



Universidad Central de Venezuela

Facultad de Ciencias

Escuela de Computación

Centro de Enseñanza Asistida por Computador - CENEAC

**IMPLEMENTACIÓN DEL MÓDULO DE GESTIÓN DE
TRÁMITES DE PREPARADORES Y AUXILIARES DOCENTES
DEL SISTEMA DE GESTIÓN DE LA PROGRAMACIÓN
DOCENTE PARA EL DEPARTAMENTO DE LA ESCUELA
DE COMPUTACIÓN DE LA FACULTAD DE CIENCIAS**

Trabajo Especial de Grado presentado ante la Ilustre
Universidad Central de Venezuela por los Bachilleres
Christian Manuel Brites De Freitas (C.I. V-20.676.340)
Andrés Alberto De Freitas Ascanio (C.I. V-19.960.207)
para optar al título de Licenciado en Computación

Tutora: Profa. Yusneyi Y. Carballo Barrera

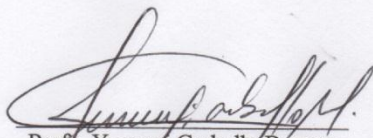
Caracas, 23 de Octubre del 2015

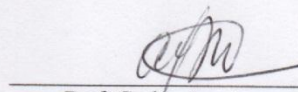
ACTA

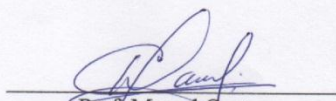
Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado **"Implementación del Módulo de Gestión de Trámites de Preparadores y Auxiliares Docentes del Sistema de Gestión de la Programación Docente para el Departamento de la Escuela de Computación de la Facultad de Ciencias"** y presentado por los Bachilleres **Christian Manuel Brites De Freitas C.I. V-20.676.340** y **Andrés Alberto De Freitas Ascanio C.I. V-19.960.207**, a los fines de optar al título de Licenciado en Computación, dejamos constancia de lo siguiente:

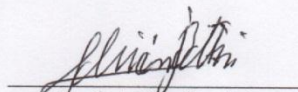
Leído como fue dicho trabajo, por cada uno de los miembros del jurado, se fijó el día **Viernes, 23 de Octubre de 2015**, a las **2:00 pm horas**, para que los autores lo defendieran en forma pública, lo que estos hicieron en **Sala Planta Alta-III** de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondieron a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar con la nota de 12 puntos.

En fe de lo cual se levanta la presente Acta, en Caracas el día 23 de Octubre de 2015.


Prof. Yusney Carballo Barrera
(Tutora firmante)


Prof. Carlos Acosta
(Tutor)


Prof. Marcel Castro
(Jurado)


Prof. Adrian Bottini
(Jurado)

AGRADECIMIENTOS

A Dios que ha estado conmigo en este recorrido, y me ha ayudado en todos los momentos de dificultad que se han presentado, dándome fuerza y sabiduría para seguir adelante.

A mis padres, hermana, abuelos y demás familiares, por guiarme siempre por el mejor camino, por estar en las buenas y en las malas, y apoyarme cuando más lo necesitaba.

A mi amigo, hermano y compañero de TEG Andrés De Freitas, por la confianza y apoyo que me ha brindado en la carrera.

A Stephanie, por ser la mejor amiga y novia, gracias por tu paciencia y apoyo cuando más lo necesitaba.

A nuestra tutora, Yusneyi Carballo por indicarnos siempre el mejor camino a seguir y prestarnos toda su colaboración para poder llevar a cabo este proyecto.

A todos los mencionados y faltantes, gracias.

Christian Brites

AGRADECIMIENTOS

A mi familia, en especial a mis padres y a mi hermano por haberme apoyado y ayudado incondicionalmente a lo largo de todo este camino, este logro también es de ustedes.

A la familia Brites De Freitas, por recibirme en su casa innumerables veces como a uno más de su familia.

A mi amigo y compañero de tesis, Christian Brites, por su ayuda y apoyo en nuestros proyectos.

A nuestra tutora, Yusneyi Carballo, por toda la ayuda que nos ha dado, y por guiarnos en este camino.

A todos los amigos que he hecho a lo largo de la carrera, que de una u otra forma me ayudaron a estar donde estoy hoy.

A mis colegas de 4Geeks, que me ayudaron a comenzar este camino en el mundo del desarrollo web.

A todos ustedes, y a los que falten, muchísimas gracias.

Andrés De Freitas



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro de Enseñanza Asistida por Computador - CENEAC

IMPLEMENTACIÓN DEL MÓDULO DE GESTIÓN DE PREPARADORES DEL SISTEMA DE GESTIÓN DE LA PROGRAMACIÓN DOCENTE PARA EL DEPARTAMENTO DE LA ESCUELA DE COMPUTACIÓN DE LA FACULTAD DE CIENCIAS

Autores: Brites, Christian. **C.I. V-20.676.340**

Correo-e: christianbrites@gmail.com

De Freitas, Andrés. **C.I. V-19.960.207**

Correo-e: adefreitas12@gmail.com

Tutora: Profa. Yusneyi Y. Carballo Barrera

Correo-e: yusneyi.carballo@ciens.ucv.ve

Fecha: Octubre de 2015

RESUMEN

El presente Trabajo Especial de Grado consiste en el desarrollo de una aplicación para gestionar los procesos asociados a la información de Preparadores y Auxiliares Docentes del Sistema de Gestión de la Planificación Docente del Departamento de la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela; y, de esta forma, automatizar los trámites administrativos y académicos en pro de ofrecer una mejor calidad de servicios.

Para la elaboración y ejecución de este módulo se combinaron las tecnologías PHP (lenguaje de programación), Laravel y AngularJS (*framework*), PostgreSQL (sistema manejador de base de datos) y la metodología Scrum modificada. Se escogieron estas tecnologías debido a su curva de aprendizaje empinada y a su extensa documentación y comunidad. Con respecto a la metodología Scrum, se usó por ser ágil y permitir realizar entregas parciales del resultado final del proyecto, priorizadas por el beneficio que aportan a los usuarios del proyecto desarrollado.

Palabras Clave: BPM, Automatización de Procesos Académicos, Gestión de Procesos, Planificación Docente, Departamento de Computación UCV.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO I. PLANTEAMIENTO DEL PROBLEMA	3
1.1 Planteamiento del problema.....	3
1.2 Situación actual	3
1.3 Objetivos del Trabajo Especial de Grado	4
1.3.1 Objetivo General.....	4
1.3.2 Objetivos Específicos.....	4
1.4 Alcance y restricciones	4
CAPÍTULO II. MARCO TEÓRICO	6
2.1 Antecedentes	6
2.2 Bases conceptuales.....	8
Aplicación web	8
Modelo Vista Controlador	8
<i>Framework</i>	9
Proceso de Negocio.....	10
<i>Business Process Management</i>	10
Base de Datos.....	10
Arquitectura Cliente Servidor	11
Interfaz de Programación de Aplicaciones	11
2.3 Herramientas tecnológicas	12
Lenguaje de Marcado de Hipertexto (HTML).....	12
Hojas de Estilo en Cascada (CSS)	12
Lenguaje de Hojas de Estilo SASS.....	12
JavaScript.....	13
<i>Bootstrap</i>	13
Preprocesador de Hipertexto (PHP).....	13
Laravel	14
AngularJS.....	15
PostgreSQL	15
Git	15
CAPÍTULO III. MARCO METODOLÓGICO	16
3.1 Metodologías de desarrollo de <i>software</i>	16
Características deseables de una metodología	16
3.2 Metodologías Vs. Ciclo de vida.....	17
3.3 Modelos de procesos en el desarrollo de <i>software</i>	20
3.4 Clasificación de las metodologías según el modelo de proceso	21
Modelos Convencionales o Prescriptivos de procesos	21
Modelo Cascada.....	22
Modelos de Procesos Incrementales	24

Modelo de Desarrollo Rápido de Aplicaciones (DRA)	25
Modelos Evolutivos	27
Modelos Iterativos	28
Modelos de Desarrollo Ágiles	29
3.5 Metodología de desarrollo	30
Visión general de Scrum	31
Características de Scrum:.....	32
El Equipo Scrum (<i>Scrum Team</i>)	35
El Dueño del Producto (<i>Product Owner</i>).....	35
El Equipo de Desarrollo (<i>Development Team</i>)	36
El Scrum <i>Master</i>	37
Eventos de Scrum	37
El <i>Sprint</i>	37
Objetivo del <i>Sprint</i> (<i>Sprint Goal</i>)	38
Artefactos de Scrum.....	38
Pila del producto	39
Pila del <i>Sprint</i>	40
3.6 ¿Por qué se seleccionó el marco de trabajo Scrum?	42
CAPÍTULO IV. DESARROLLO DE LA SOLUCIÓN	45
4.1 Objetivo General	45
4.2 Objetivos Específicos.....	45
4.3 Alcance de la aplicación	45
4.4 Plataforma utilizada en su desarrollo.....	46
4.5 Método de desarrollo utilizado y etapas seguidas para la creación de la aplicación	46
4.6 Descripción de la aplicación	47
Usuarios	47
Principales funcionalidades de la aplicación	52
Casos de uso más importantes	54
Principales interfaces de la aplicación	66
Pruebas funcionales de la aplicación	76
RESULTADOS DEL TRABAJO ESPECIAL DE GRADO.....	88
CONCLUSIONES Y RECOMENDACIONES	89
REFERENCIAS BIBLIOGRÁFICAS DIGITALES.....	91
ANEXOS	94

ÍNDICE DE FIGURAS

Figura 1 - Diagrama del Modelo Vista Controlador	9
Figura 2 - Ciclo de vida de un Software	20
Figura 3 - Ciclo de vida de un proyecto en las metodologías convencionales	22
Figura 4 - Modelo en Cascada	23
Figura 5 - Modelo Incremental	25
Figura 6 - Modelo de desarrollo rápido de aplicaciones (DRA).....	27
Figura 7 - Metodología Scrum.....	31
Figura 8 - Diagrama de secuencia - Ingresar usuario nuevo.....	50
Figura 9 - Diagrama de secuencia - Solicitud de concurso.....	51
Figura 11 - Diagrama de secuencia - Retirar preparador o auxiliar docente	52
Figura 12 - Diagrama de casos de uso - Nivel 0	54
Figura 13 - Diagrama de casos de uso - Nivel 1	55
Figura 14 - Diagrama de casos de uso - Nivel 2	56
Figura 15 - Principales interfaces - Inicio de Sesión	67
Figura 16 - Principales interfaces - Recuperación de contraseña	67
Figura 17 - Principales interfaces - Pantalla inicial	68
Figura 18 - Principales interfaces – Consulta y edición de datos personales	68
Figura 19 - Principales Interfaces - Concursos de Preparadores	69
Figura 20 - Principales Interfaces - Lista de observaciones	69
Figura 21 - Principales Interfaces - Lista de notificaciones.....	70
Figura 22 - Principales Interfaces - Modal de Solicitud de Concurso de Preparadores para Centros	70
Figura 23 - Principales Interfaces - Modal de Solicitud de Concurso de Preparadores para Materias.....	71
Figura 24 - Principales Interfaces - Descartar concurso de preparadores.....	71
Figura 25 - Principales Interfaces - Editar solicitud concurso de preparadores	72
Figura 26 - Principales Interfaces - Rechazar solicitud de concurso de preparadores.....	72
Figura 27 - Principales Interfaces - Resultados del concurso de preparadores.....	73
Figura 28 - Principales Interfaces - Gestión de preparadores	73
Figura 29 - Principales Interfaces - Registro de nuevo preparador. Paso 1: Ingreso de datos.....	74
Figura 30 - Principales Interfaces - Registro de nuevo preparador. Paso 2: Verificación de datos	74
Figura 31 - Principales Interfaces - Registro de nuevo preparador. Paso 3: Subida de archivos .	75
Figura 32 - Principales Interfaces - Registro de nuevo preparador. Paso 4: Confirmación.....	75
Figura 33: Diagrama BPM Gestión de Personal Preparador	94
Figura 34: Diagrama BPM Llenar Formulario de Coordinador de Materia	95
Figura 35: Diagrama BPM Llenar Formulario de Estudiante.....	96
Figura 36: Modelo de base de datos – Gestión de Personal Preparadores.....	97

ÍNDICE DE TABLAS

Tabla 1 - Tabla de perfiles y funcionalidades	49
Tabla 2 - Especificación de casos de uso - Solicitar concurso	57
Tabla 3 - Especificación de casos de uso - Aceptar concurso	58
Tabla 4 - Especificación de casos de uso - Rechazar concurso	59
Tabla 5 - Especificación de casos de uso - Editar concurso	60
Tabla 6 - Especificación de casos de uso - Descartar concurso.....	61
Tabla 7 - Especificación de casos de uso - Finalizar concurso	62
Tabla 8 - Especificación de casos de uso - Retirar preparador	63
Tabla 9 - Especificación de casos de uso - Ratificar preparador	64
Tabla 10 - Diagrama de casos de uso - Nivel 2	64
Tabla 11 - Especificación de casos de uso - Consultar recaudos.....	65
Tabla 12 - Especificación de casos de uso - Recuperar contraseña.....	66
Tabla 13 - Pruebas funcionales – Inicio de sesión	76
Tabla 14 - Pruebas funcionales - Coordinador de materia (solicitud de concurso)	78
Tabla 15 - Pruebas funcionales - Coordinador de centro (solicitud de concurso)	81
Tabla 16 - Pruebas funcionales - Coordinador de centro (edición de concurso)	84
Tabla 17 - Pruebas funcionales - Coordinador de materia (edición de concurso)	86
Tabla 18 - Pruebas funcionales - Usuario no autenticado - Registro de resultados de concursos - Paso 1	87

INTRODUCCIÓN

El mundo ha alcanzado un nivel de tecnología tal que requiere que cada ámbito de la vida esté al día con los adelantos que se están llevando a cabo. De esto no escapa el área académica. El Departamento de la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela, como baluarte de la educación, debe mantenerse actualizado para un desarrollo más eficaz de los servicios que ofrece tanto al personal docente y administrativo, como a la comunidad estudiantil.

