto. El ciclo de vida engloba la definición de requisitos, análisis, prototipaje y entrega.



Ilustración N° 3.1 Etapas del método AgilUS.

Cada etapa posee las siguientes tareas o actividades:

- **Requisitos:**
 - **Tormenta de ideas:** consiste en la generación de ideas por parte de un grupo multidisciplinario, liberando la mente para aceptar cualquier idea que se proponga, permitiendo, así, la libertad para la creatividad. En el caso del proceso de desarrollo de software, corresponde al equipo de desarrollo.
 - **Encuestas:** consisten en entrevistas con los usuarios en las que se formula un conjunto de preguntas y se registra las respuestas.
 - **Entrevistas:** es una técnica de indagación común, en la cual un entrevistador le hace preguntas a un experto en algún dominio, a fin de adquirir conocimientos en ese dominio.

- Cuestionarios: es menos flexible que la entrevista, pero puede llegar a un grupo más numeroso y se puede analizar con más rigor. Son listas escritas de preguntas que se distribuyen entre los usuarios.
- Evaluación de sistemas existentes: consiste en la revisión de versiones anteriores del mismo sistema, así como sistemas de la competencia o afines, con el objetivo de identificar problemas de usabilidad.
- Perfil de usuario: describe las competencias y habilidades de los usuarios; así como los conocimientos y frecuencia de uso de la tecnología. Se debe tomar en cuenta las características especiales y relevantes de los usuarios para el diseño de la interacción con el sistema.
- Requerimientos funcionales: describen la funcionalidad o los servicios que se espera que provea el sistema. Estos dependen del tipo de software y del sistema que se desarrolle y de los usuarios.
- Requerimientos no funcionales: son aquellos que no se refieren directamente a las funciones específicas que provee el sistema, sino a las propiedades emergentes de este como la usabilidad, seguridad o fiabilidad.
- Análisis:
 - Prototipos en papel: esta técnica se caracteriza por el uso de materiales y equipos sencillos para crear una simulación basada en papel de la interfaz de un sistema con el objetivo de explorar los requerimientos de usuario. El resultado obtenido se denomina frecuentemente prototipo de baja fidelidad.
 - Guías de estilo: son realizadas para el diseño de interfaces de usuario, resumen buenas prácticas, provee guías de bajo y alto nivel en el diseño de interfaces usables. La adopción de guías de estilo específicas se puede establecer como parte de los requerimientos no funcionales.
 - Modelo Casos de Uso: describe la secuencia de las interacciones que se desarrollarán entre los actores y el sistema, en respuesta a un evento que inicia un actor.
 - Modelo Objetos del Dominio: describe los objetos que existen en el dominio de la aplicación y las relaciones estáticas entre estos.
 - Objetos de interfaz: señala los objetos que se deben presentar en la interfaz de la aplicación.

- Patrones de interacción: son patrones que describen aspectos concernientes a la interfaz de usuario; están orientados a presentar soluciones apropiadas a problemas recurrentes que se les presentan a los usuarios cuando utilizan las aplicaciones interactivas. Deben estar centrados en los usuarios, por lo que deben proveer soluciones que garanticen interfaces usables.
- Prototipaje:
 - Prototipaje rápido: esta técnica está asociada a la idea de desarrollar diferentes conceptos propuestos mediante prototipos de software o hardware, para su posterior evaluación. El desarrollo de la simulación o prototipo del sistema futuro puede ser de gran ayuda, permitiendo a los usuarios visualizar el sistema (su concepto) e informar sobre el mismo, pudiéndose utilizar para aclarar opciones sobre los requerimientos de usuario y para especificar detalles de la interfaz de usuario a incluir en el sistema futuro.
 - Evaluación heurística: es una técnica de inspección, que normalmente la realiza un grupo reducido de personas evaluadores que, con base en su propia experiencia y fundamentándose en principios de usabilidad (heurísticas), evalúan de forma independiente la interfaz de usuario, contrastando finalmente los resultados con el resto de evaluadores.
 - Listas de comprobación: son recomendaciones que el equipo de desarrollo ha acordado considerar en el diseño de la interfaz de usuario. Es posible utilizar las heurísticas de Nielsen o en algunos casos adaptar los aspectos a los que se enfrenta el usuario del producto en desarrollo.
 - Pensamiento en voz alta: en esta técnica de evaluación conocida como “pensando en voz alta” (*thinking aloud*) descrita por Nielsen se les pide a los usuarios que expresen en voz alta sus pensamientos, sentimientos y opiniones mientras que interaccionan con el sistema
 - Prototipo ejecutable: es una versión más real y detallada de la interfaz o prototipo de la aplicación.
- Entrega:
 - Protocolo de preguntas: este método provoca ver las manifestaciones del usuario respecto del producto mediante la formulación de preguntas directas acerca del mismo.
 - Pruebas de aceptación: se utilizan para verificar que el sistema satisface los requerimientos funcionales y no funcionales. Estas pruebas las realiza el cliente. En general

son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario.

Todas estas técnicas pueden ser más productivas al reusar artefactos previos para incrementar el tiempo de respuesta, bajar los costos y aumentar la calidad del producto final. En nuestro caso, se va a reusar una aplicación Web llamada aTesT que gestiona evaluaciones para la Escuela de Idiomas Modernos.

3.3 Reusabilidad

La reusabilidad es el proceso de implementar o actualizar sistemas de software usando artefactos de software existentes. El tipo más común de reuso es de componentes de software. Sin embargo, existen otros artefactos producidos durante el proceso de desarrollo de software que también pueden ser reutilizados, como arquitecturas de software, modelos de análisis, modelos de diseño, patrones de diseño, esquemas de base de datos, servicios web, etc. Asimismo, el reuso de software se puede dar a través de varios sistemas similares e incluso diferentes dominios de aplicación (*Reusabilidad, 2013*).

La Fairleigh Dickinson University se refiere a la reusabilidad como la capacidad de los componentes de software de ser usados repetidamente en aplicaciones distintas para las que fueron originalmente desarrollados. Para poder ser efectivamente usado, el componente puede haber sido adaptado a los requerimientos de la nueva aplicación.

Por lo tanto, se puede decir que el reuso de software se puede definir como el proceso de aprovechar, utilizar componentes existentes y adaptarlos a las necesidades y requerimientos de la nueva aplicación o proyecto. Entonces, el propósito de reuso es evitar duplicar esfuerzo y por lo tanto reducir tiempo, costos y aumentar la calidad del producto software.

3.3.1 Beneficios de la reusabilidad

- La reusabilidad en un software tiene un impacto positivo en su calidad así como en la reducción de costos y un incremento en la productividad.
- Mejoras en la calidad: el reuso de software da como resultado mejoras en la calidad y confiabilidad.

- Calidad: al asegurar la corrección de errores reportados en un producto, ya que la próxima vez que se reutilice el producto en cuestión se hará con una menor cantidad de errores.
- Confiabilidad: los procesos de revisión y pruebas por los que pasan los productos que permiten incrementar la confiabilidad de uso de los mismos en los sistemas. Además, el uso de estos productos en diversos sistemas o proyectos permite incrementar la posibilidad de encontrar errores.
- Reducción de esfuerzo: el reúso de software disminuye el trabajo redundante, aumenta la productividad, reduce el mantenimiento y el tamaño de los equipos de trabajo.
- Trabajo redundante y tiempo de desarrollo: el desarrollo de un sistema desde “cero” implica invertir tiempo en elaborar elementos como especificaciones de requerimientos, especificaciones de casos de uso, etc. Esto puede ser evitado si se logra poner a disposición de los usuarios estos productos de trabajo que han sido desarrollados antes y esto se evidencia en una reducción de tiempo y costos.
- Productividad: el ahorro de tiempo y esfuerzo generado por reutilizar un producto permite a los usuarios incrementar su productividad. Es decir, los usuarios pueden enfocarse en otras actividades.
- Reducción de mantenimiento: debido a que los productos poseen un alto grado de calidad, los costos asociados al mantenimiento correctivo se reducen.
- Tamaño del equipo: en algunas ocasiones la sobrecarga de comunicación es un problema en equipos de desarrollo relativamente grandes. También, puede existir la creencia de que al aumentar el número de participantes de un proyecto se ha de generar un incremento en la productividad. Al reutilizar productos se puede reducir el tamaño de los equipos de desarrollo y así obtener mejoras en la comunicación e incrementos en la productividad.
- Reducción de costos: el reúso de software permite reducir los costos de los proyectos. El ahorro de costos se da al momento de reutilizar elementos existentes. Por lo tanto, mientras más frecuente sea el reúso, los costos menguarán notoriamente. Se debe tener en cuenta que la inversión que realiza un productor de un elemento reutilizable debe ser retribuido por el uso del mismo por parte de los usuarios.

Ahora bien, el modelo propuesto en esta investigación para la automatización del proceso de inscripción se expresa en términos de patrones de procesos. Estas soluciones fomentan y facilitan el reúso de soluciones exitosas. En consecuencia, se reducen los tiempos de desarrollo y se

aprovecha el conocimiento plasmado en las soluciones provistas por los patrones de proceso.

3.4 Patrón de proceso

En términos generales, un patrón es un mecanismo que permite expresar una solución exitosa a un problema recurrente en un contexto dado. En este Trabajo Especial de Grado un patrón de proceso se define como un modelo de referencia que describe una solución exitosa en términos de la estructura y funcionamiento de un problema concerniente a un tipo de proceso en un contexto específico (Zambrano, 2013).

Por lo tanto, los patrones de proceso generalizan soluciones exitosas para la automatización de los procesos de inscripción, en el dominio correspondiente a la gestión académica administrativa.

3.4.1 Metapatrón de procesos

En esta investigación se define el patrón de proceso que tiene la estructura que se describe a través del metapatrón de la Tabla 3.1. Los campos obligatorios de los patrones de proceso son: nombre, abstracción, problema, contexto, solución, rol y consecuencia. Los campos: política, planificación, tecnologías y método de desarrollo son opcionales ya que se pueden definirse en el patrón de más alto nivel de abstracción y aplica implícitamente en los patrones derivados. En el caso del campo patrones relacionados sólo se omite si el nivel de abstracción del patrón es de tipo actividad, por ser un patrón concreto, es decir, que contiene el código que implementa la solución propuesta.

Nombre	Nombre con el cual se identificará este patrón, debe expresar la funcionalidad principal del proceso
Abstracción	Corresponde al nivel de abstracción, puede ser: proceso se refiere a un patrón con mayor nivel de abstracción; subproceso se refieren a un patrón con nivel de abstracción intermedia, lo que quiere decir que está asociado a un patrón de proceso y a su vez tiene patrones relacionados; actividad se refiere a un patrón concreto que implementa la solución
Problema	Descripción del proceso a automatizar
Contexto	Condiciones que debe cumplir el entorno para resolver el problema
Solución	Descripción de la solución que automatiza el proceso
Rol	Descripción del(os) actor(es) involucrado(s) en el proceso
Política	Definición de condiciones o reglas que se debe tomar en cuenta en la automatización del proceso

Planificación	Organización de tiempo y recurso humano que se cataloga en términos de su experiencia en: avanzada, media e inicial y según su rol dentro del grupo de desarrollo de software: Líder de diseño, líder técnico y programador. Cuando el mismo equipo de desarrollo participa en todas las actividades se colocará su descripción únicamente en el patrón de proceso y éste se omitirá en el resto de los patrones que conforman el catálogo.
Consecuencia	Descripción de la salida asociada al proceso
Tecnologías	Descripción de tecnologías recomendadas para el desarrollo de la automatización
Método de desarrollo	Método de desarrollo de software sugerido

Tabla N° 3.1 Metapatrón de Proceso.

En este trabajo, el catálogo de patrones de proceso y sus relaciones, asociado a la inscripción consiste en ocho (8) patrones que se pueden apreciar según su nivel de abstracción en la siguiente Ilustración:

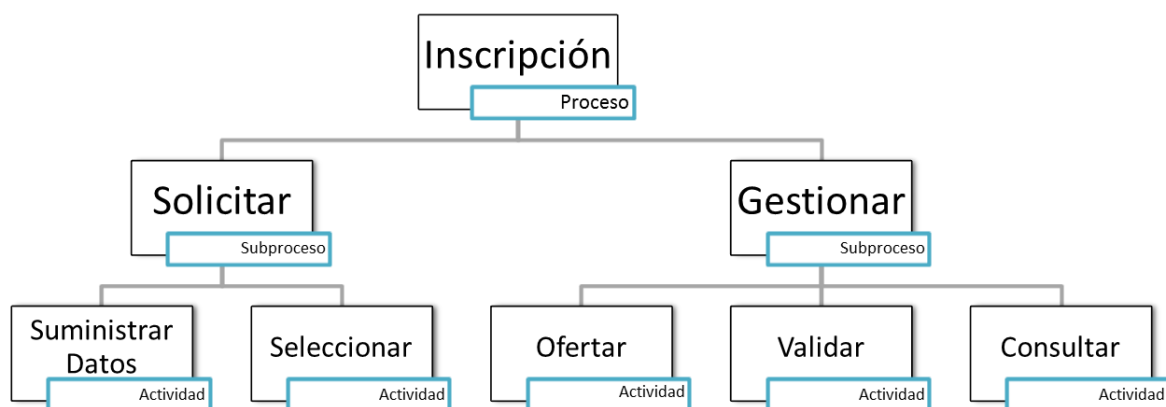


Ilustración N° 3.2 Catálogo de Patrones de Proceso.

A continuación se presenta la instanciación del catálogo para el sistema de la gestión de la evaluación instrumental.

3.4.2 Patrones de proceso de la evaluación instrumental

Los patrones de proceso para la evaluación instrumental permiten expresar los procesos y subprocesos de la problemática. A continuación, se describe el catálogo que se obtiene de esta investigación de los patrones de proceso de inscripción de la evaluación instrumental (ver Tabla N° 3.2). Dicha instanciación forma parte de la fase de análisis de AgilUS.

Nombre	Inscripción de la Evaluación Instrumental
Abstracción	Nivel proceso
Problema	La Coordinación de Extensión de la Escuela de Idiomas Modernos es la encargada de realizar la inscripción tanto del curso como de la prueba instrumental, en donde los aspirantes son estudiantes de Postgrado de la Universidad Central de Venezuela que deben aplicar la prueba instrumental de inglés. Este proceso genera muchas desventajas administrativas, de costos y tiempo de respuesta, por lo que requieren realizarlo de forma automatizada
Contexto	Se lleva a cabo manualmente
Solución	Desarrollar un módulo Web en el sistema aTeST que permita llevar a cabo el proceso de gestión para la inscripción de la evaluación instrumental para estudiantes de Postgrado de la Universidad Central de Venezuela
Rol	La Coordinación de Extensión de la Escuela de Idiomas Modernos está conformada por miembros administrativos y directivos de la Escuela, éstos se encargan de llevar a cabo todas las consideraciones, políticas y calificación que requiere la gestión para la evaluación instrumental. Los aspirantes que desean realizar o presentar el curso o examen instrumental son los interesados a un Postgrado de la Universidad Central de Venezuela
Política	Los aspirantes deben llamar por teléfono a las instalaciones, y posteriormente se realiza el Proceso de Registro de los datos personales tales como nombre, cédula, número de teléfono, área de estudio de postgrado que va a realizar el examen y el Idioma en el que lo va a presentar. De igual forma, se le indica el costo de dicho examen, el número de cuenta a depositar, la fecha del próximo examen y se le pide confirmación para la fecha que desee presentar
Planificación	Una semana Grupo de desarrollo conformado por: personal administrativo y directivo
Tecnologías	Linux Debian, HTML5, <i>jQuery</i> , CSS, Ajax, MySQL y Ruby on Rails
Método de desarrollo	AgilUs
Consecuencia	Se registran y gestionan a los aspirantes en el proceso de inscripción
Patrones Relacionados	Solicitar y Administrar

Tabla N° 3.2 Patrón de proceso de gestión para la inscripción de la Evaluación Instrumental.