Debido a la falta de automatización en los trámites realizados en el Departamento, surge el interés y la intención de innovar y desarrollar un sistema que permita la eficiencia en los servicios. El proyecto inició con una pasantía académica en el Departamento de Computación en donde se observó que la información de la programación docente principalmente se gestiona a través de varias hojas de cálculo, que especifican materias, docentes, preparadores, estimados de secciones y estudiantes, entre otros datos de interés.

Observando que el uso de múltiples hojas de cálculo y documentos de texto no es necesariamente la forma más conveniente de gestionar la planificación docente semestral, se propuso realizar una aplicación que le permita al Departamento de Computación, los docentes, preparadores, auxiliares docentes y personal administrativo procesar la información de una forma organizada y más centralizada y en el marco de una herramienta en donde se implementen los procesos de negocio y comunicación más importantes.

Para dar cumplimiento a los requerimientos identificados, se desarrolló una aplicación que combina las tecnologías PHP (lenguaje de programación), Laravel y AngularJS (*framework*), PostgreSQL (sistema manejador de base de datos) y la metodología Scrum modificada.

Este documento se encuentra estructurado en capítulos de la siguiente manera:

En el capítulo I, se describe la situación actual del Departamento en cuanto a automatización refiere, el planteamiento del problema, la justificación del problema y se indican, a su vez, el objetivo general y los objetivos específicos del Trabajo Especial de Grado.

En el capítulo II, se definen los conceptos asociados a la gestión de procesos y se describen las herramientas que se usarán para el modelado e implementación del módulo.

En el capítulo III, se describe la metodología de desarrollo de *software* utilizada.

En el capítulo IV, se desarrolla la propuesta de Trabajo Especial de Grado (TEG) y se detalla la solución planteada mediante los objetivos, alcance, descripción de la propuesta y el plan de cronograma de actividades para completar el Trabajo Especial de Grado.

Finalmente, se presentan las conclusiones del trabajo realizado, recomendaciones, las fuentes consultadas y anexos con como información complementaria.

CAPÍTULO I. PLANTEAMIENTO DEL PROBLEMA

1.1 Planteamiento del problema

Hoy en día, la tecnología abarca los espacios de la vida académica. Los procesos de programación docente: pensum, cargas horarias, inscripciones, etc., deberían manejarse a través de un sistema automatizado que mejore la gestión de los servicios.

El Departamento de la Escuela de Computación presenta una desactualización en la forma como ejecuta y ofrece dichos servicios. Se observa la necesidad de hacer mejoras en la gestión de algunos servicios y por eso en el marco de esta investigación se plantea la siguiente interrogante: ¿Cómo desarrollar una aplicación web que promueva la automatización de los servicios asociados a preparadores y auxiliares docentes de la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela? El módulo de Gestión de Trámites del Personal Docente Preparador y Auxiliar Docente, basándose en los procesos de negocios de la Escuela de Computación, permitiría ejecutar los procesos referentes a los trámites del personal docente preparador y auxiliar docente, a fin de proveer un servicio automatizado y eficaz.

El sistema que se desarrollará es una aplicación web que permitirá una administración centralizada de la Gestión de Preparadores y Auxiliares Docentes del Departamento. De esta manera, se podrá acceder al módulo de gestión de preparadores y auxiliares docente a través de un usuario y su respectiva contraseña.

1.2 Situación actual

A través de información suministrada por la Secretaria del Departamento de Computación, Sra. Gleidys Caraballo, y el Profesor Carlos Acosta, se pudo obtener una descripción más precisa de los servicios que se prestan y una versión inicial del diagrama de proceso de negocios. El manejo de los trámites de los preparadores y auxiliares docentes, pese a que algunos son digitales, carecen de automatización en un sistema o plataforma que evite ambigüedad o repeticiones en la información, disminuyendo así la confusión a la hora de realizar los concursos de acuerdo con los requerimientos de las materias y los recursos disponibles.

Por otro lado, los procesos actuales en el Departamento no favorecen el ahorro de recursos, debido a que para la realización de muchos de los trámites necesarios se requiere la impresión de formularios y planillas. De contar con una plataforma web, los procesos se realizarían minimizando o sin la necesidad de impresiones, pudiendo acceder a los servicios desde distintos dispositivos. Esto significaría una simplificación importante de pasos que el Departamento actualmente no posee.

1.3 Objetivos del Trabajo Especial de Grado

1.3.1 Objetivo General

Desarrollar el módulo de Gestión de Trámites del Personal Docente Preparador y Auxiliar Docente que, basándose en los procesos de negocios de la Escuela de Computación, permita ejecutar los procesos referentes a los trámites del personal preparador y auxiliar docente, a fin de proveer un servicio automatizado y eficaz.

1.3.2 Objetivos Específicos

1. Modelar los procesos de negocio realizados por la Escuela para su análisis e implementación.
2. Diseñar la estructura de la aplicación basándose en el patrón de diseño Modelo-Vista-Controlador (MVC).
3. Utilizar modelos de diseño web, estándares y lineamientos de usabilidad para lograr un diseño claro y limpio para la implementación del módulo.
4. Hacer uso de una metodología de desarrollo, que permita iterar sobre las distintas etapas del proceso desarrollo de la aplicación, con el fin de perfeccionar la aplicación a lo largo del ciclo de desarrollo.
5. Documentar el análisis de los procesos, así como la implementación de la aplicación.
6. Realizar pruebas de funcionalidades sobre la aplicación.

1.4 Alcance y restricciones

La aplicación web estará delimitada en implementar parte del servicio de Gestión de Personal Docente y Preparadores que presta el Departamento de la Escuela de

Computación, en específico, a la gestión de preparadores de pregrado. El servicio se encargará de:

- Gestionar concursos para preparadores y auxiliares docentes de los centros y materias pertenecientes a la Escuela de Computación.
- Gestionar la asignación del personal preparador y auxiliar docente.
- Manejar el proceso de postulación de alumnos o auxiliares docentes como personal preparador.
- Crear una interfaz, a través de la cual el personal preparador y auxiliar docente, que haya sido aceptado, pueda proveer los documentos necesarios para su registro.

CAPÍTULO II. MARCO TEÓRICO

2.1 Antecedentes

En lo referente a antecedentes de sistemas de gestión de procesos administrativos en la Escuela de Computación, destacan los TEG siguientes:

- ***“Prototipo de Sistema de Gestión de Preparadores de la Escuela de Computación, Módulos Gestión de Retiros y Administrativo”*** por las licenciadas Aguilar M. Iris H. y Bellet L. Isabelle M. y los tutores Prof. Marcel Castro y Prof. Robinson Rivas (2009) expone que “Actualmente los procesos concernientes a la gestión de preparadores y auxiliares docentes se ejecutan de forma manual lo que pudiera ocasionar: retardo en los tiempos de respuesta, pérdida de información, error en los datos, falta de información actualizada o histórica, requerida por algunas de las partes involucradas, como los preparadores, personal docente y administrativo. La Escuela de Computación cuenta con una infraestructura tecnológica capaz de soportar la implantación de un sistema de información, que opere en tiempo real y que sea manejado por los diferentes actores asociados a los procesos de la gestión de preparadores, permitiendo el seguimiento y control de los trámites a través del uso de la Internet.” y su objetivo general fue “Desarrollar los módulos de Gestión de Retiros y Administrativo del sistema de gestión de preparadores de la Escuela de Computación”.
- ***“Prototipo de Sistema de Gestión de Preparadores de la Escuela de Computación, Módulos Gestión de Concursos y Nombramientos”*** por los licenciados Gouveia C. Rosa M. y Ruiz B. Isaura L. y los tutores Prof. Marcel Castro y Prof. Robinson Rivas (2009) expone que “La gestión de preparadores es importante para la Facultad de Ciencias, por ello es pertinente la aplicación eficiente y eficaz de dicha gestión. En la actualidad, la ejecución de esta gestión se realiza de forma manual lo que pudiera ocasionar, en algunos casos, pérdida de los documentos que se generan durante el desarrollo de cada una de sus actividades. Es por ello que se planteó el desarrollo de un sistema de Información automatizado para la gestión de preparadores, manejado en tiempo real y por los diferentes actores asociados a dicha gestión”, y su objetivo general fue “Desarrollar los

módulos de gestión de concursos y nombramientos del sistema de gestión de preparadores de la Escuela de Computación, según el *workflow* propuesto para estos procesos”.

- ***“Automatización de los procesos asociados a la Planificación Docente en CONEST”*** por la licenciada Josefina J. Montilla M. y los tutores Prof. Jossie Zambrano y Prof. Sergio Rivas (2014) expone que “El inicio de un período académico en cualquier institución contempla un determinado número de procesos, los cuales se caracterizan por su complejidad y flujo de información, en este sentido la gestión académica es la más importante y para hacerla posible es necesario iniciar una Planificación Docente. De manera que la distribución y organización de los docentes, estudiantes, asignaturas y horarios recae sobre esta. Por ello es significativo disminuir el margen de error por la cantidad de personas involucradas, los tiempos de respuesta que implica hacerlo de forma manual, así como la cantidad de material y recursos físicos utilizados.”, y su objetivo general fue “Automatizar los procesos asociados a la Planificación Docente, con la finalidad de generar un impacto positivo en los tiempos de respuesta y reducir la brecha de comunicación entre las dependencias de las escuelas”.
- ***“Desarrollo de Módulos para la Gestión de Trámites Administrativos y Solicitudes de la Escuela de Computación de la Universidad Central de Venezuela”*** por las licenciadas Montenegro Z. Hailyx N. y Zapata B. Isthar L. Y. y la tutora Prof. Carballo B. Yusneyi Y (2014) expone que “La Escuela de Computación usa herramientas ofimáticas para la gestión de documentos de los trámites y solicitudes, pero la distribución de las actividades asociadas a estos se lleva a cabo de forma manual, creando la necesidad de innovar la forma en cómo se realizan estos procesos para hacerlos más eficientes”, y su objetivo general fue “Desarrollar un sistema compuesto por módulos, para facilitar la realización de tareas de forma automatizada, asociados a la gestión de trámites administrativos y solicitudes docentes”.
- ***“Automatización de procesos relacionados con las solicitudes estudiantiles y actividades administrativas y de docencia de la Facultad de Ciencias”*** por los

licenciados Boyer Yurbelis y Méndez Néstor y sus tutoras Prof. Zambrano Jossie y Prof. Di Vasta Concettina (2008) expone que “La División de Control de Estudios se encuentra automatizando sus procesos con el fin de ofrecer mejor servicio a sus usuarios. Realizaron la aplicación por docentes y estudiantes en calidad de desarrolladores de sistemas basados en plataforma de *software* libre”, y su objetivo general fue “Desarrollar una aplicación web que automatice el conjunto de procesos relacionados a las diversas solicitudes estudiantiles y actividades que realizan las escuelas y dependencias de la Facultad de Ciencias, en relación a la División de Control de Estudios, dando soporte de personal involucrado aumentando así su satisfacción y al mismo tiempo contribuir en la disminución de los costos y los tiempos de respuesta”.

2.2 Bases conceptuales

Aplicación web

De acuerdo con el portal web EcuRed (s.f), las aplicaciones web son aquellas en las que el usuario accede a un servidor web a través de Internet o de una intranet mediante un navegador. “Son aplicaciones (Software) que se codifican en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador”.

Modelo Vista Controlador

Según Jaramillo, Cardona & Villa (2008), el modelo vista controlador “es un patrón que permite separar la interfaz gráfica de usuario de los datos y de la lógica, apoyándose en tres componentes”, los cuales se muestran en la Figura 1. Asimismo, exponen dichos componentes:

- **Modelo:** es la representación de los datos y reglas de negocio. Es el encargado de manejar un registro de las vistas de los controladores que existen en el sistema.
- **Vista:** permite mostrar la información del modelo en un formato adecuado que permita que se dé la interacción, además de poseer un registro acerca del controlador asociado y brindar el servicio de actualización, que puede ser usado tanto por el controlador como por el modelo.

- **Controlador:** responde a los eventos provocados en el usuario, que implican cambios en el modelo y en la vista, dando una correcta gestión de las entradas del usuario.

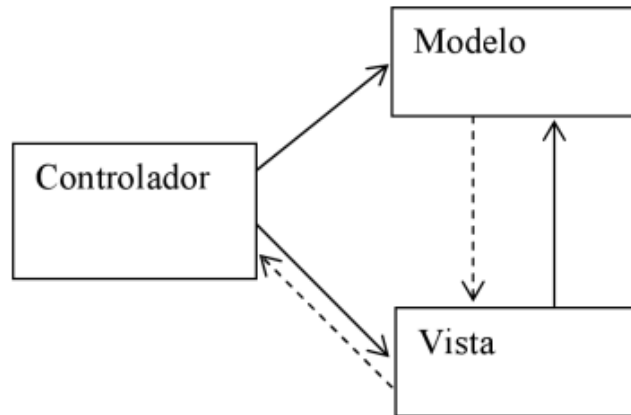


Figura 1. Diagrama del Modelo Vista Controlador
Fuente: Jaramillo, S., Cardona, S. & Villa, D. (2008)

Framework

En el desarrollo de *software*, un *framework* es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de *software* concretos, con base en que otro proyecto de *software* puede ser organizado y desarrollado. Puede incluir soporte de programas, bibliotecas y/o un lenguaje interpretado, entre otros, para ayudar a desarrollar y unir los diferentes componentes de un proyecto. (EcuRed, s.f).

Un *framework* para desarrollo de aplicaciones web es un *software* o conjunto de librerías, que está diseñado para dar soporte al desarrollo de sitios y en general, a la construcción de cualquier aplicación web. Este trata de facilitar aquellas actividades comunes realizadas durante el desarrollo de la aplicación, como por ejemplo: acceso a la base de datos, uso de plantillas y manejo de sesiones.

Muchos *frameworks* basan su estructura principal en el patrón de arquitectura MVC, con el fin de separar el modelo de datos, las entradas del usuario, y la interfaz gráfica en tres componentes que puedan evolucionar de manera independiente. Se toma como punto de partida para el *framework* esta arquitectura, debido a que es una metodología bien probada y utilizada ampliamente. Es orientada a objetos, lo que promueve la reutilización del código. (Mendoza, s.f)

Proceso de Negocio

Jeston & Nelis lo definen como “un conjunto de actividades que, impulsadas por eventos y ejecutándolas en cierta secuencia, crean un valor para un cliente, ya sea interno o externo”. (2008)

Business Process Management

El *Business Process Management* (BPM por sus siglas en inglés) o Manejo de Procesos de Negocios, se puede definir como el logro de los objetivos de una organización a través de la mejora, manejo y control de los procesos de negocios esenciales. (Jeston & Nelis, 2008)

Base de Datos

Según Cobo (2009), una base de datos “es un conjunto de datos almacenados con redundancias innecesarias en un soporte informativo y accesible simultáneamente por distintos usuarios y aplicaciones. Los datos deben estar estructurados y almacenados de una forma totalmente independiente de las aplicaciones que la utilizan”.

Llanos (2010) define la base de datos como “un conjunto, colección o depósito de datos almacenados en un soporte informático de acceso directo”. Asimismo, afirma: “Los datos deben estar relacionados y estructurados de acuerdo con un modelo capaz de recoger el contenido semántico de los datos almacenados. Dada la importancia que tienen en el mundo real las relaciones entre los datos, es imprescindible que la base de datos sea capaz de almacenar estas interrelaciones”. De acuerdo con este autor una base de datos debe cumplir una serie de requisitos:

En las bases de datos no debe existir una redundancia lógica de datos, aunque es admisible cierta redundancia física por motivos de eficiencia. Desde el punto de vista del usuario, los datos solo están almacenados una vez, aunque el sistema los puede replicar para facilitar su acceso de una manera más eficiente. Por tanto, un usuario actualiza un dato de forma única, y si existiera redundancia física del dato, el sistema se responsabilizará de efectuar todos aquellos cambios en los lugares en los que dicho dato estuviese replicado.

Las bases de datos han de dar soporte a múltiples usuarios y a diferentes aplicaciones simultáneamente. Debe existir una independencia tanto física como lógica entre datos y su proceso.

La definición y descripción del conjunto de datos contenidos en la base de datos deben ser únicas y estar integradas con los mismos datos, de manera que los datos almacenados estén auto-documentados, y cualquier cambio que se produzca en dicha documentación se ha de reflejar en la base de datos.

La base de datos debe asegurar la integridad, seguridad y confidencialidad de sus datos cuando estos se actualizan y recuperan (Llanos, 2010).

Arquitectura Cliente Servidor

“Una arquitectura es un conjunto de reglas, definiciones, términos y modelos que se emplean para producir un producto” (Llanos, 2010).

La arquitectura cliente-servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes, que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos, y servidor al proceso que responde a las solicitudes. Es el modelo de interacción más común entre aplicaciones en una red. (Llanos, 2010)

Interfaz de Programación de Aplicaciones

De acuerdo con Alegsa (2012), una interfaz de programación de aplicaciones (API por sus siglas en inglés *Application Programming Interface*) “es un grupo de rutinas que provee un sistema operativo, una aplicación o una biblioteca, que definen cómo invocar desde un programa un servicio que estos prestan. En otras palabras, una API representa una interfaz de comunicación entre componentes de *software*”.

“El *software* que provee la funcionalidad descrita por una API se dice que es una implementación del API”. (Alegsa, 2012)

2.3 Herramientas tecnológicas

Lenguaje de Marcado de Hipertexto (HTML)

World Wide Web Consortium (s.f) afirma que “HTML es el lenguaje utilizado para describir la estructura de páginas webs.” De igual forma, HTML provee los medios para:

- Publicar documentos en línea, con encabezados, textos, tablas, listas, fotos, etc.
- Recuperar información en línea vía enlaces de hipertexto.
- Diseñar formas para llevar a cabo transacciones con servicios remotos, para su uso en la búsqueda de información, reservaciones, órdenes de productos, etc.
- Incluir hojas de cálculo, videos, sonidos y otras aplicaciones directamente en sus documentos.

Con HTML, se describe la estructura de la página utilizando marcado de etiquetas. Los elementos del lenguaje contienen etiquetas como “párrafo”, “lista”, “tabla”, entre otros (W3C, s.f).

Hojas de Estilo en Cascada (CSS)

CSS es el lenguaje utilizado para describir la presentación de páginas web, incluyendo colores, posicionamiento y fuentes. Permite adaptar la presentación a distintos tipos de dispositivos, como pantallas grandes o pequeñas. La separación del HTML y el CSS hace que sea más fácil el mantenimiento de los sitios, la compartición de estilos entre páginas y la adaptación de las páginas a distintos ambientes. Esto es referido como la separación de la estructura de la presentación (Mozilla Developer Network, s.f).

Lenguaje de Hojas de Estilo SASS

De acuerdo con el significado disponible en su página web (s.f), Sass es una extensión de CSS que añade poder y elegancia a CSS. Permite el uso de variables, reglas anidadas, importaciones, todo esto manteniendo una sintaxis compatible con CSS. Sass ayuda a mantener grandes hojas de estilo bien organizadas.

JavaScript

Es un lenguaje ligero e interpretado, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, como es el caso de Node.js o Apache CouchDB. Es un lenguaje script multi-paradigma, basado en prototipos, es dinámico, soporta estilos de programación funcional, está orientado a objetos y es imperativo.

El JavaScript estándar es ECMAScript. A partir de 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1. Los navegadores más antiguos soportan al menos la versión 3. (Mozilla Developer, s.f)

Bootstrap

Según Solís (s.f), *Bootstrap*, es un *framework* originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una *Tablet* u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como diseño adaptativo.

El beneficio de usar el diseño adaptativo en un sitio web, es principalmente que el sitio web se adapta automáticamente al dispositivo desde donde se acceda. Lo que se usa con más frecuencia son los *media queries*, que es un módulo de CSS3 que permite la representación de contenido para adaptarse a condiciones como la resolución de la pantalla y si se trabajan las dimensiones del contenido en porcentajes, se puede tener una web muy fluida capaz de adaptarse a casi cualquier tamaño de forma automática. (Solis, s.f)

Preprocesador de Hipertexto (PHP)

De acuerdo con información suministrada por su página web (s.f), “es un lenguaje de programación de código abierto de propósito general ampliamente utilizado, que está específicamente adecuado para el desarrollo web y puede ser embebido en HTML”.

Lo que distingue a PHP de otros lenguajes, es que este se ejecuta en el servidor, generando una respuesta, que es enviada luego al cliente. El cliente recibe los resultados de la

ejecución del *script* sin conocer el código que fue ejecutado para la obtención de la respuesta. (PHP, s.f)

Laravel

Según información en el portal El maestro de la web, (s.f), Laravel es un *Framework* de PHP, desarrollado con la arquitectura MVC. Laravel maneja una sintaxis expresiva, elegante, con el objetivo de eliminar la molestia del desarrollo web facilitando las tareas comunes, como la autenticación, enrutamiento, sesiones y caché.

Con Laravel, como *framework* de base, en el desarrollo de proyectos web se puede obtener:

- Reducción de costos y tiempos en el desarrollo y mantenimiento.
- Curva de aprendizaje baja (en comparación con otros *framework* de PHP).
- Tiene una extensiva documentación y además posee una amplia comunidad y foros.
- Es modular y con una amplio sistemas de paquetes y drivers con el que se puede extender la funcionalidad de forma fácil, robusta y segura.
- Hace que el manejo de los datos en Laravel no sea complejo; mediante *Eloquent* (que es un ORM encargado de traducir la lógica de los objetos a la lógica relacional, y así interactuar con la base de datos), siendo compatible con la gran mayoría de las bases de datos de uso actual y facilitando la migración de los datos de una forma fácil y segura.
- Facilita el manejo de rutas en la aplicación como también la generación de *url* amigables y control de enlaces auto-actualizables lo que hace más fácil el mantenimiento de un sitio web.
- También cuenta con una herramienta de interfaces de líneas de comando llamada Artisan que permite programar tareas como por ejemplo ejecutar migraciones, pruebas programadas, etc. (El maestro de la web, s.f).

AngularJS

Es un *framework* de JavaScript, el cual facilita el desarrollo de aplicaciones web del lado de cliente. Permite utilizar HTML y extender su sintaxis para expresar los componentes de su aplicación claramente. AngularJS sincroniza automáticamente los datos de la interfaz de usuario con sus objetos JavaScript a través del enlace bidireccional de datos entre la vista y el controlador (GitHub, s.f).

PostgreSQL

Es un sistema de gestión de bases de datos objeto-relacional de código fuente disponible libremente y distribuido bajo licencia BSD, lo que permite su uso, redistribución y modificación con la única restricción de mantener el copyright del software a sus autores (PostgreSQL, s.f).

Git

Es un *software* de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente (Wikipedia, s.f).

CAPÍTULO III. MARCO METODOLÓGICO

3.1 Metodologías de desarrollo de *software*

Metodología de desarrollo de *software* se describe como el conjunto de herramientas, técnicas, procedimientos y soporte documental para el diseño de Sistemas de información.

En Ingeniería de *Software*, cuando se habla de desarrollo de *software* se habla de desarrollo de programas y, por lo tanto, se considera como una tarea de ingeniería, que se debe ejecutar en una serie de fases para obtener un programa que funcione de acuerdo con métodos ya establecidos en otras disciplinas. Las actividades que los ingenieros de *software* realizan se encuentran asociadas a un proceso de *software* donde intervienen diferentes elementos (fases, actividades, producto, roles, agentes), que permiten la definición del *software* a producir (producto), su desarrollo o el diseño, su validación tanto a lo interno (requerimientos específicos) como a lo externo (expectativas del cliente) y su evolución, donde se modifica para adaptarlo a los cambios.

Una metodología se basa en una combinación de los modelos de procesos genéricos para obtener como beneficio un *software* que solucione un problema. Adicionalmente, una metodología debería definir con precisión los artefactos, roles y actividades, junto con prácticas, técnicas recomendadas y guías de adaptación de la metodología al proyecto. Sin embargo, la complejidad del proceso de creación de *software* es netamente dependiente de la naturaleza del proyecto mismo, por lo que el escogimiento de la metodología estará acorde al nivel de aporte del proyecto, ya sea de pequeño, mediano o gran nivel. (Universidad de Oriente, 2014)

Características deseables de una metodología

- Existencia de reglas predefinidas, fases y subfases, tareas, productos intermedios, técnicas y herramientas tales que se amoldan a cualquier desarrollo.
- Cobertura total del ciclo de desarrollo.
- Verificaciones intermedias.
- Planificación y control.
- Comunicación efectiva.

- Utilización de un abanico amplio de proyectos.
- Fácil formación.
- Herramientas case.
- Actividades que mejoren el proceso de desarrollo.
- Soporte al mantenimiento. Por ejemplo: reingeniería.
- Soporte de la reutilización de *software*, no solo de código.
- Actualmente, se evitan métodos muy burocráticos o monolíticos. (Universidad de Oriente, 2014)

3.2 Metodologías Vs. Ciclo de vida

Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo, con el fin de formalizarla y optimizarla. Determina los pasos a seguir y cómo realizarlos para finalizar una tarea.

Aplicando esto a la Ingeniería de *Software*, se puede decir que una metodología:

- Optimiza el proceso y el producto *software*.
- Guía en la planificación y en el desarrollo del *software*.
- Define qué hacer, cómo y cuándo durante todo el desarrollo y mantenimiento de un proyecto.

Una metodología define una estrategia global para enfrentarse al proyecto. Entre los elementos que forman parte de una metodología se encuentran:

- Fases: tareas a realizar en cada fase.
- Productos: E/S de cada fase, documentos.
- Procedimientos y herramientas: apoyo a la realización de cada tarea.
- Criterios de evaluación: del proceso y del producto. Saber si se han logrado los objetivos.

El ciclo de vida es el conjunto de fases por las que pasa el sistema que se está desarrollando desde que nace la idea inicial hasta que el *software* es retirado o reemplazado (muere).

Entre las funciones que debe tener un ciclo de vida se pueden destacar:

- Determinar el orden de las fases del proceso de *software*.
- Establecer los criterios de transición para pasar de una fase a la siguiente.
- Definir las entradas y salidas de cada fase.
- Describir los estados por los que pasa el producto.
- Describir las actividades a realizar para transformar el producto.
- Definir un esquema que sirve como base para planificar, organizar, coordinar, desarrollar, entre otros.

Las etapas de un ciclo de vida de un *software* son:

- Inicio: en esta etapa se definen los objetivos del proyecto y los recursos necesarios para su ejecución. Se explica el objetivo que se desea alcanzar, pero no la manera en la cual se cumplirá. Las características implícitas o explícitas de cada proyecto hacen necesaria una etapa previa destinada a determinar el objetivo.
- Planificación: se genera un planeamiento detallado que guíe la gestión del proyecto, temporal y económicamente.
- Implementación: se acordará el conjunto de actividades que componen la realización del producto.
- Puesta en producción: el proyecto entra en la etapa de definición, donde se le presentará al cliente o usuario final conscientes de su funcionamiento correcto y cumplimiento de los requerimientos solicitados. Esta etapa es de relevancia no solo porque significa la aceptación del proyecto por parte del cliente o usuario final, sino por las múltiples dificultades que suelen presentarse en la práctica, que llevan a un alargamiento del proyecto, provocando costos no previstos.
- Control en producción: es el control del producto, analizando cómo el proceso difiere o no de los requerimientos originales e iniciando las acciones correctivas de ser necesarias.

Cuando se dice que hay que corregir el producto, se hace referencia a pequeñas desviaciones de los requerimientos originales que puedan llegar a surgir en el ambiente productivo.