En la Tabla N° 3.3 se describe el patrón que corresponde al subproceso de solicitud de la inscripción.

Nombre	Solicitar inscripción
Abstracción	Nivel subproceso
Problema	La Coordinación de Extensión de la Escuela de Idiomas Modernos requiere realizar el registro de la solicitud de la inscripción, para ello se requieren los datos personales, de contacto, la opción que desea presentar (examen o curso instrumental) y el postgrado
Contexto	Se lleva a cabo a través de una hoja de papel con los campos requeridos para el registro
Solución	Desarrollar un formulario en la aplicación Web que permita el registro de los aspirantes con sus respectivas opciones
Rol	Aspirante y personal administrativo
Política	El personal administrativo realiza el Proceso de Registro de los datos personales tales como nombre, cédula, número de teléfono, área de estudio de postgrado que va a realizar el examen y el idioma en el que lo va a presentar (solo si es diferente al inglés y si está disponible para el área de estudio de postgrado).

Planificación	Una (1) semana
Consecuencia	El reporte debe contener la información personal, de contacto del solicitante y la fecha de la opción requerida
Patrones Relacionados	Suministrar datos y seleccionar

Tabla N° 3.3 Patrón de proceso Solicitar inscripción.

A continuación en la Tabla N° 3.4 se describe el patrón que corresponde a suministrar los datos personales, de contacto y administrativos de los aspirantes.

Nombre	Suministrar datos para la inscripción
Abstracción	Nivel actividad
Problema	Para la inscripción de la prueba o curso instrumental se requiere conocer de los datos personales tales como nombre, cédula, número de teléfono, área de estudio de postgrado que va a realizar el examen y el número de depósito bancario
Contexto	Se lleva a cabo a través de una hoja de papel con los campos requeridos para el registro
Solución	Desarrollar un formulario que permita el registro los datos personales, de contacto y administrativos de los aspirantes a la prueba o curso instrumental
Rol	Aspirante y personal administrativo
Política	El tiempo dispuesto para llevar a cabo el proceso de inscripción es de una (1) semana para el examen y aproximadamente dos (2) semanas para el curso
Planificación	Una (1) semana o dos (2) semanas
Consecuencia	Se registran los datos personales, de contacto y administrativos del aspirante

Tabla N° 3.4 Patrón de proceso Suministrar datos para la inscripción.

A continuación en la Tabla N° 3.5 se describe el patrón que permite al aspirante seleccionar los postgrados.

Nombre	Seleccionar tipo de examen
Abstracción	Nivel actividad
Problema	La Coordinación de Extensión de la Escuela de Idiomas Modernos requiere conocer la opción de postgrado que desea llevar a cabo el aspirante
Contexto	Se lleva a cabo a través de listas
Solución	Desarrollar un componente que permita el registro de la opción requerida por el aspirante
Rol	Aspirante
Política	El aspirante debe escoger el postgrado que desea realizar
Planificación	Una (1) semana
Consecuencia	Se registra la opción que requiera el aspirante

Tabla N° 3.5 Patrón de proceso Seleccionar tipo de examen.

A continuación en la Tabla N° 3.6 se describe el patrón que permite a la organización administrativa de la inscripción.

Nombre	Gestionar el curso preparatorio y examen instrumental
Abstracción	Nivel subproceso
Problema	La Coordinación de Extensión de la Escuela de Idiomas Modernos requiere ofertar exámenes, cursos, confirmar las solicitudes de los aspirantes, consultar y gestionar el proceso de inscripción
Contexto	Se lleva a cabo a través de funcionalidades manuales que permiten ofertar, gestionar y consultar
Solución	Desarrollar funcionalidades en un módulo de la aplicación Web aTeST que permitan ofertar los exámenes o cursos, gestionar las solicitudes de los aspirantes, calificación y reportes de los aspirantes inscritos distinguidos por opción de postgrado y fecha de presentación
Rol	Personal directivo de la Coordinación de Extensión de la Escuela de Idiomas Modernos
Política	Se requiere conocer la fecha y aspirantes para una descripción completa de cada una de los exámenes y cursos ofertados. Conocer el estado del proceso de inscripción actual y anteriores en función al número de inscritos, calificación y opción de postgrado que desean optar
Planificación	Dos (2) semanas
Consecuencia	La oferta contiene el cupo e identificación del curso o prueba y los espacios físicos disponibles para cada una de ellas. Las consultas se presentan filtradas por opción y agrupadas por fecha de presentación de examen o inicio de curso
Patrones Relacionados	Ofertar exámenes, Confirmar, Consultar, Calificar

Tabla N° 3.6 Patrón de proceso Gestionar el examen y curso instrumental.

En la Tabla N° 3.7 se describe el patrón que corresponde al registro de la prueba o del curso instrumental que se ofertarán en la inscripción.

Nombre	Ofertar exámenes y cursos
Abstracción	Nivel actividad
Problema	La Coordinación de Extensión de la Escuela de Idiomas Modernos requiere ofertar los exámenes o cursos y establecer las opciones válidas o excepciones de la selección, las cuales se detallan a continuación. - Prueba instrumental: se presentan los días sábados, sólo se podrá escoger la fecha activa para el día de solicitud. - Curso instrumental: depende de la disponibilidad del profesor y de la cantidad de aspirantes al mismo, sino cumple el mínimo de requisitos entra a una lista de espera
Contexto	Se lleva a cabo manualmente con sus respectivas reglas
Solución	Desarrollar funcionalidades en un módulo Web para ofertar exámenes o cursos y establecer reglas para cada uno. Para el examen instrumental sólo se puede ofertar en una fecha específica para un momento dado. Para el curso preparatorio, sólo la Coordinación de Extensión puede gestionar la inscripción para la fecha y área que ellos crean convenientes.
Rol	Personal directivo de la Coordinación de Extensión de la Escuela de Idiomas Modernos
Política	Se requiere conocer el identificador, el nombre y la fecha; para esta actividad se estima un tiempo aproximado de una (1) semana
Planificación	Una (1) semana
Consecuencia	Se registran los exámenes y cursos con sus respectivas reglas

Tabla N° 3.7 Patrón de proceso Ofertar exámenes y cursos.

En la Tabla N° 3.8 se describe el patrón que corresponde a la confirmación de los aspirantes en la inscripción.

Nombre	Confirmar
Abstracción	Nivel actividad
Problema	La Coordinación de Extensión de la Escuela de Idiomas Modernos requiere validar la solicitud con el objetivo de confirmar su participación en el proceso de inscripción según los requisitos exigidos y la disponibilidad, tales como: verificación del pago mediante el comprobante del depósito bancario y cédula de identidad
Contexto	Se lleva a cabo manualmente
Solución	Presentar un método para ingresar el número del depósito bancario para cada aspirante por el examen o curso instrumental
Rol	Aspirantes
Política	Para optimizar el uso de los recursos físicos (aulas) y humanos se requiere de la confirmación a fin de conocer cantidad real de los aspirantes.
Planificación	Una (1) semana
Consecuencia	La confirmación genera un comprobante con dos (2) copias, una (1) para la Coordinación de Extensión de la Escuela de Idiomas Modernos y otra (1) para el aspirante

Tabla N° 3.8 Patrón de proceso Confirmar.

En la Tabla N° 3.9 se describe el patrón que permite a la Organización consultar el estado de inscripción.

Nombre	Consultar
Abstracción	Nivel actividad
Problema	La Coordinación de Extensión de la Escuela de Idiomas Modernos hace un seguimiento sobre el estado del proceso de inscripción por lo que requiere saber la cantidad de aspirantes por examen agrupados por fecha, y también la cantidad de estudiantes por curso
Contexto	Se lleva a cabo a través de una tabla escrita en una hoja de papel
Solución	Desarrollar un módulo que permita consultar los estados de los aspirantes inscritos por postgrado para el curso y examen instrumental
Rol	Personal directivo y administrativo de la Coordinación de Extensión de la Escuela de Idiomas Modernos
Política	Consultar la cantidad de aspirantes por postgrado
Planificación	Una (1) Semana
Consecuencia	Se obtiene un reporte digital que indica la cantidad de aspirantes por postgrado agrupados por una fecha de examen o curso instrumental

Tabla N° 3.9 Patrón de proceso Consultar.

En la Tabla N° 3.10 se describe el patrón que permite gestionar noticias correspondientes al curso o examen instrumental.

Nombre	Gestionar noticias
Abstracción	Nivel actividad
Problema	La Coordinación de Extensión de la Escuela de Idiomas Modernos debe informar todas las características sobre la evaluación instrumental,
Contexto	Se lleva a cabo a través de llamadas telefónicas
Solución	Desarrollar una funcionalidad en el sistema que permita visualizar todas las noticias relacionadas a la disponibilidad, pago y fechas tanto del curso como examen instrumental
Rol	Personal directivo y administrativo de la Coordinación de Extensión de la Escuela de Idiomas Modernos

Política	Consultar toda la información requerida
Planificación	Una (1) Semana
Consecuencia	Cada aspirante recibe por correo su calificación, si aprobó puede descargar su constancia o certificado

Tabla N° 3.10 Patrón de proceso Gestionar noticias.

En la Tabla N° 3.11 se describe el patrón que permite calificar a cada aspirante que presentó el curso o el examen instrumental.

Nombre	Calificar
Abstracción	Nivel sub - proceso
Problema	La Coordinación de Extensión de la Escuela de Idiomas Modernos debe asignarle una calificación al examen o curso instrumental de cada aspirante. Los posibles valores son: aprobado, reprobado, no presentó o falta de comprobante del depósito bancario
Contexto	Se lleva a cabo a través de una tabla escrita en una hoja de papel
Solución	Desarrollar una funcionalidad en el sistema que permita asignarle una calificación a cada evaluación y curso para cada aspirante inscrito
Rol	Personal directivo y administrativo de la Coordinación de Extensión de la Escuela de Idiomas Modernos
Política	Consultar la cantidad de aspirantes por postgrado
Planificación	Una (1) Semana
Consecuencia	Cada aspirante recibe por correo su calificación, si aprobó puede descargar su constancia o certificado

Tabla 3.11 Patrón de proceso de Calificar.

En la Tabla N° 3.12 se describe el patrón que permite enviar por correo la calificación obtenida para cada aspirante que presentó el curso o el examen instrumental.

Nombre	Enviar correos
Abstracción	Actividad
Problema	La Coordinación de Extensión de la Escuela de Idiomas Modernos debe llamar por teléfono o enviar correos electrónicos a cada participante al momento de conocer su calificación obtenida
Contexto	Se lleva a cabo a través un teléfono o correos electrónicos
Solución	Desarrollar una funcionalidad en el sistema que permita enviar correos electrónicos automáticamente al momento de asignarle una calificación al aspirante
Rol	Personal directivo y administrativo de la Coordinación de Extensión de la Escuela de Idiomas Modernos
Política	Notificar calificación
Planificación	Una (1) Semana
Consecuencia	Cada aspirante recibe por correo su calificación

Tabla 3.12 Patrón de proceso Enviar correos.

En la Tabla N° 3.13 se describe el patrón que permite descargar la constancia o certificado para cada aspirante que aprobó el curso o el examen instrumental.

Nombre	Descargar reportes
Abstracción	Actividad
Problema	La Coordinación de Extensión de la Escuela de Idiomas Modernos debe imprimir cada constancia o certificado para cada aspirante que haya aprobado el curso o examen instrumental, firmarlo y colocarle el sello correspondiente
Contexto	Se lleva a cabo de forma manual la impresión de dichos documentos
Solución	Desarrollar una funcionalidad en el sistema que permita descargar por la aplicación Web, los diferentes documentos de constancias o certificados con las firmas digitales, y sólo llevar a sellar en la Institución
Rol	Aspirantes a optar por un Postgrado en la UCV
Política	Descargar reportes de constancia o certificados
Planificación	Una (1) Semana
Consecuencia	Cada aspirante podrá descargar su constancia o certificado luego de ser aprobado el examen o curso instrumental

Tabla 3.13 Patrón de proceso Descargar reportes

Luego de la instanciación de cada patrón de proceso para la inscripción de la evaluación y curso instrumental, se puede representar gráficamente el catálogo descrito en la Ilustración 3.3:



Ilustración N° 3.3 Catálogo de patrón de proceso para la inscripción del curso y evaluación instrumental.

En la siguiente sección se representa una descripción ilustrativa de la aplicación del método de desarrollo de software AgilUs, el cual se aplicó en este Trabajo.

CAPÍTULO IV DESARROLLO DE SOFTWARE

Una vez que se describe el proceso de inscripción de la evaluación instrumental y se realizan las especificaciones sobre AgilUS, el cual es un método de desarrollo iterativo e incremental, se planifican y se llevan a cabo cuatro (4) iteraciones (requisitos – análisis – prototipaje - entrega), la primera llamada Análisis Global del Sistema donde sólo se encuentran las fases de requisitos y análisis, posteriormente la iteración Inscripción del curso y examen instrumental donde se encuentra el desarrollo de un módulo para esta funcionalidad, luego Gestión Administrativa donde se lleva a cabo todas las tareas por parte del administrador del sistema, y por último, la Generación de reportes constituidos por constancias o certificados. Estas tres últimas iteraciones sólo serán constituidas por las fases de Prototipaje y Entrega. Adicionalmente, se realizó una lista de chequeo para conocer el grado de satisfacción de los usuarios.

Esta distribución se puede llevar a cabo gracias a que AgilUs permite iterar dentro de cada etapa tantas veces como sea necesario, hasta conseguir el producto deseado.

4.1 Iteración 1 - Análisis Global

Constituye la primera etapa del método, en la cual se establecen los requisitos y análisis en base a las necesidades del usuario. En la etapa de requisitos se aplicaron diversas técnicas de indagación para obtener la información necesaria para un posterior análisis de requerimientos planteados por el personal directivo y administrativo de la Coordinación de Extensión de la Escuela de Idiomas Modernos para el desarrollo de las funcionalidades. Las técnicas empleadas han sido: tormenta de ideas, perfil de usuario, listado de requerimientos funcionales y no funcionales.

4.1.1 Tormenta de ideas

Esta técnica se realizó en conjunto con el personal directivo de la Coordinación de Extensión de la Escuela de Idiomas Modernos en el mes de enero del 2013.

- La Coordinación de Extensión de la Escuela de Idiomas Modernos es la encargada de toda la gestión de la evaluación instrumental.
- Sobre la inscripción actual de la evaluación instrumental: llaman los interesados y se le da el número de cuenta y el monto a depositar, se le indican que pueden depositar de lunes a jueves y deben llamar para confirmar su asistencia el sábado. Cuando llaman deben dar nombre, número

de comprobante de depósito, área de postgrado e idioma a ser presentado. Pueden escoger su idioma siempre y cuando esté disponible en ese postgrado. La persona que atiende las llamadas les confirma la información de la hora y el lugar a presentar. Se les indica que pueden traer diccionario, un documento de identidad y el comprobante de depósito para consignarlo el día del examen. El martes siguiente se le da el resultado de la prueba mediante una llamada telefónica.