Objetivos de cada etapa:

- **Expresión de necesidades:** elaborar un documento en el cual se reflejen los requerimientos y funcionalidades que ofrecerá al usuario el sistema a implementar (qué, y no cómo, se va a implementar).
- **Especificaciones:** formalizar los requerimientos; el documento obtenido en la etapa anterior se tomará como punto de partida para esta etapa.
- **Análisis:** determinar los elementos que intervienen en el sistema a desarrollar, su estructura, relaciones, evolución temporal, funcionalidades. Se describirá claramente qué producto se construirá, qué funcionalidades aportará y qué comportamiento desempeñará.
- **Diseño:** definir el cómo se debe hacer (¿cómo debe ser construido el sistema en cuestión?). Para eso se expone en detalle entidades y relaciones de las bases de datos, se selecciona el lenguaje que se va a utilizar, el Sistema Gestor de Bases de Datos, entre otros).
- **Implementación:** iniciar la codificación de algoritmos y estructuras de datos, definidos en las etapas anteriores, en el correspondiente lenguaje de programación o para un determinado sistema gestor de bases de datos. En muchos proyectos se pasa directamente a esta etapa; son proyectos muy arriesgados que adoptan un modelo de ciclo de vida de *code & fix* (codificar y corregir), en el que se eliminan las etapas de especificaciones, análisis y diseño con la consiguiente pérdida de control sobre la gestión del proyecto.
- **Depuración (Debugging):** garantizar que el programa no contenga errores de diseño o codificación. En esta etapa no se desea saber si el programa realiza lo que solicitó el usuario, esa tarea le corresponde a la etapa de implementación. En esta se persigue como objetivo encontrar la mayor cantidad de errores. Todos los programas contienen errores: encontrarlos es cuestión de tiempo. Lo ideal es encontrar la mayoría, sino todos, en esta etapa. También se pueden agregar pruebas de rendimiento.
- **Validación:** verificar que el sistema desarrollado cumpla con los requerimientos expresados inicialmente por el cliente y que han dado lugar al presente proyecto. En

muchos proyectos, las etapas de validación y *debugging* se realizan en paralelo por la estrecha relación que llevan. Sin embargo, se debe evitar la confusión: se pueden realizar en paralelo, pero no como una única etapa.

- **Evolución:** en la mayoría de los proyectos se considera esta etapa como mantenimiento y evolución, y se le asigna, no sólo el agregado de nuevas funcionalidades (evolución), sino la corrección de errores que surgen (mantenimiento). En la práctica, esta denominación no es del todo errónea, debido a que es posible que aún luego de una etapa de *debugging* y validación exhaustiva, se filtren errores. (Universidad de Oriente, 2014)



Figura 2. Ciclo de vida de un Software
Fuente: Adaptado de Universidad de Oriente (2014)

3.3 Modelos de procesos en el desarrollo de *software*

Un modelo de proceso para el desarrollo de *software* es una representación simplificada de pasos, expuesta desde una perspectiva específica. Por su naturaleza de simplificación, un modelo de procesos del *software* es una abstracción de un proceso real.

Estos modelos tienen como propósito la producción eficaz y eficiente de un producto *software* que reúna los requisitos del cliente. Este proceso es intensamente intelectual, afectado

por la creatividad y juicio de las personas involucradas. La mayoría de los modelos de procesos de desarrollo del *software* son dirigidos por el tiempo; cuanto más tarde sea, más atrás se encontrará en el proceso de desarrollo. Como todo proceso, está constituido por pasos o fases que contienen a su vez actividades. Estos modelos de desarrollo de *software* se basan en un ciclo de vida para desarrollarse como lo son:

- La necesidad de solucionar un problema (surgimiento de necesidades)
- Inicio del proceso (desarrollo); dentro de esta fase se encuentra la definición del proyecto, el análisis del contexto, determinación de requerimientos, diseño del sistema, construcción del sistema, pruebas e implantación.
- Operación y mantenimiento, en el que se realizan ajustes y se buscan fallas.
- Renovación o extinción.

Los procesos de *software* son complejos, debido a que un producto de *software* es intangible y por lo general muy abstracto; esto dificulta la definición del producto y sus requisitos, sobre todo cuando no se tienen precedentes en productos *software* similar. En general, este producto está compuesto por hardware y *software*. En cuanto al hardware, su producción se realiza sistemáticamente y la base de conocimiento para el desarrollo de dicha actividad está claramente definida. Respecto del *software*, su construcción y resultados han sido históricamente cuestionados debido a los problemas asociados. (Universidad de Oriente, 2014)

3.4 Clasificación de las metodologías según el modelo de proceso

Modelos Convencionales o Prescriptivos de procesos

Los modelos convencionales o modelos prescriptivos de procesos permiten llenar el marco de trabajo con un conjunto de tareas orientadas al desarrollo de un *software*.

Se les llama "prescriptivos" porque prescriben un conjunto de elementos del proceso, tales como:

- Actividades del Marco de Trabajo.
- Acciones de la Ingeniería del *Software*.
- Tareas.

- Productos de trabajo.
- Aseguramiento de la calidad.
- Mecanismos de control del cambio para cada proyecto. (Universidad de Oriente, 2014)

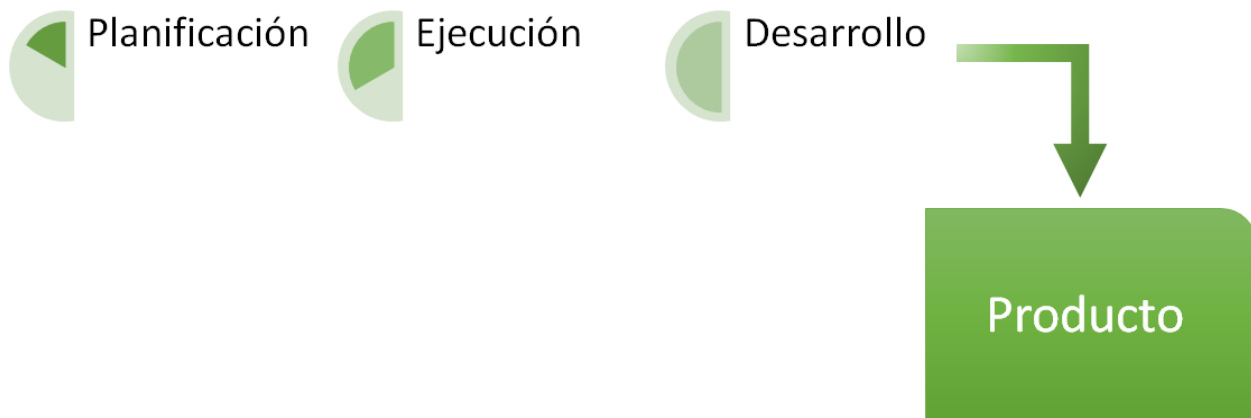


Figura 3. Ciclo de vida de un proyecto en las metodologías convencionales
Fuente: Adaptado de Universidad de Oriente (2014)

Modelo Cascada

El modelo en cascada, algunas veces llamado el ciclo de vida clásico, sugiere un enfoque sistemático, secuencial hacia el desarrollo del *software*, que se inicia con la especificación de requerimientos del cliente y que continúa con la planeación, el modelado, la construcción y el despliegue para culminar en el soporte del *software* terminado.

En la actualidad, el trabajo del *software* está acelerado y sujeto a una cadena infinita de cambios (de características, funciones y contenido de la información). Con frecuencia, el modelo en cascada no es apropiado para dicho trabajo. Sin embargo, puede servir como un modelo de proceso útil en situaciones donde los requerimientos están fijos y donde el trabajo se realiza, hasta su conclusión, de una manera lineal. (Softeng, 2014)

Las principales etapas de este modelo son:

- **Análisis y definición de requerimientos.** Los servicios, restricciones y metas del sistema se determinan a partir de las consultas con los usuarios. Se define una especificación del sistema.
- **Diseño del sistema y del software.** El proceso de diseño del sistema divide los requerimientos en sistemas hardware o *software*. Establece una arquitectura completa del sistema. El diseño del *software* identifica y describe las abstracciones fundamentales del sistema *software* y sus relaciones.
- **Implementación y prueba de unidades.** Durante esta etapa, el diseño del *software* se lleva a cabo como un conjunto o unidades de programas.
- **Integración y prueba del sistema.** Los programas o las unidades individuales de programas se integran y prueban como un sistema completo para asegurar que se cumplan los requerimientos del *software*.
- **Funcionamiento y mantenimiento.** El sistema se instala y se pone en funcionamiento práctico. El mantenimiento implica corregir errores no descubiertos en las etapas anteriores del ciclo de vida. (Softeng, 2014)

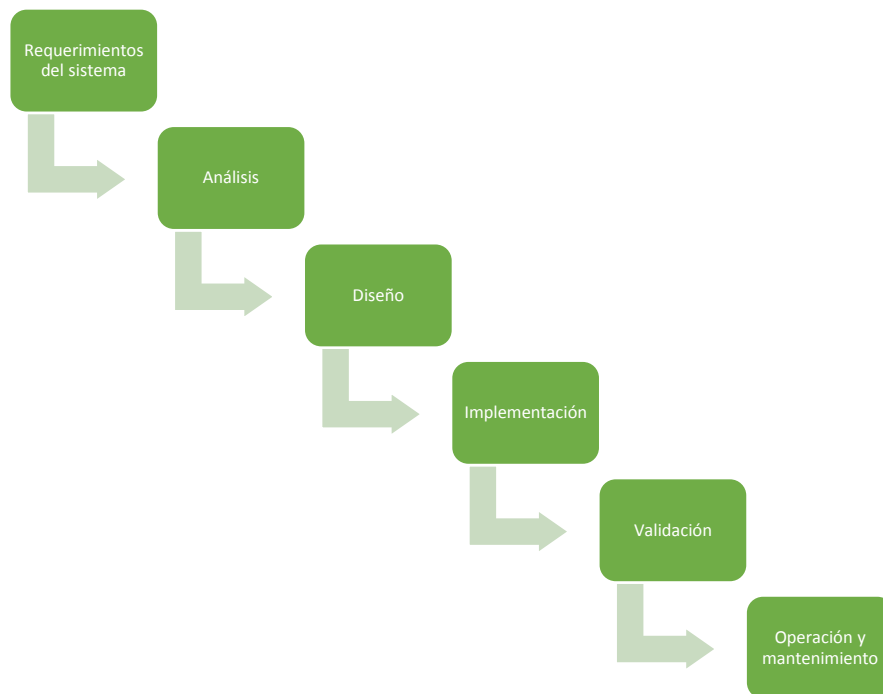


Figura 4. Modelo en Cascada
Fuente: Adaptado de Universidad de Oriente (2014)

Modelos de Procesos Incrementales

El modelo incremental combina elementos del modelo en cascada aplicado en forma iterativa. El modelo incremental aplica secuencias lineales de manera escalonada conforme avanza el tiempo en el calendario. Cada secuencia lineal produce "incrementos" del *software*. Por ejemplo, el *software* procesador de texto, desarrollado con este paradigma en su primer incremento, podría realizar funciones básicas de administración de archivos, edición y producción de documentos; en el segundo incremento, ediciones más sofisticadas, y tendría funciones más complejas de producción de documentos; en el tercer incremento, funciones de corrección ortográfica y gramatical; y en el cuarto, capacidades avanzadas de configuración de página. Se debe tener en cuenta que el flujo del proceso de cualquier incremento puede incorporar el paradigma de construcción de prototipos que se expone más adelante.

A menudo, al utilizar un modelo incremental, el primer incremento es un producto esencial, es decir, se incorporan los requisitos básicos, pero muchas características suplementarias (algunas conocidas, otras no) no se incorporan. El producto esencial queda en manos del cliente (o se somete a una evaluación detallada). Como resultado de la evaluación, se desarrolla un plan para el incremento siguiente. El plan afronta la modificación del producto esencial con el fin de satisfacer de mejor manera las necesidades del cliente y la entrega de características y funcionalidades adicionales. Este proceso se repite después de la entrega de cada incremento mientras no se haya elaborado el producto completo.

El modelo de proceso incremental, al igual que la construcción de prototipos y otros enfoques evolutivos, es iterativo por naturaleza. A diferencia de la construcción de prototipos, el modelo incremental se enfoca en la entrega de un producto operacional con cada incremento. Los primeros incrementos son versiones "incompletas del producto final", pero proporcionan al usuario la funcionalidad que necesita y una plataforma para evaluarlo. (Universidad de Oriente, 2014)

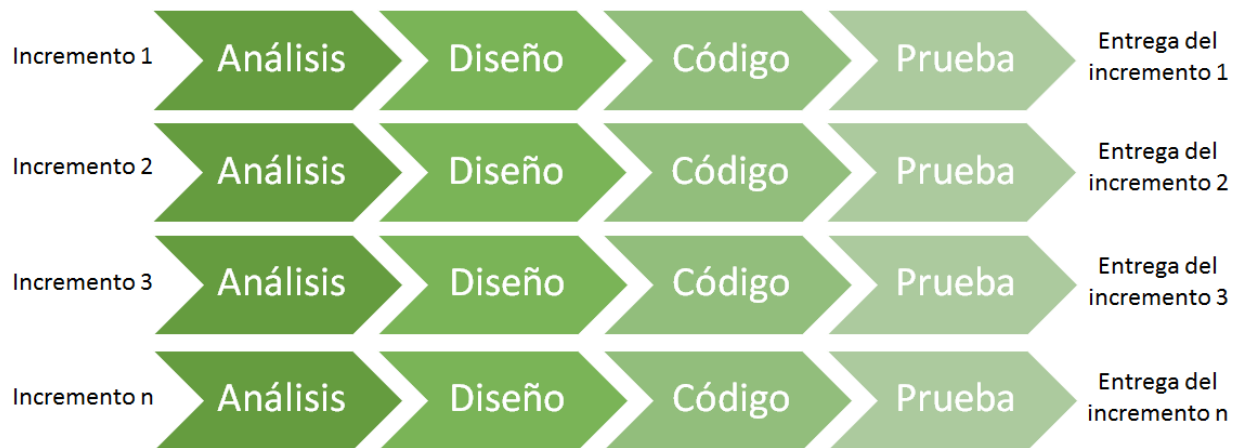


Figura 5. Modelo Incremental
Fuente: Adaptado de Universidad de Oriente (2014)

Modelo de Desarrollo Rápido de Aplicaciones (DRA)

El desarrollo rápido de aplicaciones (DRA) es un modelo de proceso de *software* incremental que resalta un ciclo de desarrollo corto. El modelo DRA es una adaptación a "alta velocidad" del modelo en cascada en el que se logra el desarrollo rápido mediante un enfoque de construcción basado en componentes. Si se entienden bien los requisitos y se limita el ámbito del proyecto, el proceso DRA permite que un equipo de desarrollo cree un "sistema completamente funcional" dentro de un periodo muy corto (por ejemplo, de 60 a 90 días).