- Sobre corrección del examen: el profesor del idioma correspondiente corrige manualmente los exámenes y en caso de dudas se convocan a otras personas para que sirvan como jurados y evaluar el examen. Cuando se tiene las notas estas se escriben manualmente en una hoja y se le va informando a cada persona que vaya llamando.
- Instrumental quiere decir que solo se obtiene la comprensión escrita.
- El profesor Lucius hace las constancias de la aprobación de los exámenes con una plantilla utilizando herramientas ofimáticas y escribiendo manualmente los datos de cada persona.
- Las constancias de estudio se crean igual que las constancias de aprobación del examen.
- Se reciben aproximadamente 80 llamadas por día.
- Se pierden documentos donde se anotan datos de personas.
- Existen días que no hay nadie en la oficina para atender.
- Poseen altos costos de teléfono.
- Mucho dinero gastado en reproducción de exámenes y constancias.
- En cuanto al curso instrumental se reciben llamadas para anotarse en lista de espera. Luego de esto, se avisa por mensaje de texto la información acerca del curso y los datos para realizar la inscripción del mismo.

4.1.2 Requerimientos funcionales

En base a la observación realizada en el desarrollo de la gestión de la evaluación instrumental y las reuniones con el personal administrativo y directivo de la Coordinación de Extensión de la Escuela de Idiomas Modernos, se recopilaban diversos requerimientos considerados al momento del desarrollo de las funcionalidades de gestión en la aplicación web. A continuación se detallan cada uno de estos requerimientos.

Nombre	Visualizar noticias relacionadas a la evaluación instrumental
Identificación del requerimiento	Visualizar noticias
Características	Los usuarios interesados en la evaluación instrumental (estudiantes de postgrado de la UCV), podrán visualizar toda la información o noticias relacionadas con este proceso, como la fecha del examen instrumental activo para esa semana y la información de contacto para conocer los detalles del curso preparatorio
La prioridad es: Alta	

Tabla N° 4.1 Requerimiento para Visualizar noticias.

Nombre	Administrar fecha del examen instrumental
Identificación del requerimiento	Agregar, modificar, eliminar y consultar la fecha para el examen instrumental
Características	El administrador podrá agregar, modificar, eliminar o consultar la fecha para el examen instrumental activo o alguna anterior
La prioridad es: Alta	

Tabla N° 4.2 Requerimiento para Administrar el examen instrumental.

Nombre	Inscribir curso o prueba instrumental
Identificación del requerimiento	Inscripciones
Características	Desarrollar un módulo que permita las inscripciones tanto del curso como de la evaluación instrumental para los estudiantes de postgrado de la UCV
La prioridad es: Alta	

Tabla N° 4.3 Requerimiento para realizar las Inscripciones del curso o prueba instrumental.

Nombre	Mecanismo que maneje el envío de correo electrónico
Identificación del requerimiento	Mensajes de correo electrónico
Características	Implementar un mecanismo que maneje el envío de correo electrónico, para informar detalles sobre el estado de distintos procesos involucrados en la evaluación instrumental
La prioridad es: Media	

Tabla N° 4.4 Requerimiento para tener un Mecanismo que maneje correos electrónicos.

Nombre	Generar constancias y certificados
Identificación del requerimiento	Módulo para la generación de constancias y certificados

Características	Implementar un módulo para la generación de constancias (si realizó el curso o aprobó el examen instrumental) y certificados sólo si realizó el curso, en formato PDF con los datos necesarios
<i>La prioridad es: Media</i>	

Tabla N° 4.5 Requerimiento para generar constancias y certificados.

Nombre	Administrar inscripción del curso y evaluación instrumental
Identificación del requerimiento	Módulo para agregar, consultar, modificar o eliminar especificaciones sobre la inscripción
Características	Implementar un módulo donde el administrador o profesor pueda agregar, consultar, modificar o eliminar especificaciones sobre la inscripción tanto del curso como de la prueba instrumental
<i>La prioridad es: Alta</i>	

Tabla N° 4.6 Requerimiento para Administrar la inscripción del curso y evaluación instrumental.

Nombre	Almacenar exámenes
Identificación del requerimiento	Módulo para almacenar los exámenes para los distintos postgrados
Características	Implementar un módulo donde el administrador o profesor pueda almacenar los exámenes según el postgrado que corresponda
<i>La prioridad es: Media</i>	

Tabla N° 4.7 Requerimiento para Almacenar exámenes.

4.1.3 Requerimientos no funcionales

Son aquellos requerimientos que no se refieren directamente a las funciones específicas que ofrece el sistema, sino a propiedades importantes emergentes que brindará como la usabilidad, confiabilidad, seguridad, mantenimiento y restricción del diseño. Los cuales generan la satisfacción al usuario.

4.1.4 Perfil de usuario

Esta aplicación Web está orientada a dos tipos de usuarios los cuales asumen un rol dependiendo de las responsabilidades que desempeñan dentro de la gestión de la evaluación instrumental. Los cuales se describen a continuación:

- El primer usuario es el administrador que será llevado a cabo por el personal de la Coordinación de Extensión de la Escuela de Idiomas Modernos, como el Director, Coordinador,

entre otros. Los cuales están familiarizados con el proceso de evaluación y curso instrumental, y se le otorgarán suficientes permisos en la aplicación para permitirles manejar la gestión de los procesos mencionados anteriormente. Estos usuarios requieren conocimientos en el uso del computador para navegar en internet y ser autenticados correctamente en el sistema.

- El segundo usuario es el aspirante, que está conformado por todos los estudiantes de postgrado de la Universidad Central de Venezuela que desean realizar el curso o evaluación instrumental. Este grupo tendrá la función de utilizar la aplicación para consultar noticias, realizar inscripción, solicitar constancia o certificado y actividades básicas en la gestión de los procedimientos antes mencionados. Estos usuarios requieren conocimientos en el uso del computador para navegar en Internet.

Para la segunda etapa del método empleado, se lleva a cabo el análisis de los requerimientos obtenidos en la fase anterior. En ésta se definirán el modelo de casos de uso, el modelo de objetos del dominio y los patrones de interacción.

4.1.5 Modelo de casos de uso

Se diseña un diagrama de casos de uso para describir las funcionalidades de esta iteración y la relación existente con los usuarios del mismo.

Las siguientes ilustraciones demuestran los diferentes casos de uso generados en cada nivel, se puede apreciar el nivel 0 en la Ilustración N° 4.1 y en la Ilustración N° 4.2 se puede distinguir el nivel 1:

Nivel 0:

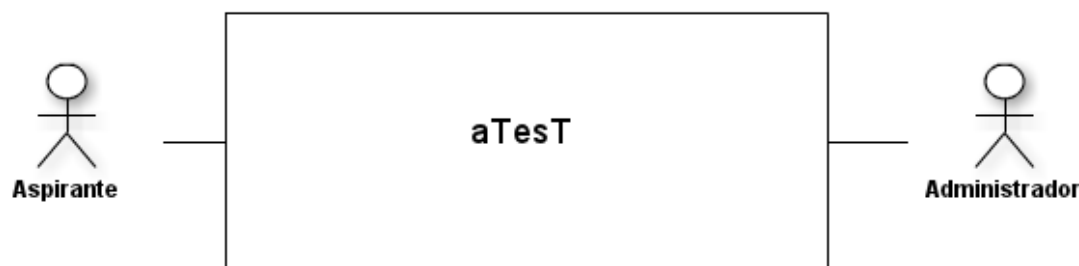


Ilustración N° 4.1 Nivel 0 del diagrama de Casos de Uso.

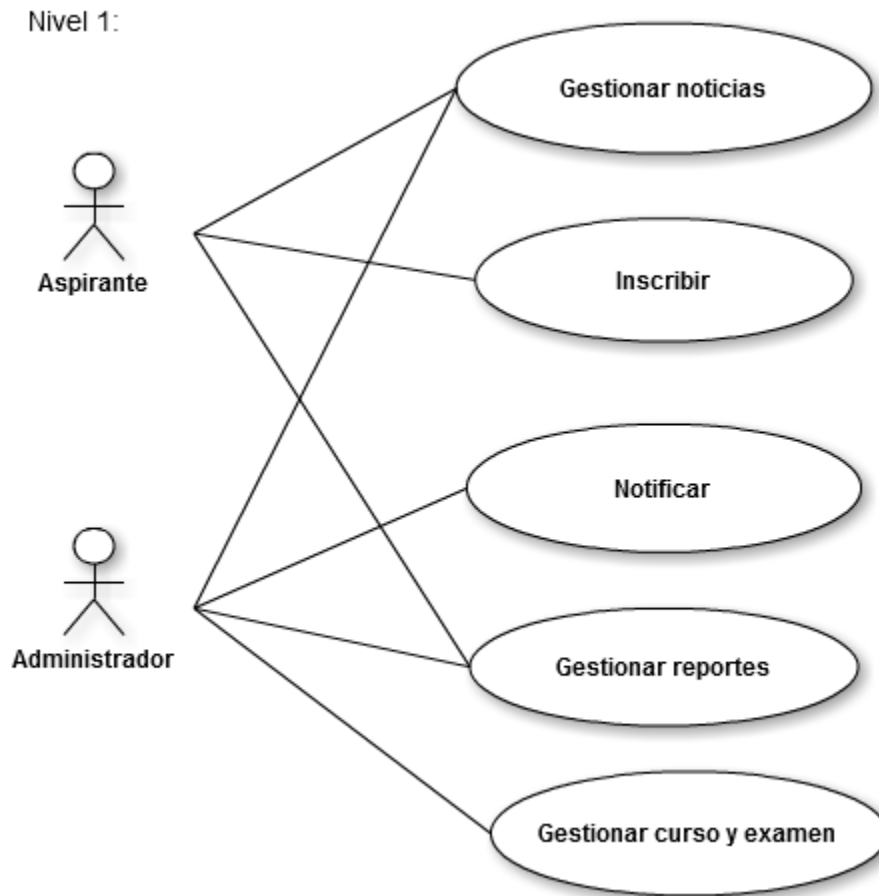


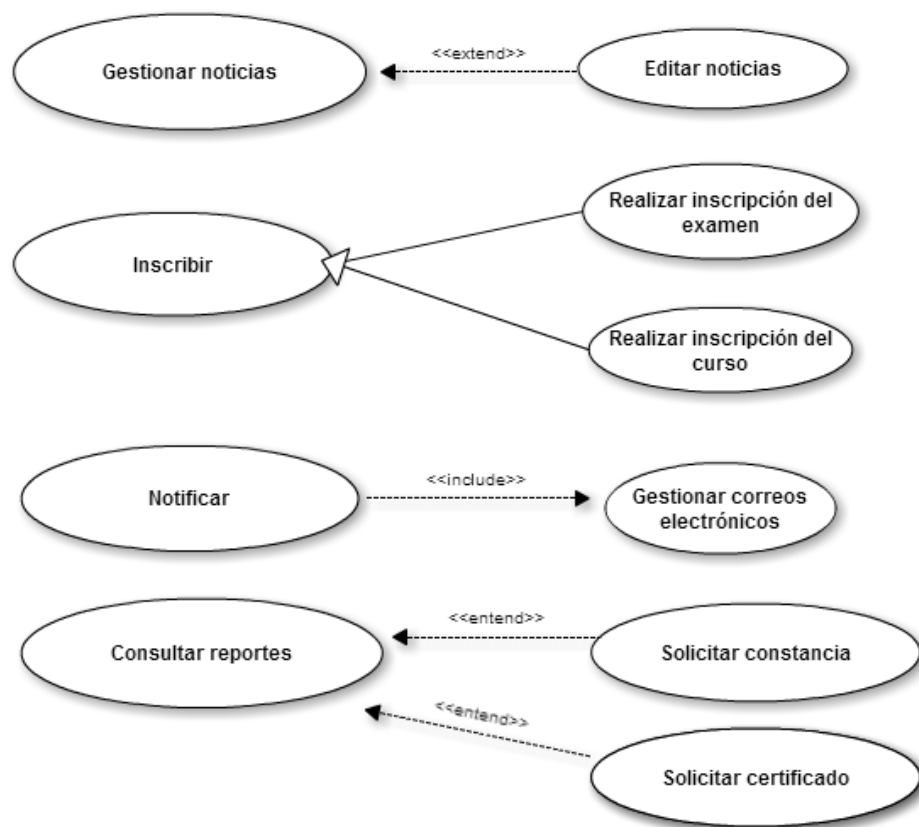
Ilustración N° 4.2 Nivel 1 del diagrama de Casos de Uso.

Se definen dos (2) actores principales, el aspirante, que representa a los estudiantes de postgrado de la Universidad Central de Venezuela interesados en presentar el curso o examen instrumental, y el administrador, que representa al personal administrativo y directivo de la Coordinación de Extensión. Dentro de los casos de uso se encuentran:

- Gestionar noticias: permite al aspirante visualizar todas las noticias e información relevante sobre el proceso de evaluación instrumental, y además brinda la posibilidad al administrador para editar las noticias.
- Inscribir: permite al aspirante realizar la inscripción para presentar el examen o curso instrumental, de manera que pueda notificar toda la información necesaria y al mismo tiempo conocer los detalles de dicha inscripción.

- Notificar: brinda la facilidad al administrador de gestionar los mensajes de texto o correos electrónicos para comunicar a los aspirantes información relevante.
- Gestionar reportes: le permite al aspirante que ha aprobado el curso o examen descargar la constancia o certificado.
- Gestionar curso y examen: administra todo el proceso de inscripción, posee actividades como agregar, modificar, consultar o eliminar inscripción.

A continuación se puede apreciar el nivel 2 en la Ilustración N° 4.3 y el nivel 3 en la Ilustración N° 4.4.



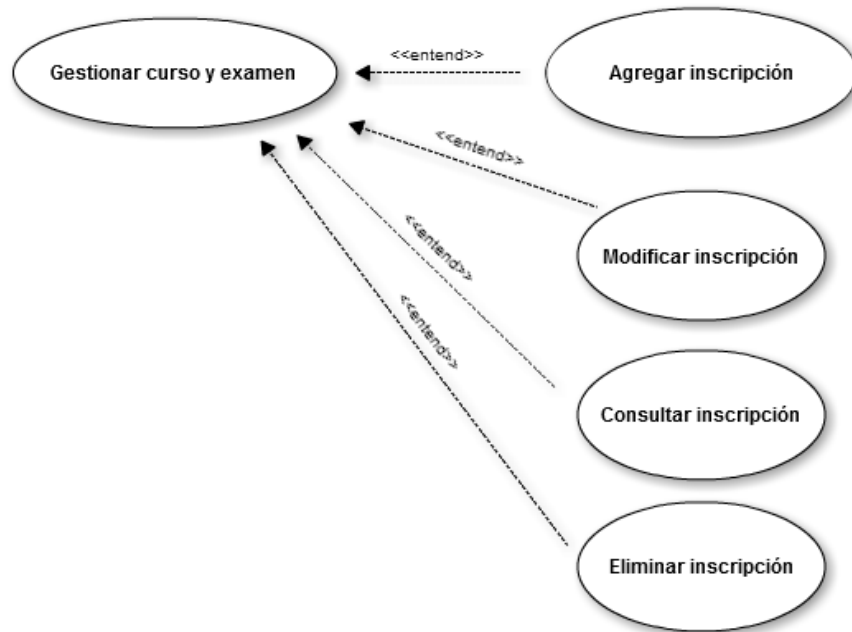


Ilustración N° 4.3 Nivel 2 del diagrama de Casos de uso.

Nivel 3:

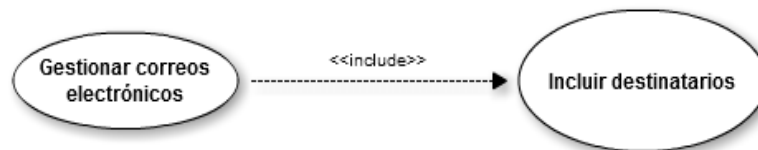


Ilustración N° 4.4 Nivel 3 del diagrama de Casos de Uso.

4.1.6 Modelo de objetos del dominio

A través de este modelo se puede definir el dominio que compone la gestión, a partir de los objetos que lo conforman y sus relaciones. Los objetos presentes en el modelo de objetos de dominio de gestión son evidenciados en la Ilustración N° 4.5:

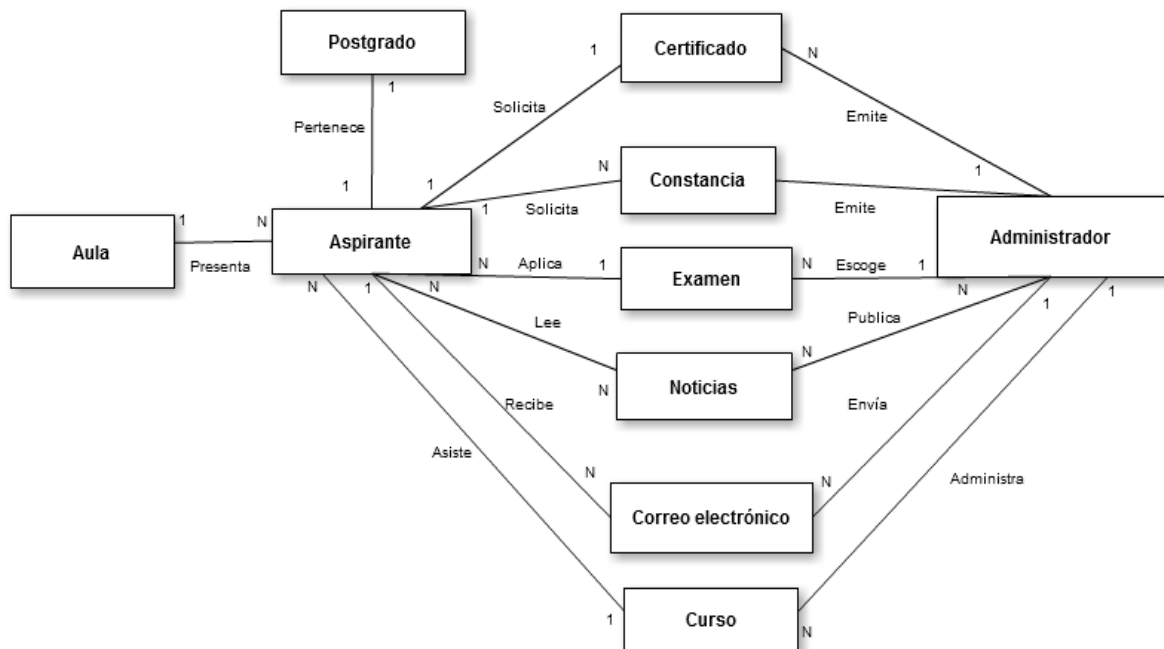


Ilustración N° 4.5 Modelo de objetos del dominio.

4.1.7 Patrones de interacción

Un patrón de interacción describe una solución exitosa a un problema recurrente concerniente a la interfaz de usuario, en un contexto dado. A continuación se describen cada uno de los patrones relacionados a la aplicación en cuestión:

- Patrón de dominio:

Nombre	Aplicación Web para gestión administrativa	
Descripción	Aplicación Web basada en el paradigma de Cliente-Servidor, con la finalidad de automatizar diferentes procesos para facilitar la gestión administrativa de una empresa en particular	

Tabla N° 4.8 Patrón de dominio.

- Patrón de usuario:

Nombre	Aspirante	
---------------	-----------	---

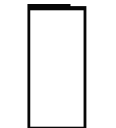
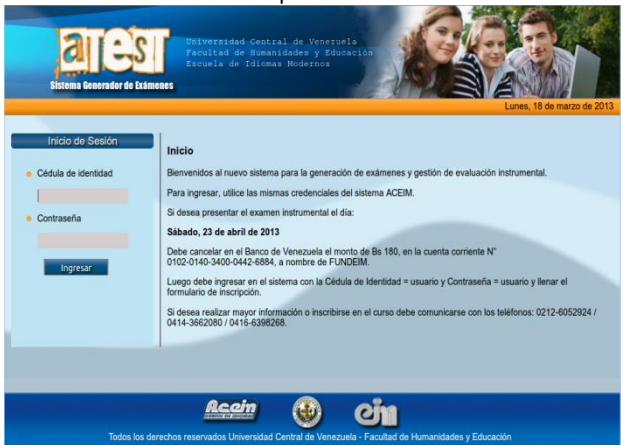
Descripción	Usuario aspirante a optar un postgrado ofrecido por la Universidad Central de Venezuela, y desea presentar o conocer información sobre el examen o curso instrumental
--------------------	---

Tabla N° 4.9 Patrón de usuario Aspirante.

Nombre	Administrador	
Descripción	Usuario administrador del sistema de gestión administrativa para procesos como inscripción, generación de constancias y certificados, entre otros. Como por ejemplo, el Director de la Escuela de Idiomas Modernos o algún Instructor	

Tabla N° 4.10 Patrón de usuario Administrador.

- Patrón de sistema:

Nombre, clasificación, confianza y autor	Módulo para una aplicación Web de gestión de los procesos correspondientes a la evaluación instrumental Sistema 0 Daniela Colmenares	
Problema	Estos procesos correspondientes a la evaluación instrumental se realizan en su mayoría de forma manual, lo que aumenta el flujo de errores y el tiempo de respuesta. Los costos son elevados gracias a impresiones, fotocopias y mensajes de texto	
Solución	Desarrollar una aplicación Web que automatice la inscripción del curso y prueba, las actividades logísticas de la presentación del examen y la generación de certificados y constancias de la evaluación instrumental que realiza la Escuela de Idiomas Modernos 	
Contexto	Esta evaluación y curso instrumental se aplica a estudiantes de postgrado (en todas las áreas) de la Universidad Central de Venezuela	
Fuerzas	Al ser una aplicación Web, está sujeta a los inconvenientes que pueda presentar la conexión a Internet o el Servidor	

Usabilidad	Satisfacción a todos los usuarios, fácil uso y acceso a la aplicación Web, tolerancia a fallos
Consecuencias	Automatizar diferentes procesos relacionados con la evaluación instrumental

Tabla N° 4.11 Patrón de sistema.

- Patrones de tareas:

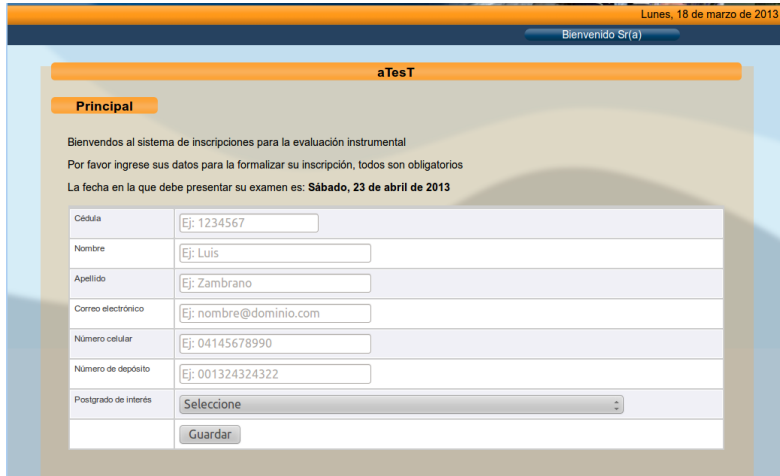
Nombre, clasificación, confianza y autor	Registro Tarea 0 Daniela Colmenares	
Problema	Los usuarios repetidamente necesitan introducir una gran cantidad de datos personales	
Solución	Ofrecer a los usuarios la posibilidad de almacenar su información personal para su uso posterior 	
Contexto	Los aspirantes podrán ingresar su información personal y administrativa para presentar el examen instrumental, y el administrador podrá ingresar la información de los aspirantes al curso	
Fuerzas	Puede introducir algún campo de forma incorrecta, pero se le notificará	
Usabilidad	Satisfacción a todos los usuarios, fácil uso y acceso a la aplicación Web, tolerancia a fallos	
Consecuencia	Almacenamiento de la información personal de cada aspirante	

Tabla N° 4.12 Patrón de tarea Registro.

Nombre, clasificación, confianza y autor	Formulario Tarea 0 Daniela Colmenares	
---	--	---

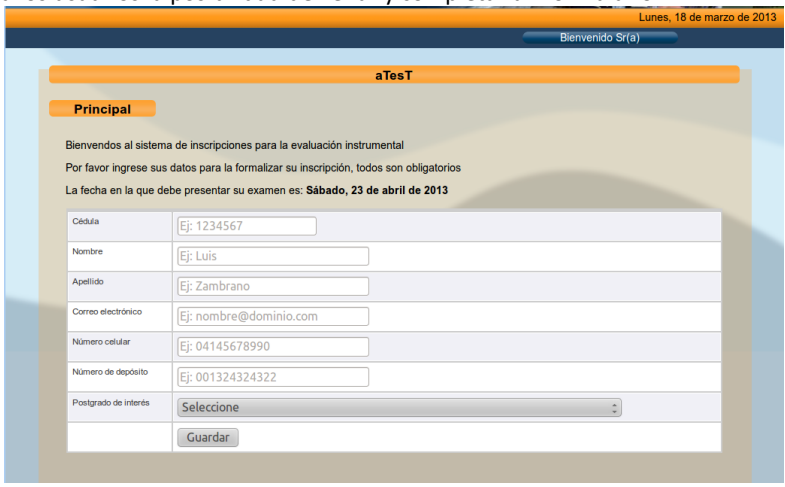
Problema	Los usuarios necesitan completar un formulario con diferentes opciones
Solución	Ofrecer a los usuarios la posibilidad de llenar y completar un formulario 
Contexto	Los aspirantes podrán ingresar toda la información necesaria en el formulario correspondiente
Fuerzas	Puede introducir algún campo de forma incorrecta, pero se le notificará
Usabilidad	Satisfacción a todos los usuarios, fácil uso y acceso a la aplicación Web, tolerancia a fallos
Consecuencia	Almacenamiento de la información y petición de cada aspirante

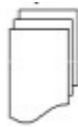
Tabla N° 4.13 Patrón de tarea Formulario.

- Patrones de elementos compuestos:

Nombre, clasificación, confianza y autor	Acordeón Elemento compuesto 0 Daniela Colmenares	
Problema	El usuario tiene que realizar una búsqueda en la barra de navegación principal	
Solución	Apilar los paneles horizontalmente y abrir uno de ellos en un momento determinado, mientras que el resto quedan cerrados	

	
Contexto	En la gestión del examen o curso instrumental, el administrador podrá ver el detalle de cada inscripción, ordenada por fecha de presentación del examen o fecha del comienzo del curso
Fuerzas	Sólo se puede ver el detalle de un examen o curso a la vez
Usabilidad	Satisfacción a todos los usuarios, fácil uso y acceso a la aplicación Web, tolerancia a fallos
Consecuencias	Organización de los exámenes o cursos

Tabla N° 4.14 Patrón de elemento compuesto Acordeón.

Nombre, clasificación, confianza y autor	Menú desplegable Elemento compuesto 0 Daniela Colmenares	
Problema	Los usuarios deben tener acceso a la navegación principal del sitio	
Solución	Presentar el menú en la posición del puntero del ratón después de que los usuarios hagan clic en la opción de su preferencia 	

Contexto	El administrador debe acceder a la aplicación y seleccionar la opción de su preferencia
Fuerzas	Se puede equivocar de selección
Usabilidad	Satisfacción a todos los usuarios, fácil uso y acceso a la aplicación Web, tolerancia a fallos
Consecuencias	Debe cambiar la interfaz a la deseada al momento de dar click

Tabla N° 4.15 Patrón de elemento compuesto Menú desplegable.

- Patrones de elementos simples:

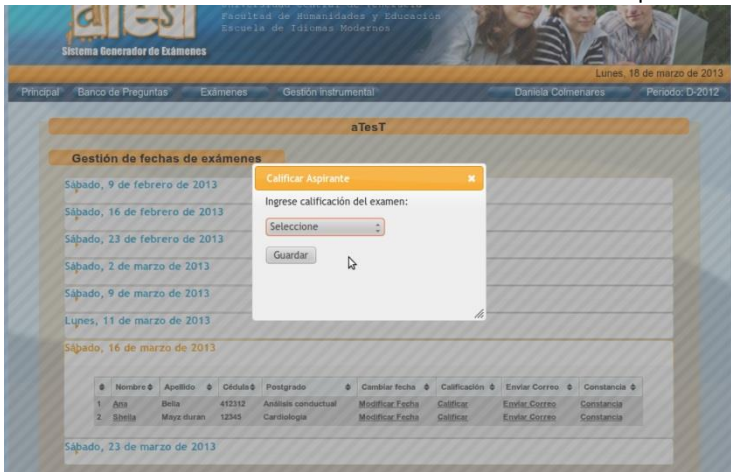
Nombre, clasificación, confianza y autor	Acceso directo de selección Elementos simples 0 Daniela Colmenares	
Problema	Los usuarios quieren acceder un valor específico de una manera directa	
Solución	Permite a los usuarios seleccionar un valor determinado de una serie de posibles datos 	
Contexto	El administrador debe ingresar una calificación para el examen o curso de cada aspirante	
Fuerzas	Se puede equivocar de selección	
Usabilidad	Satisfacción a todos los usuarios, fácil uso y acceso a la aplicación Web, tolerancia a fallos	
Consecuencias	Asignar una calificación para el examen o curso para cada aspirante	

Tabla N° 4.16 Patrón de elementos simples Acceso directo de selección

De la especificación anterior se puede llegar a la siguiente organización de los patrones de interacción (taxonomía) para la gestión evaluación instrumental:

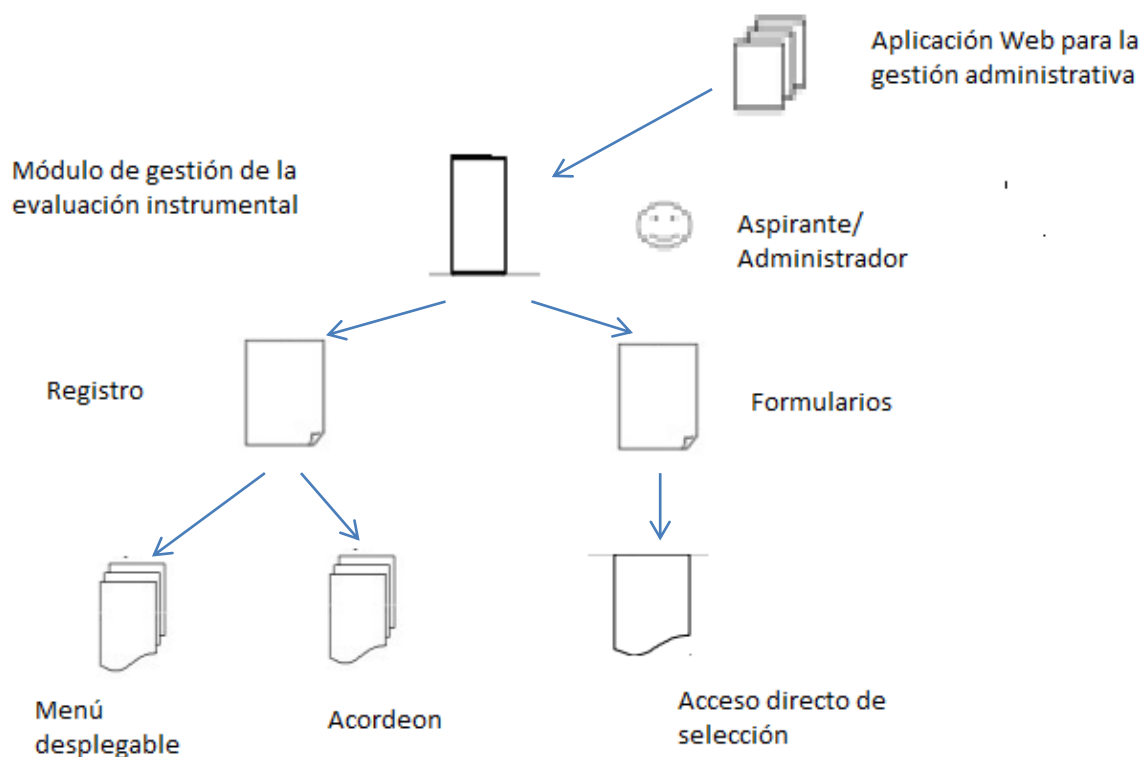


Ilustración N° 4.6 Taxonomía de los patrones de interacción para la evaluación instrumental.