Como otros modelos de proceso, el enfoque DRA cumple con las actividades genéricas del marco de trabajo que ya se han presentado. La comunicación trabaja para entender el problema de negocios y las características de información que debe incluir el *software*. La planeación es esencial porque varios equipos de *software* trabajan en paralelo sobre diferentes funciones del sistema. El modelado incluye tres grandes fases — modelado de negocios, modelado de datos y modelado del proceso — y establece representaciones del diseño que sirven como base para la actividad de construcción del modelo DRA. La construcción resalta el empleo de componentes de *software* existentes y la aplicación de la generación automática de código. Por último, el despliegue establece una base para las iteraciones subsecuentes, si estas son necesarias.

Es un modelo de proceso del *software* incremental que resulta un ciclo de desarrollo corto. El modelo DRA es una adaptación a alta velocidad del modelo en cascada, en el que se logra el desarrollo rápido mediante un enfoque de construcción basado en componentes. Hace un uso intensivo de componentes reusables de *software* con un ciclo de desarrollo extremadamente corto. (Universidad de Oriente, 2014)



Figura 6 - Modelo de desarrollo rápido de aplicaciones (DRA)
Fuente: Adaptado de Universidad de Oriente (2014)

Modelos Evolutivos

Se reconoce que el *software*, al igual que todos los sistemas complejos, evoluciona con el tiempo; los requisitos de gestión y de producto cambian a menudo conforme a que el desarrollo proceda, haciendo que el camino que lleva al producto final no sea real.

El desarrollo evolutivo consta del desarrollo de una versión inicial que, luego de exponerse, se va refinando de acuerdo con los comentarios o nuevos requerimientos por parte del cliente o del usuario final. Los modelos evolutivos son iterativos, se caracterizan por la forma en que permiten a los ingenieros en *software* desarrollar versiones cada vez más completas del *software*. A continuación se presentan algunos de los modelos que se clasifican en esta categoría.

- Construcción de prototipos
- Modelos en espiral
- Modelo de desarrollo concurrente (Universidad de Oriente, 2014)

Modelos Iterativos

Este modelo busca reducir el riesgo que surge entre las necesidades del usuario y el producto final por malos entendidos durante la etapa de recogida de requisitos. Consiste en la iteración de varios ciclos de vida en cascada. Al final de cada iteración se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto. El cliente es quien, después de cada iteración, evalúa el producto y lo corrige o propone mejoras. Estas iteraciones se repetirán hasta obtener un producto que satisfaga las necesidades del cliente.

El modelo iterativo se suele utilizar en proyectos en los cuales los requisitos no están claros por parte del usuario, por lo que se hace necesaria la creación de distintos prototipos para presentarlos y conseguir la conformidad del cliente. Cada iteración es un mini proyecto en cascada autocontenido compuesto de actividades como análisis de requerimientos, diseño, programación y pruebas.

Ventajas:

- Permite crear cada vez versiones más completas de *software*.
- Resolución de problemas de alto riesgo en tiempos tempranos del proyecto.
- Visión de avance en el desarrollo desde las etapas iniciales del desarrollo.
- Menor tasa de fallo del proyecto, mejor productividad del equipo, y menor cantidad de defectos.
- El trabajo iterativo deja una experiencia en el equipo que permite ir ajustando y mejorando las planificaciones, logrando menores desvíos en la duración total del proyecto.

Desventajas:

- Requiere de un cliente involucrado durante todo el curso del proyecto. Hay clientes que no están dispuestos a invertir el tiempo necesario.
- Infunde responsabilidad en el equipo de desarrollo al trabajar directamente con el cliente, requiriendo de profesionales sobre el promedio. (ScrumManager, 2014)

Modelos de Desarrollo Ágiles

Las metodologías ágiles son un conjunto de métodos de Ingeniería del *Software*, que se basan en el desarrollo iterativo e incremental, teniendo presente los cambios y respondiendo a estos mediante la colaboración de un grupo de desarrolladores auto-organizados y multidisciplinarios.

En las metodologías ágiles, los procesos se desarrollan de manera solapada, donde el ciclo de vida del proyecto es cíclico. La diferencia en el ciclo de vida de un proyecto ágil, en comparación con uno tradicional, es la forma en la que la agilidad solapa los procesos de manera iterativa.

La tendencia del control de procesos para desarrollo de *software* ha traído como resultado que proyectos no resulten exitosos debido al cambiante contexto que existe; las metodologías ágiles pretenden resolver este inconveniente construyendo soluciones a la medida, asegurando la calidad. Los métodos ágiles fueron pensados especialmente para equipos de desarrollo pequeños, con plazos reducidos, requisitos volátiles y nuevas tecnologías. La filosofía de las metodologías ágiles, pretenden dar mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del *software* con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo, pero manteniendo una alta calidad.

Estos valores inspiraron los doce principios del Manifiesto Ágil:

1. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de *software* que le aporte un valor.
2. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
3. Entregar frecuentemente *software* que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
4. Los dueños del proyecto y los desarrolladores deben trabajar juntos a lo largo del proyecto.

5. Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
6. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
7. El *software* que funciona es la medida principal de progreso.
8. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
9. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
10. La simplicidad es esencial.
11. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
12. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo y, según esto, ajusta su comportamiento.

Las características esenciales del proceso de desarrollo ágil para la mayoría de los proyectos son:

- Busca la satisfacción del cliente.
- Entrega temprana de *software* incremental.
- Utilizan métodos informales.
- Simplicidad general del desarrollo.
- La comunicación entre los desarrolladores y los clientes durante el desarrollo del proyecto debe ser activa y continua.
- La idea es que se mantenga un canal de retroalimentación con el cliente, a través de entregas de prototipos ejecutables o porción de un sistema operacional en periodos cortos para que la adaptabilidad mantenga un buen ritmo con el cambio. (Universidad de Oriente, 2014)

3.5 Metodología de desarrollo

Scrum es una metodología ágil y flexible para gestionar el desarrollo de *software*, cuyo principal objetivo es maximizar el retorno de la inversión para su empresa (ROI). Se basa en

construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, auto-gestión e innovación. (Softeng, 2014)

Scrum es una metodología ágil de gestión de proyectos, que busca elevar al máximo la productividad de un equipo; fue desarrollado por Jeff Sutherland y elaborado más formalmente por Ken Schwaber. Se enfoca en el hecho de que procesos definidos y repetibles solo funcionan para atacar problemas definidos y repetibles con actores definidos y repetibles en ambientes definidos y repetibles. Se divide un proyecto en iteraciones (que ellos llaman carreras cortas) de 30 días. La literatura de Scrum se orienta principalmente en la planeación iterativa y el seguimiento del proceso. (Universidad de Oriente, 2014)

Visión general de Scrum

Scrum: es un marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente. Según Schwaber & Sutherland Scrum es:

- Ligero.
- Fácil de entender.
- Difícil de llegar a dominar.

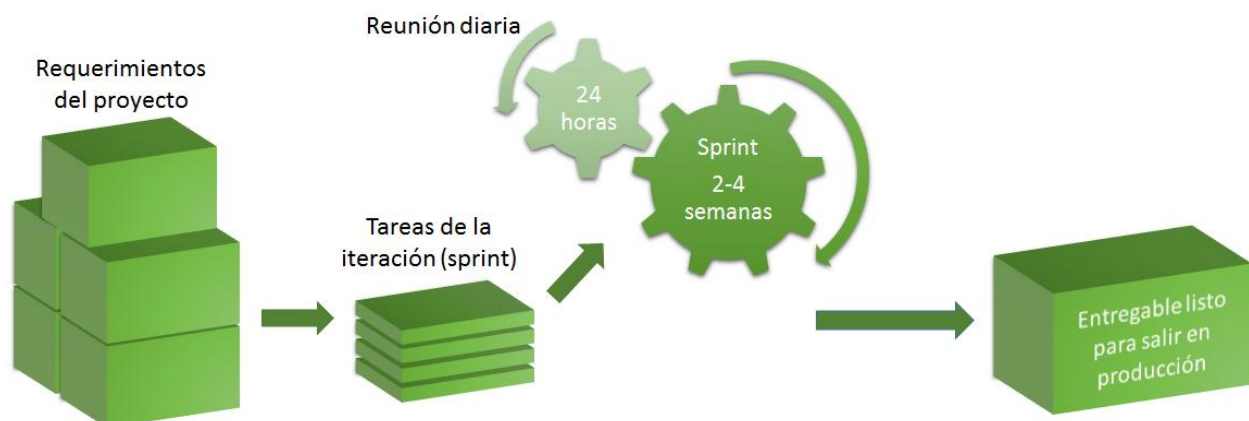


Figura 7. Metodología Scrum
Fuente: Adaptado de Universidad de Oriente (2014)

Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. No es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que se pueda mejorar.

El marco de trabajo Scrum consiste en los equipos Scrum, roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso.

Las reglas de Scrum relacionan los eventos, roles y artefactos, gobernando las relaciones e interacciones entre ellos. Las reglas de Scrum se describen en el presente documento.

Las estrategias específicas para usar el marco de trabajo Scrum son diversas y están descritas en otros lugares. (Schwaber & Sutherland, 2013)

Características de Scrum:

- Enfatiza valores y prácticas de gestión sin pronunciarse sobre requerimientos, prácticas de desarrollo, implementación y demás cuestiones técnicas.
- Hace uso de equipos auto-dirigidos y auto-organizados.
- Puede ser aplicado teóricamente a cualquier contexto en donde un grupo de gente necesita trabajar junta para lograr una meta común.
- Desarrollo de *software* iterativo incremental basado en prácticas ágiles.
- Iteraciones de treinta días; aunque se pueden realizar con más frecuencia estas iteraciones conocidas como *Sprint*.
- Dentro de cada *Sprint* se denomina al Scrum Master al Líder de Proyecto quien llevará a cabo la gestión de la iteración.
- Se convoca diariamente un “Scrum Daily Meeting”, el cual representa una reunión de avance diario de no más de 15 minutos con el propósito de tener retroalimentación sobre las tareas de los recursos y los obstáculos que se presentan. En esta categoría se responden preguntas como: ¿Qué se ha hecho desde el último encuentro? ¿Qué

obstáculos hay para cumplir la meta? ¿Qué se hará antes del próximo encuentro?.
(Universidad de Oriente, 2014)

Teoría de Scrum

Scrum se basa en la teoría de control de procesos de forma empírica. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo.

Tres pilares soportan toda la implementación del control de procesos empírico: transparencia, inspección y adaptación.

Transparencia

Los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado. La transparencia requiere que dichos aspectos sean definidos por un estándar común, de tal modo que los observadores compartan un entendimiento común de lo que se está observando.

Por ejemplo:

- Todos los participantes deben compartir un lenguaje común para referirse al proceso; y,
- Aquellos que desempeñan el trabajo y aceptan el producto de dicho trabajo deben compartir una definición común de “Terminado”.

Definición de “Terminado” (*Definition of “Done”*)

Cuando un elemento de la Lista de Producto o un Incremento se describe como “Terminado”, todo el mundo debe entender lo que engloba esta denominación. Aunque esto varía significativamente para cada Equipo Scrum, los miembros del Equipo deben tener un entendimiento compartido de lo que significa que el trabajo esté completado, para asegurar la transparencia. Esta es la definición de “Terminado” para el Equipo Scrum y se utiliza para evaluar cuándo se ha completado el trabajo sobre el Incremento de producto.

Esta misma definición guía al Equipo de Desarrollo en saber cuántos elementos de la lista de producto pueden seleccionarse durante una reunión de Planificación de *Sprint*. El propósito de cada *Sprint* es entregar incrementos de funcionalidad que potencialmente se puedan poner en producción, y que se ajustan a la definición de “Terminado” actual del Equipo Scrum.

Los Equipos de Desarrollo entregan un incremento de funcionalidad de producto en cada *Sprint*. Este incremento es utilizable, de modo que el dueño de producto podría elegir liberarlo inmediatamente. Si la definición de “Terminado” para un incremento es parte de las convenciones, estándares o guías de la organización de desarrollo, al menos todos los Equipos Scrum deben seguirla. Si “Terminado” para un incremento no es una convención de la organización de desarrollo, el Equipo de Desarrollo del Equipo Scrum debe definir una denominación de “Terminado” apropiada para el producto. Si hay múltiples Equipos Scrum trabajando en la entrega del sistema o producto, los equipos de desarrolladores en todos los Equipos Scrum deben definir en conjunto la definición de “Terminado”.

Cada incremento se integra con todos los incrementos anteriores y es probado exhaustivamente, asegurando que todos los Incrementos funcionan en conjunto. A medida que los Equipos Scrum maduran, se espera que su definición de “Terminado” se amplíe para incluir criterios más rigurosos que garanticen una mayor calidad. Cualquier producto o sistema debería tener una definición de “Terminado” que es un estándar para cualquier trabajo realizado sobre él.

Inspección: Los usuarios de Scrum deben inspeccionar frecuentemente los artefactos de Scrum y el progreso hacia un objetivo, para detectar variaciones. Su inspección no debe ser tan frecuente como para que interfiera en el trabajo. Las inspecciones son más beneficiosas cuando se realizan de forma diligente por inspectores expertos, en el mismo lugar de trabajo.

Adaptación: Si un inspector determina que uno o más aspectos de un proceso se desvían de límites aceptables, y que el producto resultante no será aceptable, el proceso o el material que está siendo procesado deben ser ajustados. Dicho ajuste debe realizarse cuanto antes para minimizar desviaciones mayores.

Scrum prescribe cuatro eventos formales, contenidos dentro del *Sprint*, para la inspección y adaptación, tal y como se describen en la sección Eventos de Scrum del presente documento.

- Reunión de Planificación del *Sprint* (*Sprint Planning Meeting*).
- Scrum Diario (*Daily Scrum*).
- Revisión del *Sprint* (*Sprint Review*).
- Retrospectiva del *Sprint* (*Sprint Retrospective*). (Softeng, 2014)

El Equipo Scrum (*Scrum Team*)

El Equipo Scrum consiste en un Dueño de Producto (*Product Owner*), el Equipo de Desarrollo (*Development Team*) y un Scrum Master. Los equipos Scrum son autoorganizados y multifuncionales. Los equipos autoorganizados eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas externas al equipo. Los equipos multifuncionales tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad.

Los equipos Scrum entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener retroalimentación. Las entregas incrementales de producto “Terminado” aseguran que siempre estará disponible una versión potencialmente útil y funcional del producto. (Softeng, 2014)

El Dueño del Producto (*Product Owner*)

El Dueño de Producto es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. El cómo se lleva a cabo esto podría variar ampliamente entre distintas organizaciones, Equipos Scrum e individuos.

El Dueño de Producto es la única persona responsable de gestionar la Lista del Producto (*Product Backlog*). La gestión de la Lista del Producto incluye:

- Expresar claramente los elementos que la componen.
- Ordenar sus elementos para alcanzar los objetivos y misiones de la mejor manera posible.
- Optimizar el valor del trabajo desempeñado por el Equipo de Desarrollo.

- Asegurar que la Lista del Producto es visible, transparente y clara para todos, y que muestra aquello en lo que el equipo trabajará a continuación.
- Asegurar que el Equipo de Desarrollo entiende los elementos de la Lista del Producto al nivel necesario. (Schwaber & Sutherland, 2013)

El Equipo de Desarrollo (*Development Team*)

El Equipo de Desarrollo consiste en los profesionales que desempeñan el trabajo de entregar un incremento de producto “Terminado”, que potencialmente se pueda poner en producción al final de cada *Sprint*. Solo los miembros del Equipo de Desarrollo participan en la creación del incremento.

Los Equipos de Desarrollo son estructurados y empoderados por la organización para emprender y gestionar su propio trabajo. La sinergia resultante optimiza la eficiencia y efectividad del Equipo de Desarrollo.

Los Equipos de Desarrollo tienen las siguientes características:

- Son autoorganizados. Nadie (ni siquiera el Scrum Master) indica al Equipo de Desarrollo cómo convertir elementos de la Lista del Producto en incrementos de funcionalidad potencialmente despleables;
- Los Equipos de Desarrollo son multifuncionales, contando como equipo con todas las habilidades necesarias para crear un incremento de producto;
- Scrum no reconoce títulos para los miembros de un Equipo de Desarrollo, todos son desarrolladores, independientemente del trabajo que realice cada persona; no hay excepciones a esta regla;
- Scrum no reconoce sub-equipos en los equipos de desarrollo, no importan los dominios particulares que requieran ser tenidos en cuenta, como pruebas o análisis de negocio; y,
- Los Miembros individuales del Equipo de Desarrollo pueden tener habilidades especializadas y áreas en las que estén más enfocados, pero la responsabilidad recae en el Equipo de Desarrollo como un todo. (Schwaber & Sutherland, 2013)

El Scrum Master

El Scrum Master es el responsable de asegurar que Scrum es entendido y adoptado. Los Scrum Masters hacen esto ratificando que el Equipo Scrum trabaje ajustándose a la teoría, prácticas y reglas de Scrum.

El Scrum Master es un líder que está al servicio del Equipo Scrum; ayuda a las personas externas al Equipo Scrum a entender cuáles interacciones con el Equipo Scrum pueden ser de ayuda y cuáles no. El Scrum Master permite que todos puedan modificar estas interacciones para maximizar el valor creado por el Equipo Scrum. (Schwaber & Sutherland, 2013)

Eventos de Scrum

En Scrum existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum. Todos los eventos son bloques de tiempo (*time-boxes*), de modo que todos tienen una duración máxima. Una vez que comienza un *Sprint*, su duración es fija y no puede acortarse o alargarse. Los demás eventos pueden terminar siempre que se alcance el objetivo del mismo, asegurando que se emplee una cantidad apropiada de tiempo evitando el desperdicio en el proceso.

Además del propio *Sprint*, que es un contenedor del resto de eventos, cada uno de los eventos de Scrum constituye una oportunidad formal para la inspección y adaptación de algún aspecto. Estos eventos están diseñados específicamente para habilitar la transparencia y la inspección. La falta de alguno de estos da como resultado una reducción de la transparencia y constituye una oportunidad perdida para inspeccionar y adaptarse. (Schwaber & Sutherland, 2013)

El Sprint

El corazón de Scrum es el *Sprint*, es un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable. Es más conveniente si la duración de los *Sprints* es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo *Sprint* comienza inmediatamente después de la finalización del *Sprint* previo.

Los *Sprints* contienen y provienen de la Reunión de Planificación del *Sprint* (*Sprint Planning Meeting*), los Scrums Diarios (*Daily Scrums*), el trabajo de desarrollo, la Revisión del *Sprint* (*Sprint Review*), y la Retrospectiva del *Sprint* (*Sprint Retrospective*).

Durante el *Sprint*:

- No se realizan cambios que puedan afectar al Objetivo del *Sprint* (*Sprint Goal*);
- Los objetivos de calidad no disminuyen; y,
- El alcance puede ser clarificado y renegociado entre el Dueño de Producto y el Equipo de Desarrollo. (Schwaber & Sutherland, 2013)

Objetivo del *Sprint* (*Sprint Goal*)

El Objetivo del *Sprint* es una meta establecida para el *Sprint* que puede ser alcanzada mediante la implementación de la Lista de Producto. Proporciona una guía al Equipo de Desarrollo acerca de por qué se está construyendo el incremento. Es creado durante la reunión de planificación del *Sprint*. El Objetivo del *Sprint* ofrece al Equipo de Desarrollo cierta flexibilidad con respecto a la funcionalidad implementada en el *Sprint*. Los elementos de la Lista del Producto seleccionados ofrecen una función coherente, que puede ser el objetivo del *Sprint*. Este puede representar otro nexo de unión que haga que el Equipo de Desarrollo trabaje en conjunto y no en iniciativas separadas.

A medida que el Equipo de Desarrollo trabaja, se mantiene el objetivo del *Sprint* en mente. Con el fin de satisfacer esta meta, se implementa la funcionalidad y la tecnología. Si el trabajo resulta ser diferente a lo que el Equipo de Desarrollo espera, ellos colaboran con el Dueño del Producto para negociar el alcance de la Lista de pendientes del *Sprint* (*Sprint Backlog*). (Schwaber & Sutherland, 2013)

Artefactos de Scrum

Los artefactos de Scrum representan trabajo o valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación. Los artefactos definidos por Scrum están diseñados específicamente para maximizar la transparencia de la

información clave, que es necesaria para asegurar que todos tengan el mismo entendimiento del artefacto. (Schwaber & Sutherland, 2013)

Pila del producto

En inglés *product backlog*, Lista ágil de los requisitos del cliente o requisitos del sistema:

- Simples: expresados de forma breve, con una sentencia para cada uno, habitualmente con formato de historia de usuario o similar.
- Estimados: está medido el esfuerzo de construcción de cada requisito.
- Priorizados: se encuentran ordenados según la importancia para el cliente o responsable de la lista.

La Ingeniería del *Software* clásica diferencia dos ámbitos de requisitos:

- Requisitos del sistema
- Requisitos del *software*

Los requisitos del sistema forman parte del proceso de adquisición, y, por tanto, es responsabilidad del cliente: la definición del problema y de las funcionalidades que deben aportar la solución.

No importa si se trata de gestión tradicional o ágil. La pila del producto es responsabilidad del cliente, aunque se aborda de forma diferente en cada caso. Es el inventario de funcionalidades, mejoras, tecnología y corrección de errores que deben incorporarse al producto a través de los sucesivos *Sprints*.

Representa todo aquello que esperan el cliente, los usuarios y, en general, los interesados. Todo lo que suponga un trabajo que debe realizar el equipo debe estar reflejado en esta pila.

La pila de requisitos del producto nunca se da por completada; está en continuo crecimiento y evolución. Al comenzar el proyecto, incluye los requisitos conocidos y mejor entendidos, y conforme avanza el desarrollo y evoluciona el entorno en el que será usado. Su

continuo dinamismo refleja aquello que el producto necesita incorporar para ser el más adecuado a las circunstancias.

Para comenzar el desarrollo se necesita la visión del objetivo de negocio que se quiere conseguir con el proyecto, comprendida y conocida por todo el equipo; también deben estar al tanto de los elementos suficientes en la pila para llevar a cabo el primer *Sprint*. Habitualmente, se comienza a elaborar la pila con el resultado de una reunión de “tormenta de ideas”, o “fertilización cruzada”, o un proceso de “exploración” (*eXtreme Programming*), donde colaboran los miembros del equipo, partiendo de la visión del propietario del producto.

El formato de la visión no es relevante. Según los casos, puede ser una presentación informal del responsable del producto, un informe de requisitos del Departamento de marketing, u otros. Sin embargo, sí es importante disponer de una visión real, comprendida y compartida por todo el equipo. El propietario del producto mantiene la pila ordenada por la prioridad de los elementos, siendo los más relevantes aquellos que confieren mayor valor al producto, o por alguna razón resulten más necesarios, y determinen las actividades de desarrollo inmediatas.

El detalle de los requisitos en la pila del producto debe ser proporcional a la prioridad: Los elementos de mayor importancia deben tener mayor nivel de comprensión y detalle que el resto. De esta forma, el Equipo de Desarrollo puede descomponer un elemento de prioridad alta en tareas con la precisión suficiente para ser hecho en un *Sprint*. Los elementos de la pila del producto, que pueden ser incorporados a un *Sprint*, se denominan “preparados” o “accionables” y son los que pueden seleccionarse en la reunión de planificación del *Sprint*. (ScrumManager, 2014)

Pila del *Sprint*

Lista de tareas que va a realizar el Equipo en una iteración, para construir un Incremento. Para cada una se registra la siguiente información:

- Descripción breve.
- Persona que la tiene asignada.
- Esfuerzo pendiente para terminarla.

Información

Es útil porque descompone el proyecto en unidades de tamaño adecuado para determinar el avance a diario e identificar riesgos y problemas sin necesidad de procesos complejos de gestión. Es también una herramienta de soporte para la comunicación directa del Equipo.

Condiciones

- Realizada de forma conjunta por todos los miembros del Equipo.
- Cubre todas las tareas identificadas para conseguir el Objetivo del *Sprint*.
- Solo el equipo la puede modificar durante el *Sprint*.
- Las tareas demasiado grandes deben descomponerse en otras más pequeñas. Se deben considerar “grandes” las tareas que necesitan más de un día para realizarse.
- Es visible para todo el equipo. Idealmente en un tablero o pared en el mismo espacio físico donde se lleva a cabo el trabajo.

Formato y soporte

Son soportes habituales:

- Tablero físico o pared.
- Hoja de cálculo.
- Herramienta colaborativa o de gestión de proyectos.

Sobre el más adecuado a las características del proyecto, oficina y equipo, lo apropiado es diseñar el formato más cómodo para todos, teniendo en cuenta los siguientes criterios:

- Incluir la siguiente información: pila del *Sprint*, persona responsable de cada tarea, estado en el que se encuentra y tiempo de trabajo que queda para completarla.
- Incluir solo la información estrictamente necesaria.
- Debe servir de medio para registrar en cada reunión diaria del *Sprint*, el tiempo que le queda a cada tarea.
- Facilitar la consulta y la comunicación diaria y directa del equipo.

Durante el *Sprint*, el equipo actualiza a diario en ella los tiempos pendientes de cada tarea. Asimismo, con estos datos traza el gráfico de avance o trabajo consumido (*burn-down*). (ScrumManager, 2014)

3.6 ¿Por qué se seleccionó el marco de trabajo Scrum?

La utilización de procesos ágiles podría ser ventajoso para el desarrollo de aplicaciones web. La necesidad del cliente que contrata un desarrollo web es que su producto esté disponible en la red lo más pronto posible. Si no se tiene en cuenta esta necesidad, la aplicación no resultará un producto beneficioso para el cliente. Puesto que los procesos ágiles permiten tener versiones de producto previas a la versión final, si se aplican correctamente estos procesos, el cliente podrá disponer de forma rápida de alguna versión intermedia.

Además, el ciclo de desarrollo de la mayoría de los sitios y aplicaciones web es extremadamente corto. Los desarrollos web se perciben como sencillos por lo que los desarrolladores son sometidos a una gran presión de trabajo para terminar lo más pronto posible. Esta forma de trabajar implica, sin duda alguna, modificaciones. Por otra parte, sería conveniente garantizar un proceso de desarrollo adaptable a los cambios. Otro aspecto a tomar en cuenta es que las aplicaciones web se desarrollan sin conocer los perfiles de los usuarios finales de las mismas, es decir, sin conocer los requisitos de usuario del sistema. Esto provocará cambios en los requisitos inicialmente detectados, lo que corrobora nuevamente la necesidad de elegir un proceso adaptativo.

Scrum es un proceso en el que se aplica de manera regular un conjunto de prácticas para trabajar en equipo y obtener el resultado eficiente de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio sobre la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales del resultado final del proyecto, dado el gran beneficio que le brindan al receptor del proyecto. Esta metodología está especialmente indicada para proyectos en entornos complejos, que requieren ser hechos en menor tiempo, donde los requisitos del cliente son cambiantes y donde la innovación, la competitividad y la productividad son fundamentales.

Aunque Scrum surgió como modelo para el desarrollo de productos tecnológicos, también se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; situaciones frecuentes en el desarrollo de determinados sistemas de *software*, y donde al aplicarlo proporciona ventajas como:

- Entrega de un producto funcional al finalizar cada iteración.
- Posibilidad de ajustar la funcionalidad con base en la necesidad de negocio del cliente.
- Visualización del proyecto día a día.
- Alcance acotado y viable.
- Equipos integrados y comprometidos con el proyecto, dado que ellos definieron el proyecto.

Entre los principales beneficios que reporta utilizar Scrum como metodología de desarrollo de un sistema se encuentran además:

- Entrega mensual (o quincenal) de resultados (los requisitos más prioritarios en ese momento, ya completados).
- Productividad y calidad. De manera regular, el equipo va mejorando y simplificando su forma de trabajar.
- Alineamiento entre el cliente y el Equipo de Desarrollo. Los resultados y esfuerzos del proyecto se miden en forma de objetivos y requisitos entregados al negocio. Todos los participantes en el proyecto conocen cuál es el objetivo a conseguir. El producto se enriquece con las aportaciones de todos.
- Equipo motivado. Las personas están más motivadas cuando pueden usar su creatividad para resolver problemas y cuando pueden decidir organizar su trabajo.