4.2 Iteración 2 - Inscripción del curso y examen instrumental

Constituye la segunda etapa del método, en la cual se establecen las fases de prototipaje y entrega sobre el análisis del sistema. En la etapa de prototipaje fueron elaboradas diferentes interfaces una rápida y otra ejecutable, donde se puede apreciar el registro para realizar la inscripción del examen y curso instrumental. Posteriormente, se lleva a cabo una evaluación heurística, que consiste en un análisis de los aspectos de usabilidad para determinar si se encuentran incluidos en cada elemento de la interfaz de usuario, para un posterior análisis de requerimientos planteados por el personal administrativo de la Coordinación de Extensión de la Escuela de Idiomas Modernos para el desarrollo de las funcionalidades. Por parte de la fase de entrega, se prepara el ambiente para colocar en sistema en producción.

4.2.1 Prototipo rápido

Esta técnica está asociada a la idea de desarrollar diferentes conceptos propuestos mediante prototipos de software o hardware, para su posterior evaluación. La Ilustración N° 4.7 demuestra la interfaz que es el primer prototipo para el registro de la inscripción para el examen instrumental:

Nombres	
Apellidos	
Cédula	
Correo electrónico	
Número telefónico	
Número de depósito	
Postgrado que aspira	
	Guardar

Ilustración N° 4.7 Prototipo rápido para las inscripciones.

4.2.2 Prototipo ejecutable

Esta técnica consiste en una versión más real y detallada de la interfaz o prototipo de la aplicación. Las Ilustraciones N° 4.8 y N° 4.9 exhiben las pantallas de inscripción para el examen y curso instrumental respectivamente:

Lunes, 18 de marzo de 2013

Bienvenido Sr(a)

aTest

Principal

Bienvenidos al sistema de inscripciones para la evaluación instrumental
 Por favor ingrese sus datos para la formalizar su inscripción, todos son obligatorios
 La fecha en la que debe presentar su examen es: **Sábado, 23 de abril de 2013**

Cédula	Ej: 1234567
Nombre	Ej: Luis
Apellido	Ej: Zambrano
Correo electrónico	Ej: nombre@dominio.com
Número celular	Ej: 04145678990
Número de depósito	Ej: 001324324322
Postgrado de interés	Seleccione
Guardar	

Ilustración N° 4.8 Prototipo ejecutable de la inscripción del examen.

Lunes, 18 de marzo de 2013

Daniela Colmenares Período: D-2012

aTest Universidad Central de Venezuela
 Facultad de Humanidades y Educación
 Escuela de Idiomas Modernos

Sistema Generador de Exámenes

Principal Banco de Preguntas EX

Gestión de talleres

[Agregar nuevo estudiante](#)
[Consultar curso activo](#)
[Cerrar el curso activo](#)
[Modificar curso activo](#)
[Consultar estudiantes del curso activo](#)
[Consultar cursos anteriores](#)

Agregar un nuevo estudiante al curso

Por favor ingrese sus datos para la formalizar la inscripción, todos son obligatorios

Cédula	Ej: 1234553
Nombre	Ej: José
Apellido	Ej: 1234553
Correo electrónico	Ej: nombre@dominio.com
Número celular	Ej: 04141234553
Número de depósito	Ej: 001324212
Guardar	





Ilustración N° 4.9 Prototipo ejecutable de la inscripción del curso instrumental.

4.2.3 Evaluación heurística

Este tipo de pruebas se realizan con el fin de encontrar problemas de usabilidad en la interfaz de usuario. Para ello, es necesario utilizar las heurísticas creadas por Jakob Nielsen, ellas son (Wikipedia, Heurísticas de Nielsen, 2013):

- H1. Visibilidad de sistema. El sistema debe informar a los usuarios del estado del sistema, dando una retroalimentación apropiada en un tiempo razonable.

- H2. Utilizar el lenguaje de los usuarios. El sistema debe utilizar el lenguaje de los usuarios, con palabras o frases que le sean conocidas, en lugar de los términos que se utilizan en el sistema, para que al usuario no se le dificulte utilizar el sistema.
- H3. Control y libertad para el usuario. En caso que los usuarios elijan una opción del sistema por error, éste debe contar con las opciones de deshacer y rehacer para proveer al usuario de una salida fácil sin tener que utilizar diálogo extendido.
- H4. Consistencia y estándares. El usuario debe seguir las normas y convenciones de la plataforma sobre la que está implementando el sistema, para que no se tenga que preguntar el significado de las palabras, situaciones o acciones del sistema.
- H5. Prevención de errores. Es más importante prevenir la aparición de errores que de generar buenos mensajes de error.
- H6. Minimizar la carga de la memoria del usuario. En la interfaz se deben mantener los objetos, acciones y opciones visibles para que el usuario no pierda el tiempo recordando la información de una parte del diálogo a otra.
- H7. Flexibilidad y eficiencia de uso. Los aceleradores permiten aumentar la velocidad de interacción para el usuario experto tal que el sistema pueda atraer a usuarios principiantes y experimentados. Las instrucciones para el uso del sistema tienen que ser visibles o fácilmente accesibles cada vez que se necesiten, para que al usuario inexperto se le facilite aprender a utilizar el sistema. Es importante que el sistema permita personalizar acciones frecuentes, para acelerar el uso del sistema.
- H8. Los diálogos estéticos y diseño minimalista. La interfaz no debe contener información que no sea relevante o se utilice raramente, pues cada unidad adicional de información en un diálogo compite con las unidades relevantes de la información y disminuye su visibilidad relativa.
- H9. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores. Los mensajes de error deben expresarse en un lenguaje claro, indicar exactamente el problema, y deben ser constructivos.
- H10. Ayuda y documentación. En caso que el sistema necesite disponer de ayuda y documentación, ésta tiene que ser fácil de encontrar, centrada en las tareas del usuario, tener información de las etapas a realizar y que no sea muy extensa.

La escala de evaluación está comprendida entre 0-4 y contempla los siguientes valores:

- 0: No es un problema de usabilidad.

- 1: Problema cosmético.
- 2: Problema menor.
- 3: Problema grave de usabilidad, importante fijar solución.
- 4: Usabilidad catastrófica, imperativo fijar solución.

En la Tabla N° 4.16 se muestran los problemas encontrados durante la evaluación aplicada a cuatro (4) personas.

Problema	Heurísticas	Valoración	Solución
Para el ingreso a un aspirante nuevo al realizar la inscripción, éste debe escribir 'usuario' en el nombre y contraseña	H7	2	Colocar un acceso directo en la página principal para la inscripción de aspirantes nuevos para el examen instrumental
En el registro no se pide contraseña	H7	3	Pedir al usuario que introduzca su contraseña al momento de inscribirse por primera vez
No se señalan los campos obligatorios	H8	2	Señalar los campos obligatorios

Tabla N° 4.17 Tabla de evaluación heurística iteración 2.

4.2.4 Entrega

En esta etapa se actualizan las versiones del proyecto en el repositorio Git y se actualiza la versión de la aplicación que está en el servidor con la finalidad de agregar las nuevas funcionalidades al sistema. Finalmente, se entrega el software y es puesto en producción con todas las funcionalidades que se enmarcan en este Trabajo Especial de Grado, con la finalidad de realizar pruebas piloto para las inscripciones.

4.3 Iteración 3 Gestión Administrativa

Constituye la tercera etapa del método, en la cual se establecen las fases de prototipaje y entrega sobre el análisis del sistema. En la etapa de prototipaje fueron elaboradas diferentes interfaces una rápida y otra ejecutable, sobre la gestión administrativa. Posteriormente, se lleva a cabo una evaluación heurística, que consiste en un análisis de los aspectos de usabilidad para determinar si se encuentran incluidos en cada elemento de la interfaz de usuario, para un posterior análisis de requerimientos planteados por el personal directivo y administrativo de la Coordinación de Extensión de la Escuela de Idiomas Modernos para el desarrollo de las funcionalidades. Por parte de la fase de entrega, se prepara el ambiente para colocar en sistema en producción.

4.3.1 Prototipo rápido

Esta técnica está asociada a la idea de desarrollar diferentes conceptos propuestos mediante prototipos de software o hardware, para su posterior evaluación. Las Ilustraciones N° 4.10, N° 4.11 y N° 4.12 evidencian las interfaces que son los primeros prototipos para la gestión administrativa:

Facultad de Ciencias	
<u>Computación</u>	<u>Agregar examen</u>
<u>Examen 1</u>	<u>Eliminar examen</u>
<u>Examen 2</u>	<u>Eliminar examen</u>
<u>Biología</u>	<u>Agregar examen</u>
No posee exámenes	
Facultad de Humanidades	
Facultad de Odontología	
Facultad de Medicina	
Facultad de Ingeniería	

Ilustración N° 4.10 Prototipo rápido para el almacenamiento de exámenes.

[Buscar](#)

Sábado 9/03/2013

7/60 por calificar [Enviar Correo masivo](#)

Nombre	CI	Postgrado	Cambiar fecha presentación	Calificación	Enviar correo	Generar constancia
Daniela Colmenares	19023671	Computación	Modificar fecha	Calificar	Enviar Correo	Constancia
Jossie Zambrano	11111	Computación	Modificar fecha	Aprobado	Enviar Correo	Constancia

Sábado 16/03/2013

0/60 por calificar [Enviar Correo masivo](#)

Sábado 9/03/2013

0/60 por calificar [Enviar Correo masivo](#)

Cambiar fecha presentación examen

Fecha nueva presentación:

[Modificar](#)

Calificación

Calificación obtenida:

[Modificar](#)

Redactar correo electrónico

[Enviar](#)

Ilustración N° 4.11 Prototipo rápido para el manejo de fechas de presentación de los exámenes.

Configuración General

[Agregar fecha de presentación del examen](#)

[Eliminar fecha de presentación del examen](#)

[Cerrar inscripciones del próximo examen](#)

[Agregar fecha del taller](#)

[Agregar estudiantes para el taller](#)

[Generar reporte administrativo](#)

Ilustración N° 4.12 Prototipo rápido para la configuración general.

4.3.2 Prototipo ejecutable

Esta técnica consiste en una versión más real y detallada de la interfaz o prototipo de la aplicación. Las siguientes ilustraciones exhiben las pantallas de gestión administrativa por funcionalidad. Las cuales se describen a continuación:

Esta gestión consiste en diferentes actividades para el director y personal administrativo de la Coordinación de Extensión de la Escuela de Idiomas Modernos de la UCV, entre las que se encuentran:

- Gestionar fechas para exámenes: consiste en administrar a los aspirantes inscritos, organizados por fecha de presentación de examen. Esta interfaz demostrada en la Ilustración N° 4.13 contiene el examen activo en la fecha actual, y siete (7) fechas anteriores; cada una contiene información de los aspirantes inscritos como nombre, apellido, cédula, calificación y la posibilidad de obtener su constancia o enviar un correo electrónico.



Ilustración N° 4.13 Gestionar fechas para exámenes.

Otras funcionalidades que se pueden llevar a cabo son:

- Modificar fecha: donde se puede modificar la fecha de presentación de un aspirante que haya comunicado al personal administrativo que no puede asistir al examen activo para esa semana. Expuesta en la Ilustración N° 4.14.

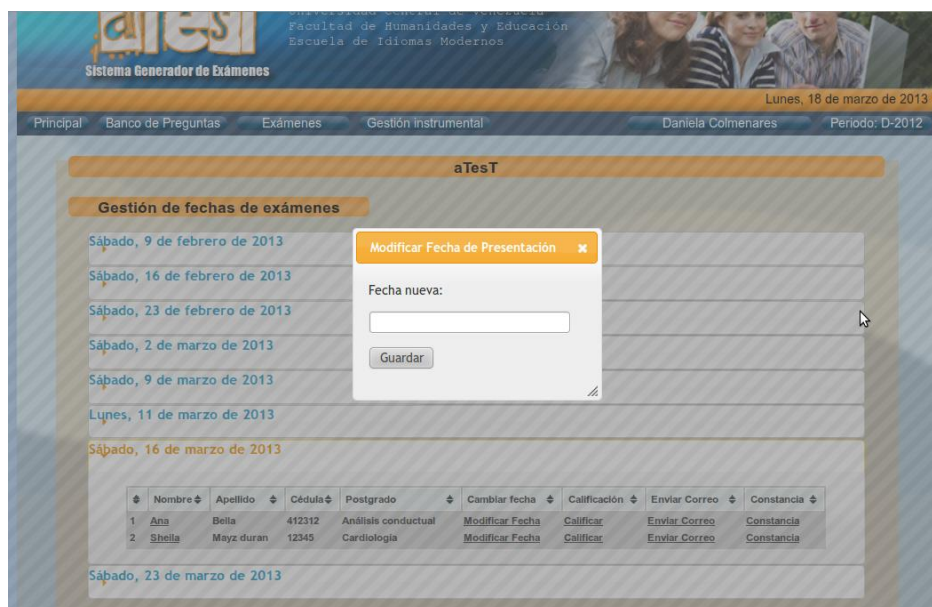


Ilustración N° 4.14 Funcionalidad para modificar fecha de presentación.

- Calificar: permite al director asignar una calificación al examen, los posibles valores son: aprobado, reprobado, no presentó el examen o no llevó comprobante del depósito bancario. Por defecto, todos los aspirantes se encuentran sin calificación. Una vez que el Director haya ingresado la nota de algún aspirante, el mismo recibirá un correo automáticamente informándole la calificación obtenida. Evidenciada en la Ilustración N° 4.15.

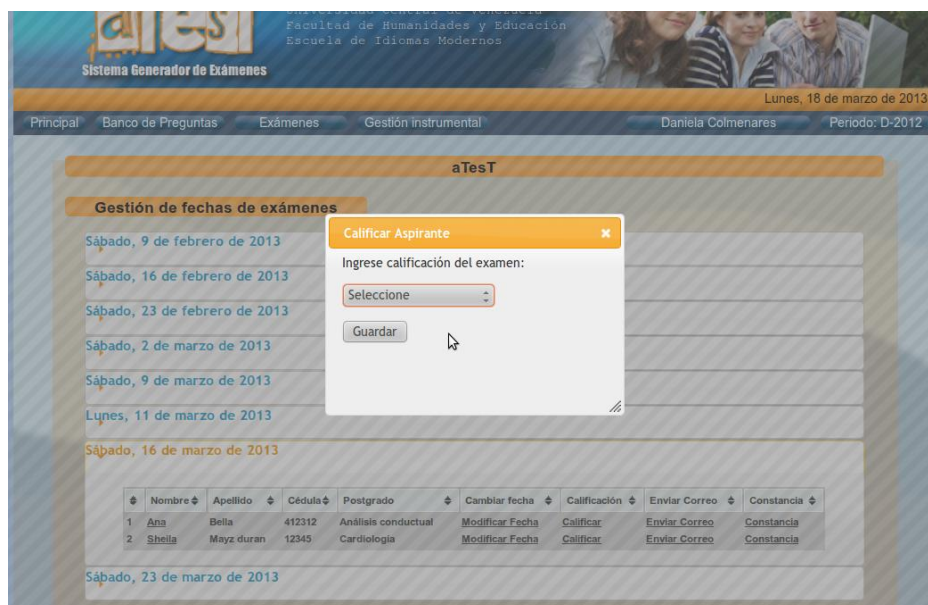


Ilustración N° 4.15 Funcionalidad para calificar.