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable; cuando se necesita capacidad de reacción ante la competencia; cuando la moral de los equipos es baja y la rotación alta; cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto. (ScrumManager, 2014)

CAPÍTULO IV. DESARROLLO DE LA SOLUCIÓN

4.1 Objetivo General

Implementar un módulo para la gestión de la Programación Docente, específicamente la gestión de preparadores de pregrado, mediante la automatización de sus procesos.

4.2 Objetivos Específicos

1. Permitir a los Coordinadores de Centro y a los Coordinadores de Materia solicitar la apertura de concursos para sus respectivos centros o materias, y el ingreso de los resultados de concurso.
2. Proveer al Jefe de Departamento una interfaz para gestionar las solicitudes de apertura de concursos generadas por los Coordinadores.
3. Asignar un ID detalle a cada miembro del personal preparador y auxiliar docente para facilitar su ingreso en nómina.
4. Proporcionar la funcionalidad de ratificar o retirar preparadores y auxiliares docentes que estén asignados a una materia o centro.
5. Suministrar una interfaz a los preparadores y auxiliares docentes para la subida de sus recaudos al momento de registrarse como preparadores de una materia.
6. Permitir a los usuarios autorizados consultar la información referente a los preparadores y auxiliares docentes y descargar los recaudos de los mismos para su entrega al Consejo de Escuela.
7. Proveer al módulo de la generación de la programación docente los datos de los preparadores y auxiliares docentes asignados a las distintas materias y/o centros.
8. Asignar roles a los distintos usuarios.
9. Enviar notificaciones masivas por correo.

4.3 Alcance de la aplicación

Debido a la complejidad del manejo del ID detalle para la asignación de un preparador a nómina, ha sido excluida de esta solución la posibilidad de registrar a un preparador ‘Ad Honorem’ en el sistema. El manejo de preparadores ‘Ad Honorem’ quedaría para futuras etapas de desarrollo sobre esta aplicación.

4.4 Plataforma utilizada en su desarrollo

En lo referente a la plataforma, se utilizaron dos computadoras corriendo sobre el sistema operativo Ubuntu Gnome. Para facilitar el manejo de los cambios en el código se usó el manejador de versiones Git. En cuanto al desarrollo, para el lado del servidor se utilizó el *framework* Laravel de PHP, y para el lado del cliente AngularJS y PostgreSQL como sistema de base de datos.

Para el uso de la aplicación, es necesario utilizar un dispositivo móvil o de escritorio que posea un navegador que soporte los estándares de HTML5 y CSS3 y que tenga habilitado el uso de JavaScript.

4.5 Método de desarrollo utilizado y etapas seguidas para la creación de la aplicación

El método de desarrollo utilizado fue SCRUM modificado y estructurado para un grupo de trabajo de 2 personas, con algunos artefactos extras como los diagramas de casos de uso y de secuencia. En cuanto a las etapas seguidas para la creación de la aplicación, se aplicó un proceso incremental, dividiendo el desarrollo en etapas de análisis, diseño, código y pruebas.

- **Iteración 1** - Levantamiento de Requerimientos: realización de entrevistas y reuniones con los profesores Carlos Acosta y Marcel Castro para analizar los procesos de negocios asociados al Departamento y definir las funcionalidades del sistema.
- **Iteración 2** - Diseño de la interfaz: establecimiento de los lineamientos básicos de diseño a seguir en la interfaz, así como la estructura base para todo el sistema.
- **Iteración 3** - Manejo de Sesiones: definición de la manera en la que se manejarían las sesiones, tanto del lado del cliente como del lado del servidor.
- **Iteración 4** - Modificación de Datos Personales: creación de una interfaz que permita a los usuarios actualizar sus datos personales.
- **Iteración 5** - Gestión de Centros y Materias: desarrollo de una interfaz para operaciones de CRUD (acrónimo de Crear, Obtener, Actualizar y Borrar proveniente del original en inglés: *Create, Read, Update and Delete*) sobre centros y materias.

- **Iteración 6** - Solicitudes de Concursos de Preparadores: creación del flujo a seguir para la solicitud de apertura de nuevos concursos de preparadores y se desarrollaron las interfaces con sus controladores.
- **Iteración 7** - Sistema de Notificaciones: diseño e implementación de un sistema de notificaciones globales para todos los módulos del sistema.
- **Iteración 8** - Gestión de Personal Preparador: establecimiento de las funcionalidades asociadas a la gestión de preparadores, incluyendo la subida de los recaudos necesarios.
- **Iteración 9** - Gestión de Roles de Usuario: definición de un módulo para modificar los distintos roles de los usuarios, en caso de ser necesario.
- **Iteración 10** - Pruebas y Entrega: prueba de todas las funcionalidades del sistema y corrección de los errores encontrados.

4.6 Descripción de la aplicación

Usuarios

Entre los usuarios de la aplicación se encuentran el personal docente (profesores, preparadores y auxiliares docentes), administrativo, autoridades de la Escuela de Computación y otras instancias involucradas en el procesamiento de solicitudes del sistema.

En la siguiente tabla se muestra la relación entre los perfiles definidos para los usuarios y funcionalidades provistas en la aplicación.

Funciones Perfil	Creación	Consulta	Modificación	Creación	Consulta	Modificación	Notificaciones	Solicitudes de concurso	Llenar formulario de propuestas	Aceptar o rechazar propuestas	Creación	Consulta	Modificación	Descarga	Acceder a la bitácora del sistema	Documentos
	Usuarios y perfiles			Materias y Centros							Planificación Docente Tentativa					
Administrador del sistema	Si			Si			Enviar	n/a	n/a	n/a	n/a				n/a	n/a
Jefe de Departamento	n/a			Si			Enviar	Gestionar	n/a	Si	Si				Si	Exportar
Director de Escuela	n/a			Si			Enviar	Gestionar	n/a	Si	Si				Si	Exportar
Secretaria del Departamento	n/a			Si (Solo consulta)			n/a	Consultar	n/a	n/a	Si (Solo consulta)				n/a	Exportar
Secretaria del Dirección	n/a			Si (Solo consulta)			n/a	Consultar	n/a	n/a	Si (Solo consulta)				n/a	Exportar
Coordinador de Centro	n/a			Si (Solo consulta)			Enviar	Gestionar	n/a	Si	n/a				n/a	n/a
Coordinador de Materia	n/a			Si (Solo consulta)			Enviar	Gestionar	n/a	n/a	n/a				n/a	n/a
Profesor	n/a			Si (Solo consulta)			n/a	n/a	Si	n/a	n/a				n/a	n/a

Auxiliar Docente	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Importar y exportar
Preparador	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Importar y exportar

Tabla 1. Tabla de perfiles y funcionalidades

A continuación se aprecia con mayor detalle el proceso de las diferentes funcionalidades que formarán parte de la solución de la gestión de usuarios y perfiles, materias y centros, notificaciones, solicitudes de concurso y documentos; utilizando diagramas de secuencia y/o diagramas de actividades.

Gestión de usuarios y perfiles: la solución para este proceso involucra al administrador del sistema, que está en la posibilidad de modificar cualquier parámetro asociado a cada usuario, sea profesor o estudiante.

Los pasos a aplicar serían tanto para creación, consulta y modificación

- El administrador del sistema ingresa al sistema.
- Selecciona la opción “Gestionar usuarios y perfiles”.
- En caso de actualización, el usuario ingresa la nueva información y esta es incorporada al sistema.

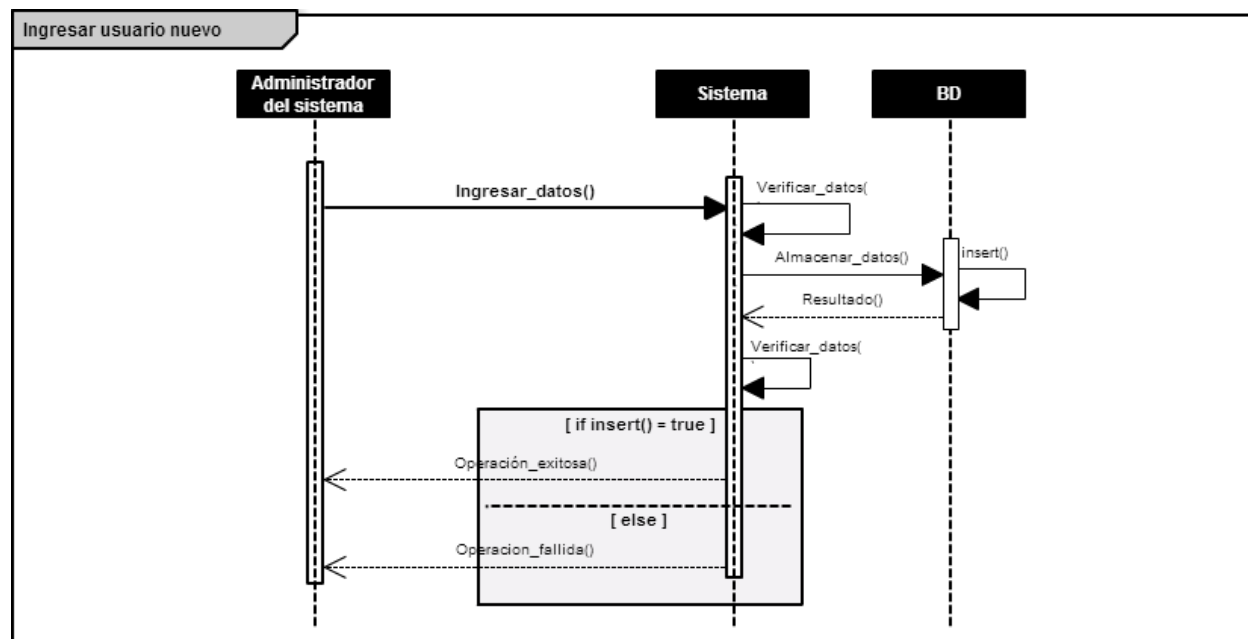


Figura 8. Diagrama de secuencia - Ingresar usuario nuevo

Solicitud de concurso: La solución para este proceso involucra a los coordinadores de centro o materia y el Jefe de Departamento y tiene como fin optimizar y organizar los concursos.

Los concursos tendrán cuatro estados: pendiente, aceptado, rechazado y finalizado. Los pasos a aplicar, para la creación de concursos, serían:

- El Coordinador ingresa al sistema.
- Selecciona la opción “Consultar concursos”.
- Aceptar o rechazar concurso por parte del Jefe de Departamento
- Editar o descartar concurso por parte del coordinador.

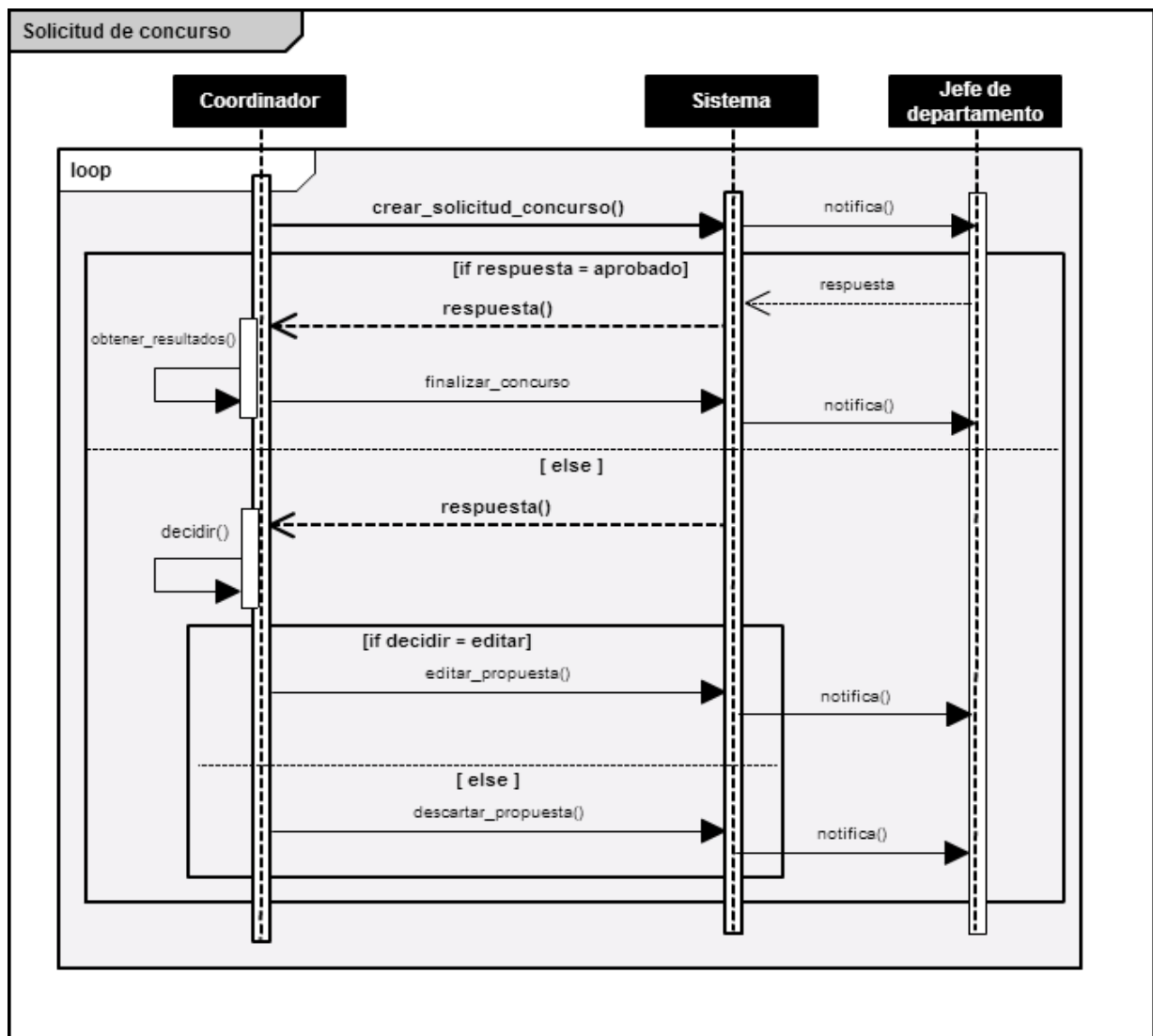


Figura 9. Diagrama de secuencia - Solicitud de concurso

Ratificar y retirar preparadores o auxiliares docentes: La solución para este proceso involucra a los coordinadores de centro o materia y el Jefe de Departamento. Tiene como objetivo mantener un orden en las plazas disponibles y en la cantidad de preparadores o

auxiliares docentes que se encuentren en una materia o centro. Los pasos a aplicar para ratificar a un preparador o auxiliar docente serían:

- El Coordinador o Jefe de Departamento ingresa al sistema.
- Selecciona la opción “Gestionar preparadores”.
- Ratificar o retirar preparadores o auxiliares docentes.