- Enviar Correo: permite enviar un correo específico al aspirante de su preferencia.

Demostrado en la Ilustración N° 4.16.

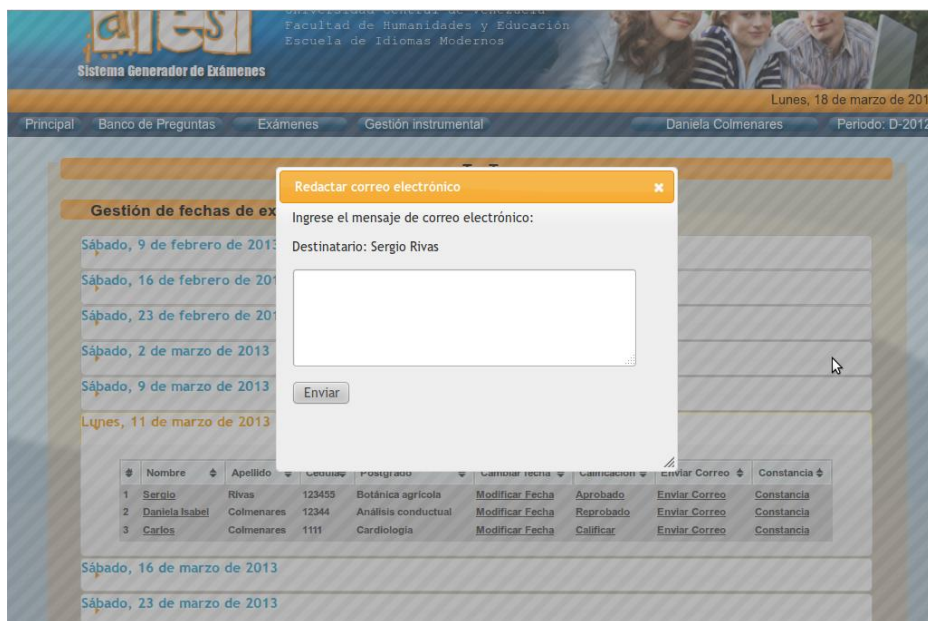


Ilustración N° 4.16 Funcionalidad para enviar correos.

- Repositorio del examen instrumental: permite almacenar y descargar exámenes por cada postgrado de las diferentes facultades de la UCV. Demostrado en la Ilustración N° 4.17.

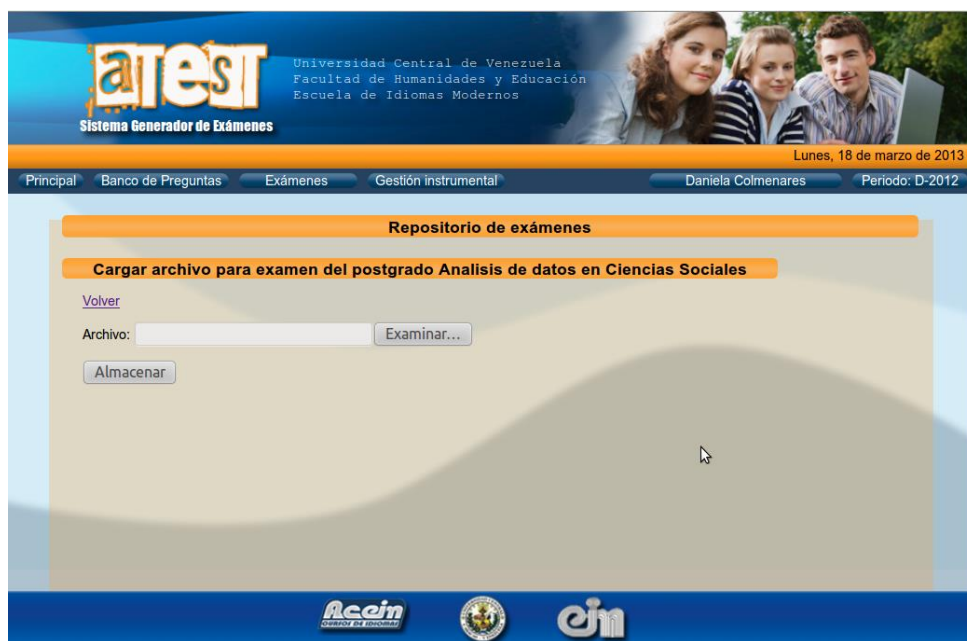


Ilustración N° 4.17 Repositorio de los exámenes.

- Gestión del curso instrumental: el personal directivo y administrativo podrán administrar la inscripción de un curso en un área específica. Expuesto en la Ilustración N° 4.18.



Ilustración N° 4.18 Gestión del curso instrumental.

Posee las siguientes funcionalidades:

- Agregar nuevo estudiante: se podrá inscribir a un estudiante, ingresando sus datos personales y número del depósito bancario. Evidenciado en la Ilustración N° 4.19.

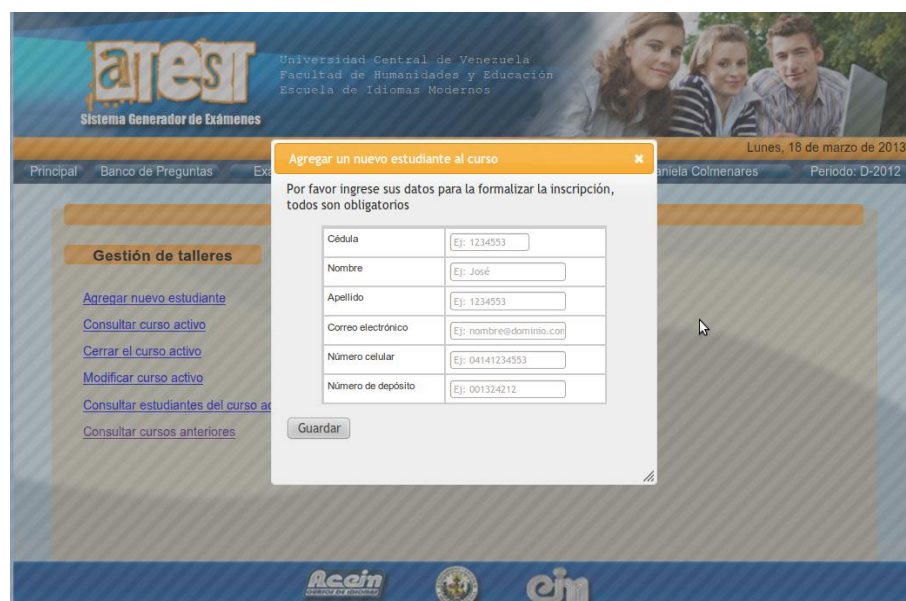


Ilustración N° 4.19 Funcionalidad para agregar un nuevo estudiante.

- Consultar curso activo: permite consultar el área, fecha de comienzo y cantidad de alumnos inscritos para la fecha. Evidenciado en la Ilustración N° 4.20.

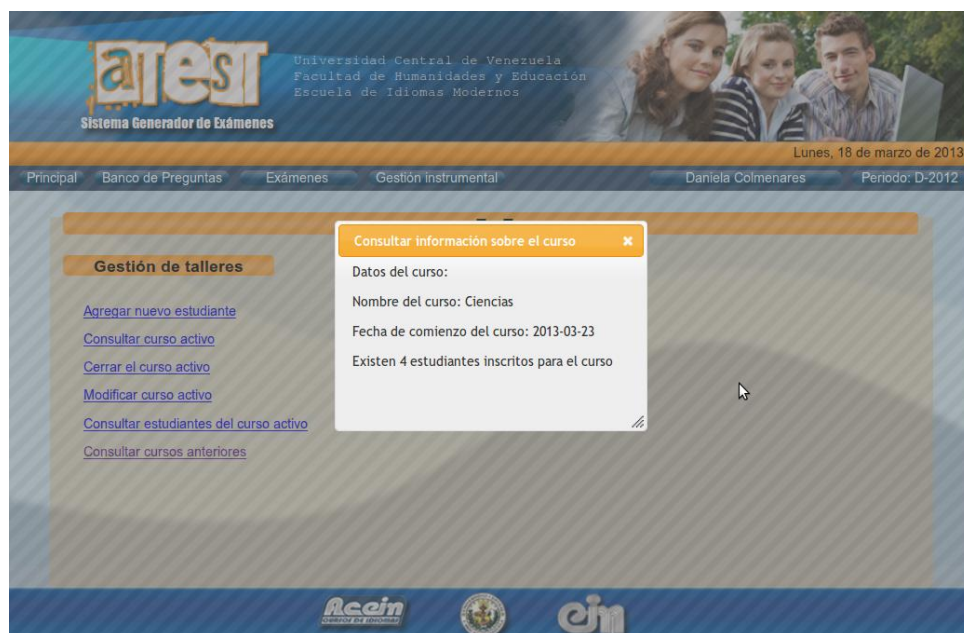


Ilustración N° 4.20 Funcionalidad para consultar curso activo.

- Cerrar curso activo: permite cerrar las inscripciones del curso activo en la actualidad. Demostrado en la Ilustración N° 4.21.

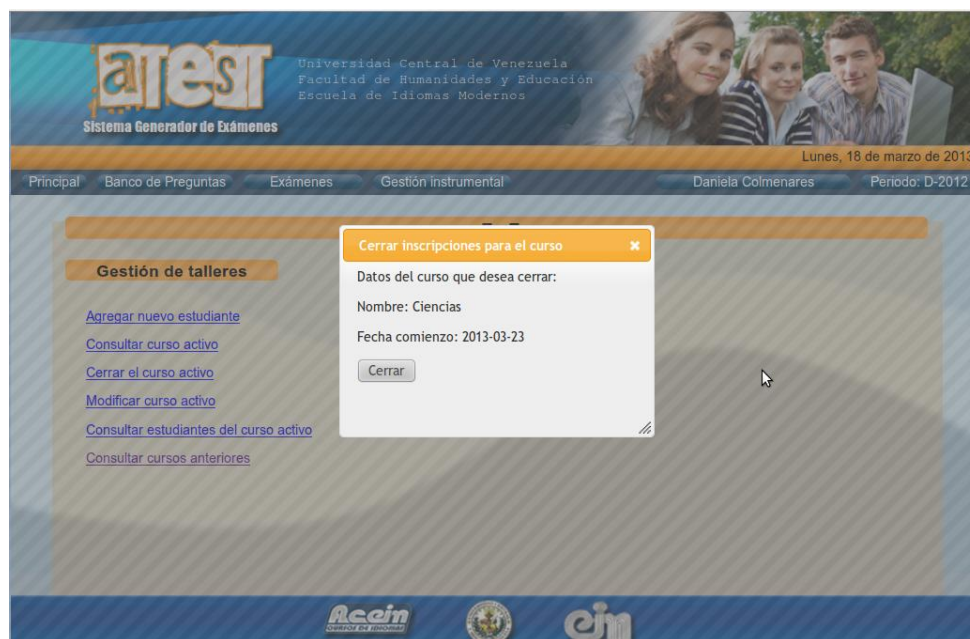


Ilustración N° 4.21 Funcionalidad para cerrar el curso activo.

- Consultar estudiantes del curso activo: permite consultar los datos personales de los estudiantes inscritos para el curso activo en la actividad. Además brindará la posibilidad de mandar correo electrónico para cada aspirante. Expuesto en la Ilustración N° 4.22.

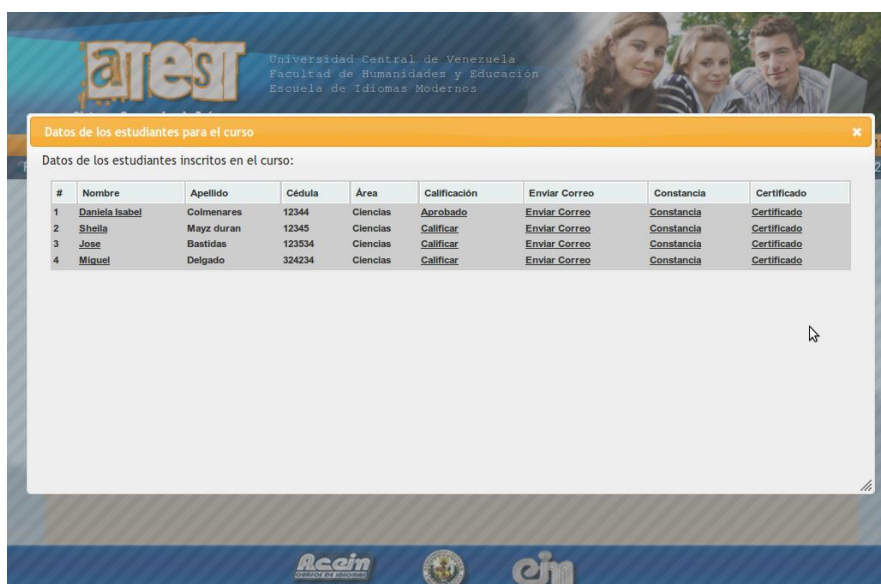


Ilustración N° 4.22 Funcionalidad para consultar estudiantes del curso activo.

- Consultar cursos anteriores: permite consultar los datos de todos los cursos que se han registrado anteriormente, con sus respectivos aspirantes, información personal y calificación. Evidenciado en la Ilustración N° 4.23.

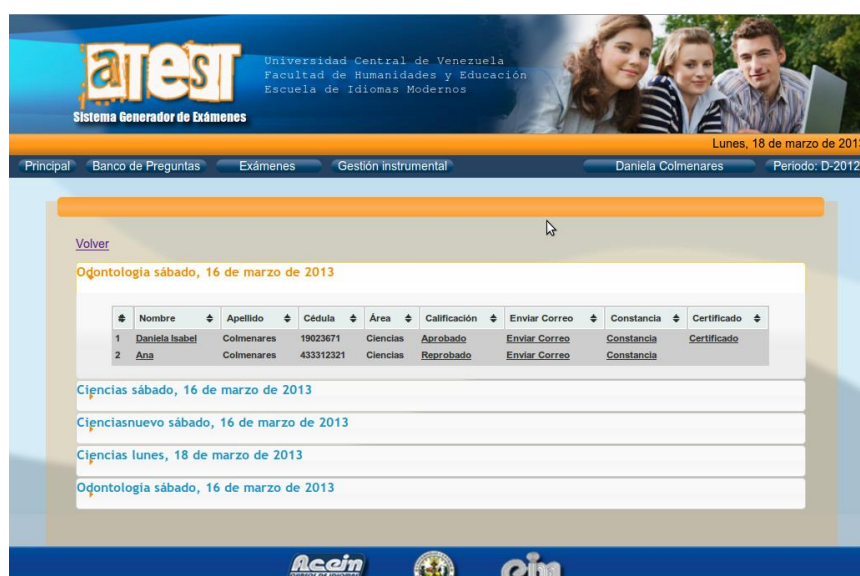


Ilustración N° 4.23 Funcionalidad para consultar cursos anteriores.

- Cerrar inscripciones para el examen activo: permite cerrar las inscripciones para la fecha de presentación del examen activo en un momento dado. Al momento de confirmar esta operación, automáticamente se abrirán las inscripciones para la próxima fecha de presentación que se tenga registrada en el sistema. Expuesta en la Ilustración N° 4.24.

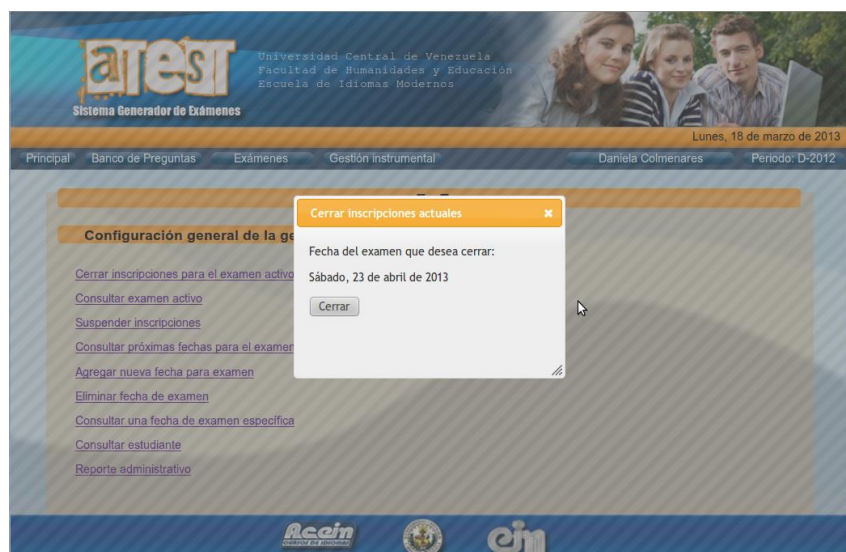


Ilustración N° 4.24 Funcionalidad para cerrar la inscripción del examen activo.

- Consultar examen activo: permite consultar la fecha de presentación del examen activo para esa semana, y la cantidad de aspirantes inscritos para ese momento, organizados por postgrado. Demostrado en la Ilustración N° 4.25

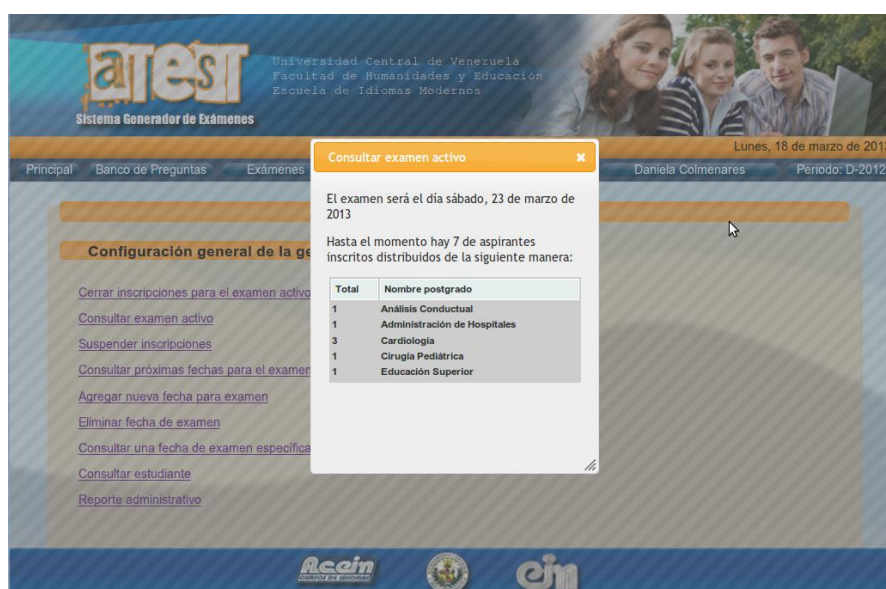


Ilustración N° 4.25 Funcionalidad para consultar examen activo.

- Suspendir inscripciones: permite suspender las inscripciones para el examen instrumental por el tiempo que considere el administrador. Evidenciado en la Ilustración N° 4.26.



Ilustración N° 4.26 Funcionalidad para suspender inscripciones.

- Consultar próximas fechas para el examen: el administrador podrá consultar las próximas ocho (8) fechas de presentación que se encuentran registradas en el sistema. Evidenciado en la Ilustración N° 4.27.

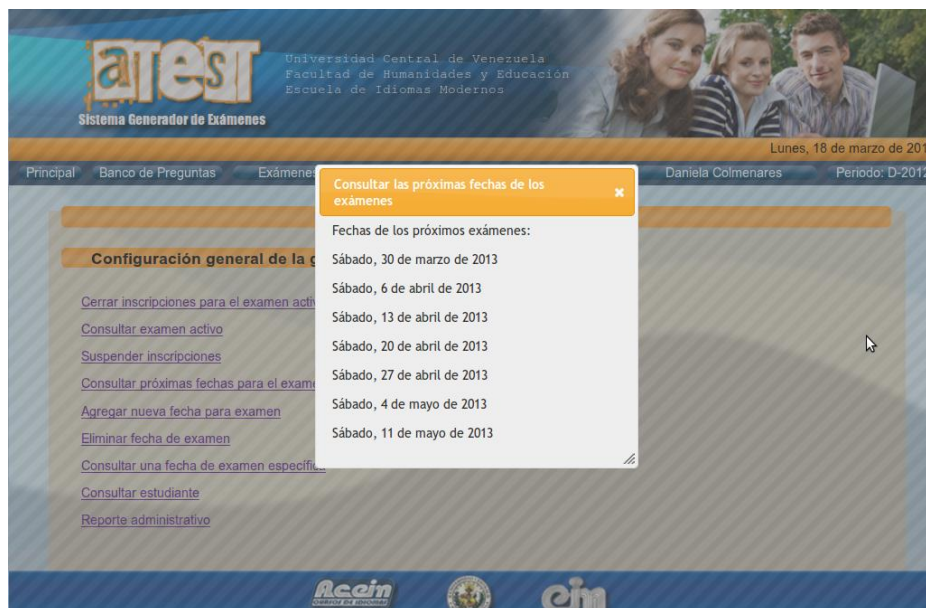


Ilustración N° 4.27 Funcionalidad para consultar próximas fechas de presentación.

- Agregar nueva fecha para el examen: permite agregar una nueva fecha para el examen que no se encuentre registrada en el sistema. Demostrado en la Ilustración N° 4.28.

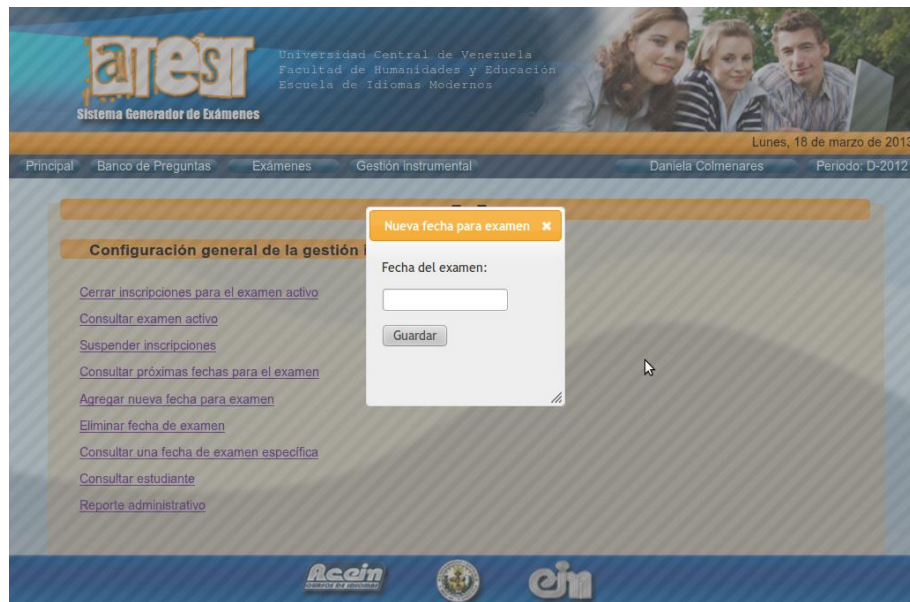


Ilustración N° 4.28 Funcionalidad para agregar nueva fecha de presentación.

- Eliminar fecha de examen: elimina una fecha de presentación que se encuentre registrada en el sistema. Expuesto en la Ilustración N° 4.29.

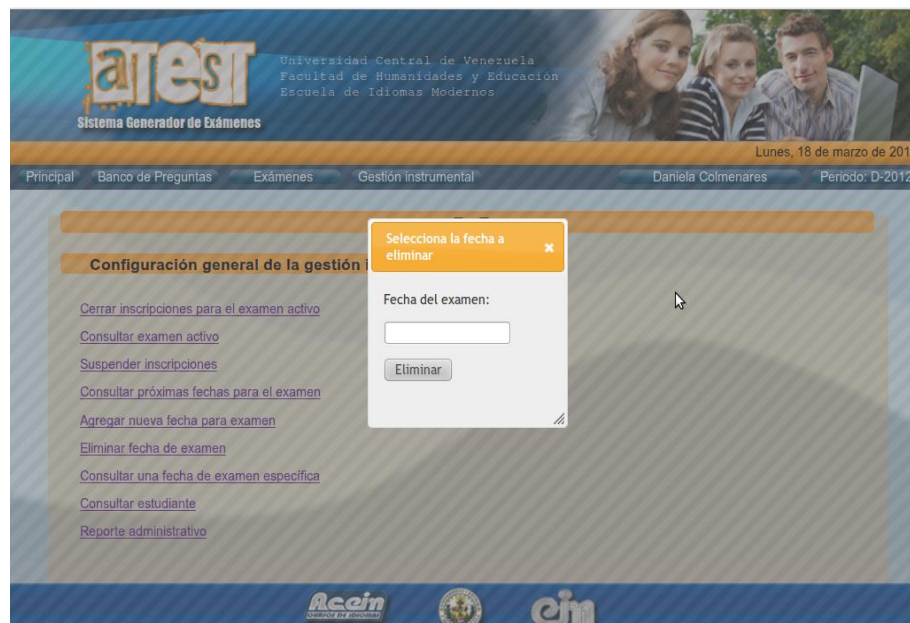


Ilustración N° 4.29 Funcionalidad para eliminar una fecha de presentación.

- Consultar una fecha de examen específica: permite consultar los aspirantes y descripción sobre una fecha de presentación del examen que desee el administrador. Expuesto en la Ilustración N° 4.30.

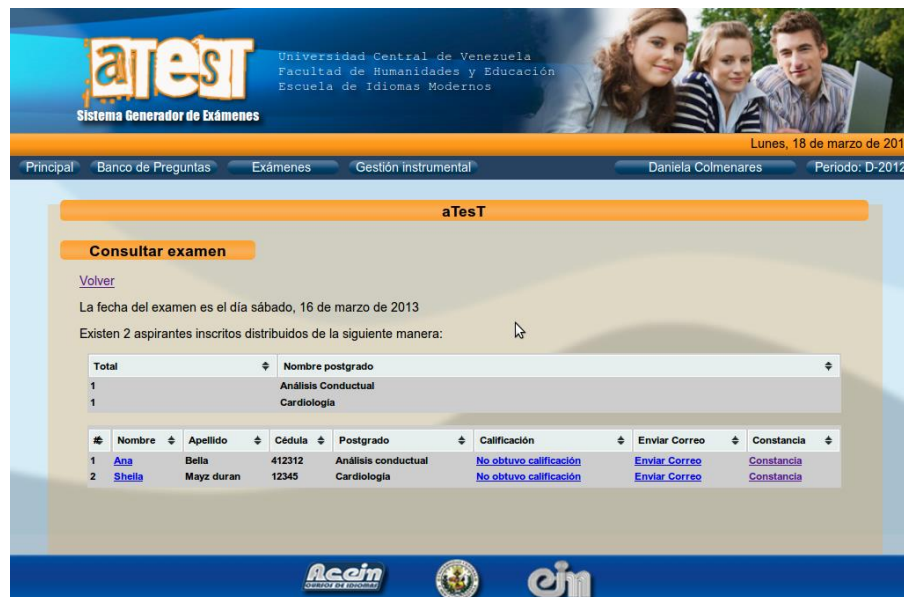


Ilustración N° 4.30 Funcionalidad para consultar una fecha de presentación específica.

- Consultar estudiante: permite consultar un aspirante en específico, ver su historial, enviar correo y otras funcionalidades relacionadas a ese aspirante. Evidenciado en la Ilustración N° 4.31.

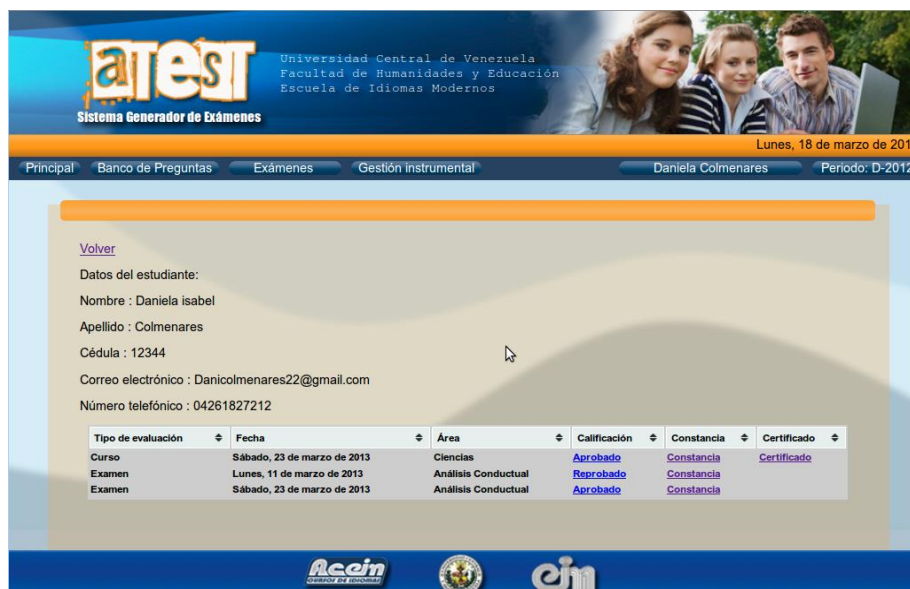


Ilustración N° 4.31 Funcionalidad para consultar un estudiante.

4.3.3 Evaluación heurística

En la Tabla N° 4.27 se muestran los problemas encontrados durante la evaluación aplicada a seis (6) personas.

Problema	Heurísticas	Valoración	Solución
Ingresar fecha en los campos necesarios	H8	2	Facilitar un calendario para seleccionar la fecha deseada
No se puede visualizar los exámenes almacenados previamente	H5 y H7	3	Listar los exámenes almacenados previamente para poder descargarlos
No se puede enviar correo individual	H8	2	Colocar opción para enviar correo individual

Tabla N° 4.18 Tabla de evaluación heurística para la iteración 3.

4.3.4 Entrega

En esta etapa se actualizan las versiones del proyecto en el repositorio Git y se actualiza la versión de la aplicación que está en el servidor con la finalidad de agregar las nuevas funcionalidades para la gestión administrativa. Finalmente, se entrega el software y es puesto en producción con todas las funcionalidades que se enmarcan en este Trabajo Especial de Grado, con la finalidad de realizar pruebas piloto.

4.4 Iteración 4 Generación de reportes

Constituye la cuarta y última iteración, en donde la Coordinación de Extensión de la Escuela de Idiomas Modernos necesita una alternativa para generar constancias y certificados a los aspirantes que aprueban tanto el curso como el examen instrumental. Es por esto que se llevó a cabo la elaboración de un plan de iteración sobre reportes, el cual consiste en la generación de información haciendo uso de los datos existentes, dispuesta de una forma sencilla y comprensible para facilitar el análisis y la toma de decisiones por parte del grupo administrativo y directivo.

Se establecen las fases de prototipaje y entrega sobre el análisis del sistema. En la etapa de prototipaje fueron elaboradas diferentes interfaces una rápida y otra ejecutable. Posteriormente, se lleva a cabo una evaluación heurística, que consiste en un análisis de los aspectos de usabilidad para determinar si se encuentran incluidos en cada elemento de la interfaz de usuario, para un posterior análisis de requerimientos planteados por el personal directivo y administrativo de la Coordinación de Extensión de la Escuela de Idiomas Modernos para el desarrollo de las funcionalidades. Por parte de la fase de entrega, se prepara el ambiente para colocar en sistema en producción.

4.4.1 Prototipo rápido

Esta técnica está asociada a la idea de desarrollar diferentes conceptos propuestos mediante prototipos de software o hardware, para su posterior evaluación. La Ilustración N° 4.32 representa la interfaz del primer prototipo para la gestión de reportes:

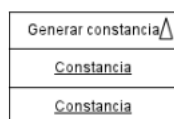


Ilustración N° 4.32 Prototipo rápido para generar reportes.

4.4.2 Prototipo ejecutable

Esta técnica consiste en una versión más real y detallada de la interfaz o prototipo de la aplicación. Las siguientes Ilustraciones exhiben las pantallas de gestión administrativa por funcionalidad. Las cuales se describen a continuación:

Esta gestión consiste en diferentes actividades para el director y personal administrativo de la Coordinación de Extensión de la Escuela de Idiomas Modernos de la UCV, evidenciados en las Ilustraciones N° 4.33 y N° 4.34:



Ilustración N° 4.33 Solicitud de constancia.

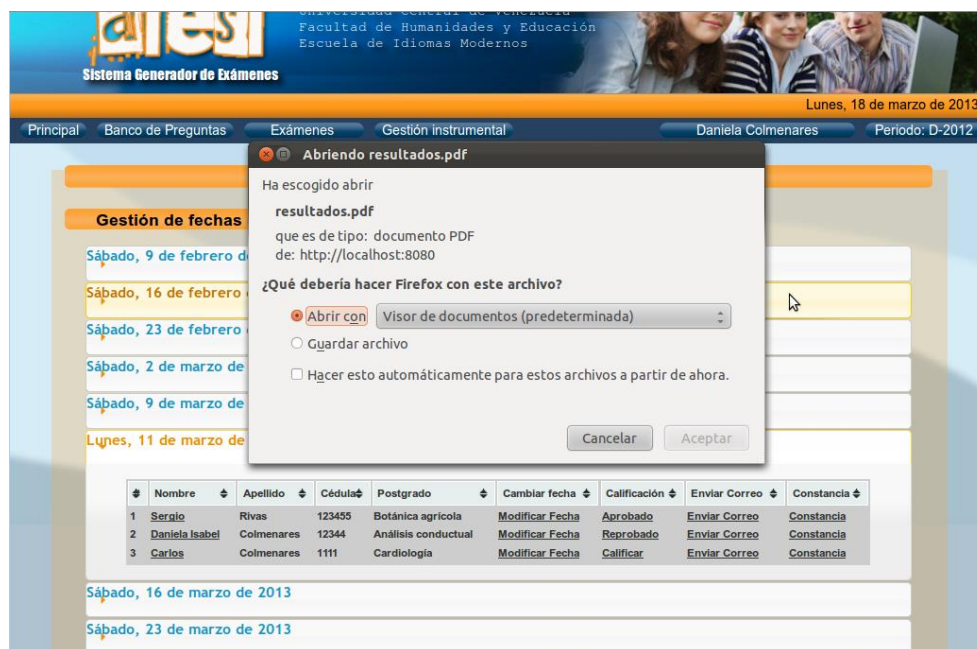


Ilustración N° 4.34 Descarga del reporte.

4.4.3 Evaluación heurística

En la Tabla N° 4.18 se muestran los problemas encontrados durante la evaluación aplicada a seis (6) personas.

Problema	Heurísticas	Valoración	Solución
El aspirante que aprueba el examen o curso instrumental no puede descargar su constancia o certificado	H5 y H7	3	Colocar un enlace a los aspirantes que aprueban para que puedan descargar su constancia o certificado
El administrador puede descargar constancia cuando el estudiante ha reprobado el curso o examen	H5 y H7	3	Sólo se puede descargar constancia o certificado a los aspirantes que aprueban el curso o examen

Tabla N° 4.19 Tabla de evaluación heurística para la iteración 4.

4.4.3 Entrega

En esta etapa se actualizan las versiones del proyecto en el repositorio Git y se actualiza la versión de la aplicación que está en el servidor con la finalidad de agregar las nuevas funcionalidades al sistema. Finalmente, se entrega el software y es puesto en producción con todas las funcionalidades que se enmarcan en este Trabajo Especial de Grado, con la finalidad de realizar pruebas piloto para la generación de algunos reportes.

4.5 Aceptación de la aplicación

Una vez que la aplicación fue puesta en producción, se realizó una simulación de la inscripción para el examen instrumental, y pruebas de aceptación a los usuarios que lo ejecutaron basado en una lista de chequeo de usabilidad con una escala para medir el grado de satisfacción (siendo 5 la calificación más alta), que se muestra a continuación en la Tabla N° 4.19:

Pregunta	1	2	3	4	5
¿Considera usted que la aplicación es fácil de aprender?					
¿Considera usted que la aplicación provee feedback a las acciones que realizó?					
¿Considera usted que la aplicación posee una interfaz simple, sencilla y organizada?					
¿Considera usted que la aplicación está diseñada para minimizar las posibilidades de cometer errores?					
¿Considera usted que la aplicación posee controles del sistema claramente visibles y acorde a las funciones que cumplen?					

Tabla N° 4.20 Lista de chequeo.

La lista de chequeo se aplicó a diez (10) usuarios, los resultados son expuestos en la Tabla N° 4.20:

Pregunta	1	2	3	4	5
¿Considera usted que la aplicación es fácil de aprender?				1 (10%)	9 (90%)
¿Considera usted que la aplicación provee feedback a las acciones que realizó?					10 (100%)
¿Considera usted que la aplicación posee una interfaz simple, sencilla y organizada?			1 (10%)	1 (10%)	8 (80%)
¿Considera usted que la aplicación está diseñada para minimizar las posibilidades de cometer errores?				2 (20%)	8 (80%)
¿Considera usted que la aplicación posee controles del sistema claramente visibles y acordes a las funciones que cumplen?				1 (10%)	9 (90%)

Tabla N° 4.21 Resultado de la Lista de Comprobación aplicada a 10 usuarios.

Gracias a estos resultados se evidencia que la aplicación tuvo una buena aceptación en la simulación de la inscripción para el examen instrumental, debido a que el 90% de los usuarios encuestados consideran que es de fácil aprendizaje, el 100% piensa que la aplicación provee feedback a las acciones que concierne la inscripción, el 80% considera que posee una interfaz simple, sencilla y organizada, el 90% piensa que posee controles del sistema claramente visibles y acordes a las funciones que cumplen y un 80% considera que la aplicación está diseñada para minimizar las posibilidades de cometer errores. Por lo tanto, este módulo cumple con las expectativas de los usuarios en cuestión y el objetivo principal de inscribirlos a un examen instrumental.

CONCLUSIONES Y RECOMENDACIONES

Al igual que muchas organizaciones, la Coordinación de Extensión de la Escuela de Idiomas Modernos se ha visto beneficiada con el uso de aplicaciones Web para mejorar sus servicios. Sin embargo, existen procesos que se realizan de forma no automatizada, razón por la cual se cumplió el objetivo general de esta investigación, ya que se desarrolló exitosamente un nuevo módulo en aTeST para la gestión de procesos relacionados a la evaluación instrumental.

El desarrollo de este módulo se realizó siguiendo el método AgilUs, que permitió implementar un conjunto de técnicas y actividades de forma organizada y rápida. Además, AgilUs fue posible de adaptar fácilmente a las necesidades por parte del equipo de desarrollo y del usuario para la implementación de esta aplicación. La característica iterativa e incremental de este método es un factor relevante que permite dividir el sistema en subsistemas que agrupan distintas funcionalidades que se entregan progresivamente para que puedan ser utilizadas, sin la necesidad de implementar la aplicación en su totalidad. Por otra parte, involucrar al usuario final desde el principio del desarrollo del sistema, es un factor clave de este método que permite adaptar el software a las necesidades de éste generando mayor satisfacción a los usuarios y no viceversa. En cuanto a la usabilidad, AgilUs propone técnicas que permiten medir este factor de calidad en un sistema. Se aplicaron evaluaciones heurísticas a distintos usuarios con el perfil necesario para usar la aplicación y se encontraron problemas de usabilidad que fueron solventados. Además, se facilitó la interfaz de los distintos módulos a los potenciales usuarios del sistema y éstos aceptaron la usabilidad de los respectivos módulos evidenciado gracias a las pruebas de aceptación de la aplicación (lista de chequeo).

Los patrones de proceso generaron soluciones exitosas para la automatización de los procesos de inscripción, en el dominio correspondiente a la gestión académica administrativa. Ya que ayudaron a implementar de una manera más eficiente cada tarea que involucran estas inscripciones, gracias a que este proceso se divide en la solicitud de inscripción por parte del aspirante y la gestión de ésta por parte del administrador.

Las tecnologías del lado cliente y servidor que fueron utilizadas para el desarrollo de esta aplicación fueron apropiadas y de vanguardia; y facilitaron la implementación. Con excepción del framework Ruby on Rails, ya que se generaron problemas por cuanto fallas que se presentaban en él, pero que fue posible resolver tras la instalación de algunas gemas. Otro inconveniente fue la discrepancia que existe entre los ambientes de ejecución de la aplicación, es decir, entre el

ambiente de desarrollo y producción. Ya que el ambiente de producción posee formato de codificación de caracteres Unicode de 8 bits (UTF – 8) para la base de datos.

La aplicación de estándares de programación ayudó a la obtención de un código legible y ordenado, el cual facilita el mantenimiento y evolución de la aplicación a lo largo del tiempo. Mediante la implementación del patrón MVC, se facilitó la incorporación de las nuevas funcionalidades a la aplicación ya existente.

Para implementar los módulos que permitan gestionar la inscripción de los estudiantes de postgrado de la UCV, la gestión administrativa y la generación de reportes, se obtuvieron listas de requerimientos que engloban todas las funcionalidades para estas actividades. Además, se desarrolló cada requerimiento como una funcionalidad del sistema y se presentó al usuario para determinar su aprobación.

La inscripción y gestión inicialmente se trataban de procesos complejos y costosos. Con las funciones automatizadas que actualmente se tienen a disposición, resulta más sencillo llevar a cabo la inscripción o gestión administrativa. La información se encuentra clasificada y ordenada, lo cual permite acceder directamente a los elementos necesarios para que se lleven a cabo los procesos de manera satisfactoria. El personal administrativo no depende del grupo de desarrollo ya que tiene a disposición las herramientas necesarias para realizar sus actividades asignadas. Existe un control de errores, los cuales son validados por la aplicación. Las funcionalidades pueden ser accedidas desde cualquier lugar y en cualquier momento, permitiendo realizar algún cambio en caso de surgir algún imprevisto.

Las recomendaciones que están fuera del alcance del trabajo especial de grado y que pudieran ser aplicadas a las bases del sistema existente, son las siguientes:

- Agregar idiomas extranjeros diferentes al inglés para el examen instrumental.
- Realizar un módulo para enviar mensajes de texto a celulares, con la finalidad de informar noticias por un medio más rápido sobre el curso o examen.
- Elaborar reportes financieros importantes a las necesidades económicas de la Coordinación y Extensión de la Escuela de Idiomas Modernos.

Este trabajo permite la automatización de un proceso importante dentro de la Coordinación de Extensión de la Escuela de Idiomas Modernos de la UCV, logrando así un ahorro en tiempo, recurso humano y costos. Además, la Coordinación cuenta con un repositorio de exámenes que pueden ser consultados, modificados e incluso impresos nuevamente.

REFERENCIAS BIBLIOGRÁFICAS

(Acosta, 2011) Acosta Alecia Eleonora. "AgilUs: un método ágil de desarrollo de software que incorpora la usabilidad" Lecturas en Ciencia de la Computación. Universidad Central de Venezuela, Facultad de Ciencias, Escuela de Computación.

(AJAX, 2012) Asynchronous JavaScript And XML. Consultado el 5 de abril del 2012 en <http://www.webtaller.com/maletin/articulos/que-es-ajax.php>

(Apache, 2012) Servidor HTTP Apache. Consultado el 6 de abril de 2012 en http://es.wikipedia.org/wiki/Servidor_HTTP_Apache

(Aplicaciones Web, 2012) Aplicaciones Web. Consultado el 6 de abril del 2012 en http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_Web
<http://www.maestrosdelweb.com/editorial/css-3-mas-social-que-nunca/>

(Arquitectura Cliente/Servidor, 2012) Consultado el 5 de abril del 2012 en <http://www.tucodigofuente.com/2011/11/24/acceso-a-datos-programacion-por-capas/>

(Ballard & Moncur, 2012) AJAX, JavaScript y PHP. Consultado el 5 de abril del 2012.

(Debian, 2012) El proyecto Debian. Consultado el 4 de abril del 2012 en <http://www.debian.org/intro/about>

(Eguíluz, 2012 .a) Introducción a JavaScript. Consultado el 5 de abril del 2012.

(Eguíluz, 2012 .b) Introducción a CSS. Consultado el 4 de abril del 2012.

(Harwani, 2012) jQuery Recipes. Consultado el 5 de abril del 2012.

(Hogan, 2012 .a) HTML5 & CSS3. Consultado el 4 de abril del 2012.

(Hogan, 2012. b) HTML5 & CSS3. Consultado el 4 de abril del 2012.

(Ingeniería, 2012) Consultado el 16 de septiembre del 2012 en http://www.ulavirtual.cl/courses/IS2007/document/Apuntes_Primer_Prueba_DELETED_2/Libro_de_ingenieria_de_software_%28Completo%29/29-Cap%EDtulo.pdf

(jQuery UI, 2012). Consultado el 5 de abril del 2012 en <http://jqueryui.com/home>

(Metodología, 2012) Pérez Isaías, Pérez Rodrigo, Rodríguez Aureliano "Metodologías de desarrollo de software" 2008, consultado el 16 de septiembre del 2012.

(MVC, 2012) Modelo Vista Controlar. Consultado el 4 de abril del 2012 en <http://www.comusoft.com/modelo-vista-controlador-definicion-y-caracteristicas>

(MySQL, 2012) MySQL. Consultado el 19 de abril del 2012 en <http://www.webestilo.com/mysql/intro.phtml>

(Reusabilidad, 2013) Abril Falcón, Jonathan Paúl, Otazú Estrada, Víctor Luis. Gestión de reuso de software. Universidad Peruana de Ciencias Aplicadas. Lima-Perú 2009.

(Ruby & Thomas & Hansson, 2012) *Agile Web Development with Rails Fourth Edition*. Consultado el 6 de abril del 2012.

(Scrum, 2012) Scrum. Consultado el 16 de septiembre del 2012 en <http://scrummethodology.com/>

(Servicios, 2012) Servicios Web. Consultado el 28 de julio del 2012 en <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>

(Unicorn, 2012) Servidor HTTP Unicorn. Consultado el 8 de abril del 2012 en <http://unicorn.bogomips.org/>

(XP, 2012) "Una amplia explicación sobre metodologías ágiles y la programación extrema (XP)". Publicación de la Universidad Politécnica de Valencia. Consultada el 16 de septiembre en <http://www.willydev.net/descargas/prev/TodoAgil.pdf>

(Zambrano, 2013) Jossie Zambrano, Lic. y Alecia Eleonora Acosta Dra. Un modelo de Gestión Académico Administrativa Universitaria: Proceso de Inscripción en la Evaluación Diagnóstica por Área de Conocimiento Ciencia y Tecnología (Octubre 2012). Centro de Investigación de Software y Sistemas (ISYS) Escuela de Computación Facultad de Ciencias Universidad Central de Venezuela