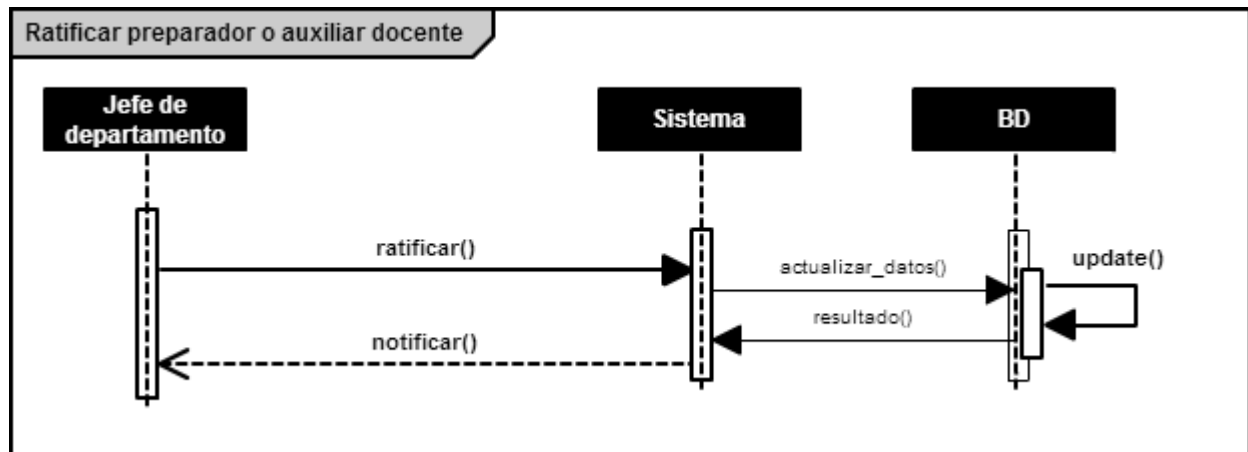


Figura 10. Diagrama de secuencia - Ratificar preparador o auxiliar docente

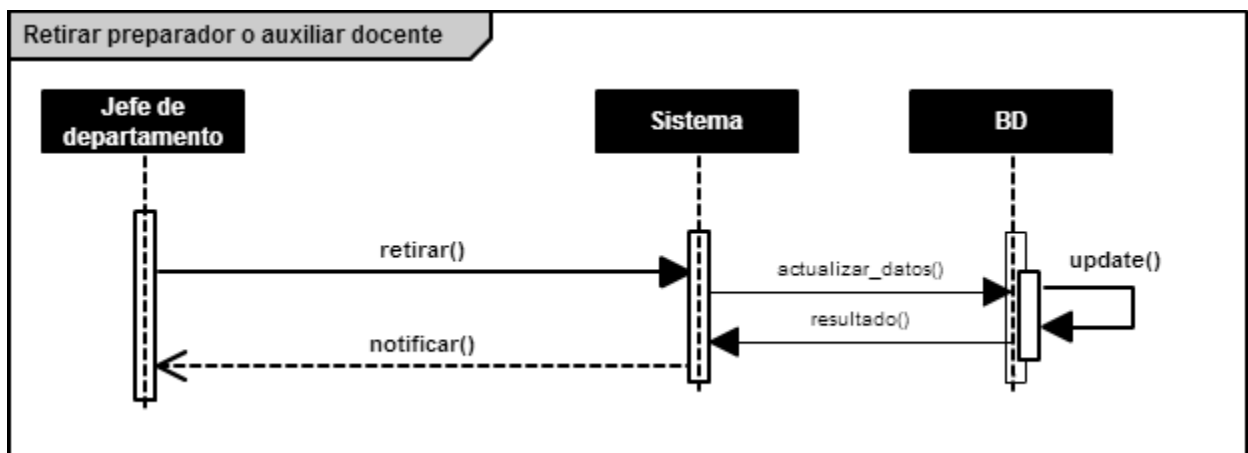


Figura 11. Diagrama de secuencia - Retirar preparador o auxiliar docente

Principales funcionalidades de la aplicación

- **Gestión de concursos de preparadores y auxiliares docentes**
 - **Solicitud de concursos:** permite a los coordinadores, ya sean de centro o materia, solicitar la apertura de un concurso enviando una petición al Jefe

de Departamento, quien tendrá que decidir si la Escuela está o no en capacidad de plazas para preparadores o auxiliares docentes.

- **Edición de concursos:** los coordinadores, en caso de que su solicitud de concurso sea rechazada, tienen la posibilidad de editar dicha solicitud y reenviarla.
- **Rechazo de concursos:** el Jefe del Departamento tiene la permisología y autoridad de rechazar cualquier solicitud de concurso, haciendo observaciones e indicando al coordinador que la solicitó el porqué del rechazo.
- **Aceptación de concursos:** el Jefe del Departamento aceptará un concurso si considera que es necesario y si la Escuela tiene plazas disponibles.
- **Finalización de concursos:** después de que el Jefe de Departamento acepte un concurso y el coordinador realice dicho concurso y obtenga a él o los ganadores, este debe proceder a finalizar el concurso ingresando los datos de las personas que obtuvieron el cargo de preparador o auxiliar docente.
- **Registro de preparador y auxiliar docente:** después de que una persona resulte ganadora de un concurso, ya sea preparador o auxiliar docente, tiene que formalizar su registro en SIGEPROD.
- **Gestión de preparadores y auxiliares docentes**
 - **Ratificación:** la duración de un contrato de preparador o auxiliar docente es de un año. Si esta persona quiere seguir prestando sus servicios a la Escuela y el Coordinador lo aprueba, este debe ratificarlo anualmente.
 - **Retiro:** si el preparador o auxiliar docente no quiere seguir prestando servicio o el Coordinador decide que no puede seguir ocupando el cargo, este debe retirarlo.
- **Envío de notificaciones masivas:** funcionalidad que le permite a ciertos roles del sistema como, por ejemplo, al Jefe del Departamento enviar notificaciones, tanto en el sistema web como al correo registrado de los destinatarios.
- **Gestión de materias y centros**
 - **Crear:** permite crear centros y materias en el sistema.

- **Editar:** permite editar centros y materias en el sistema.
- **Edición de datos personales:** cada usuario de SIGEPROD puede editar sus datos personales, nombres, correo, teléfono, entre otros.
- **Consulta de documentos:** ciertos roles en el sistema tendrán la posibilidad de consultar y descargar los documentos asociados a los preparadores.
- **Consulta de bitácora de actividades:** SIGEPROD cuenta con una bitácora en la que se reflejan las acciones de todos los usuarios del sistema.
- **Actualización de documentos:** los preparadores o auxiliares docentes pueden actualizar los documentos que previamente subieron al momento de registrarse en el sistema.

Casos de uso más importantes

Las Figuras 12, 13 y 14 muestran el nivel 0, 1 y 2 del modelado a través de diagramas de casos de uso.

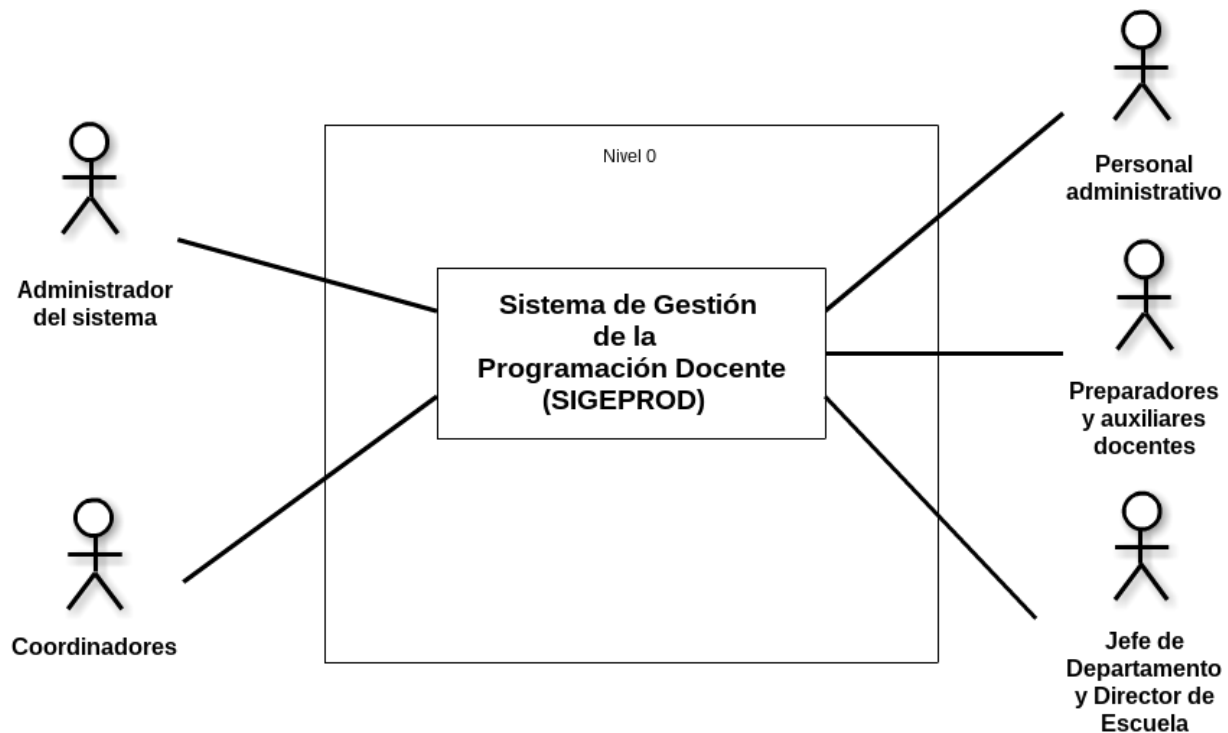


Figura 12. Diagrama de casos de uso - Nivel 0

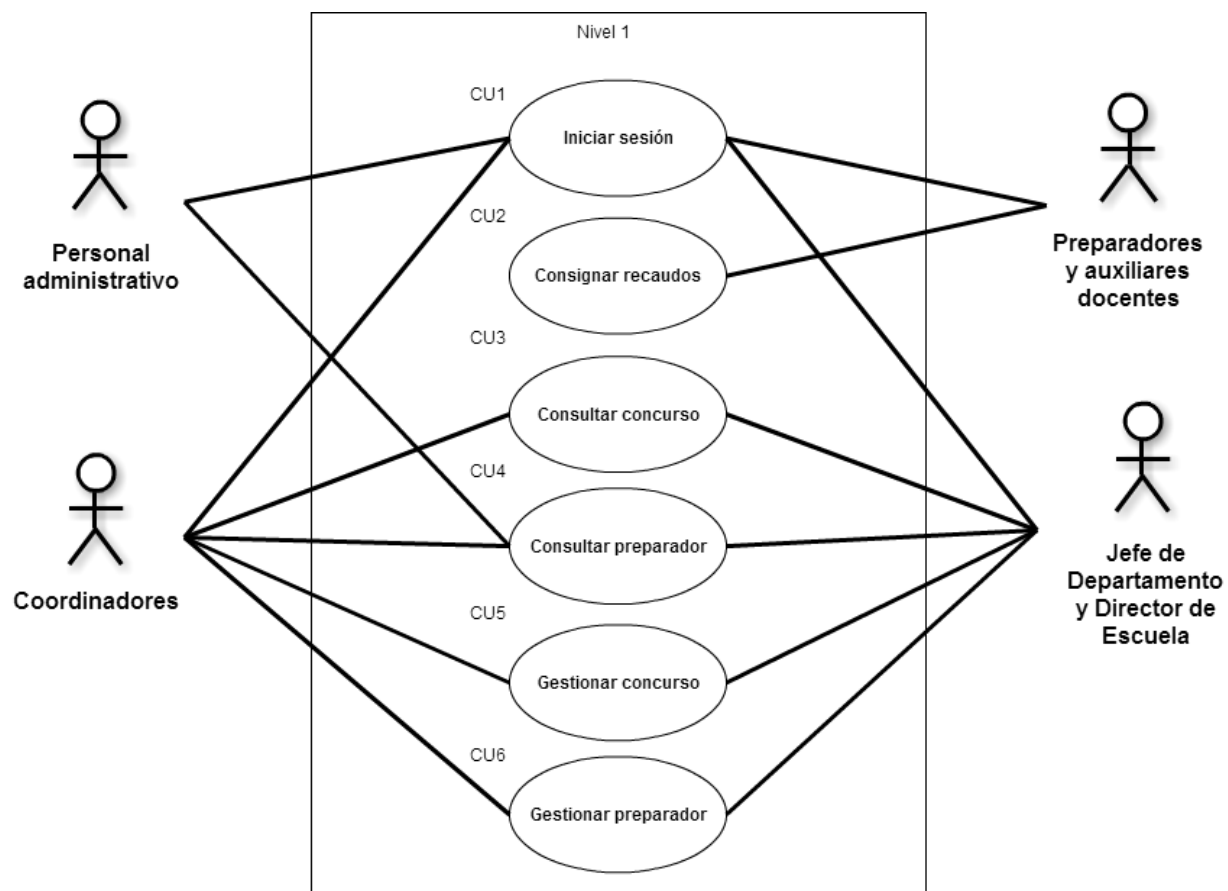


Figura 13. Diagrama de casos de uso - Nivel 1

Nivel 2

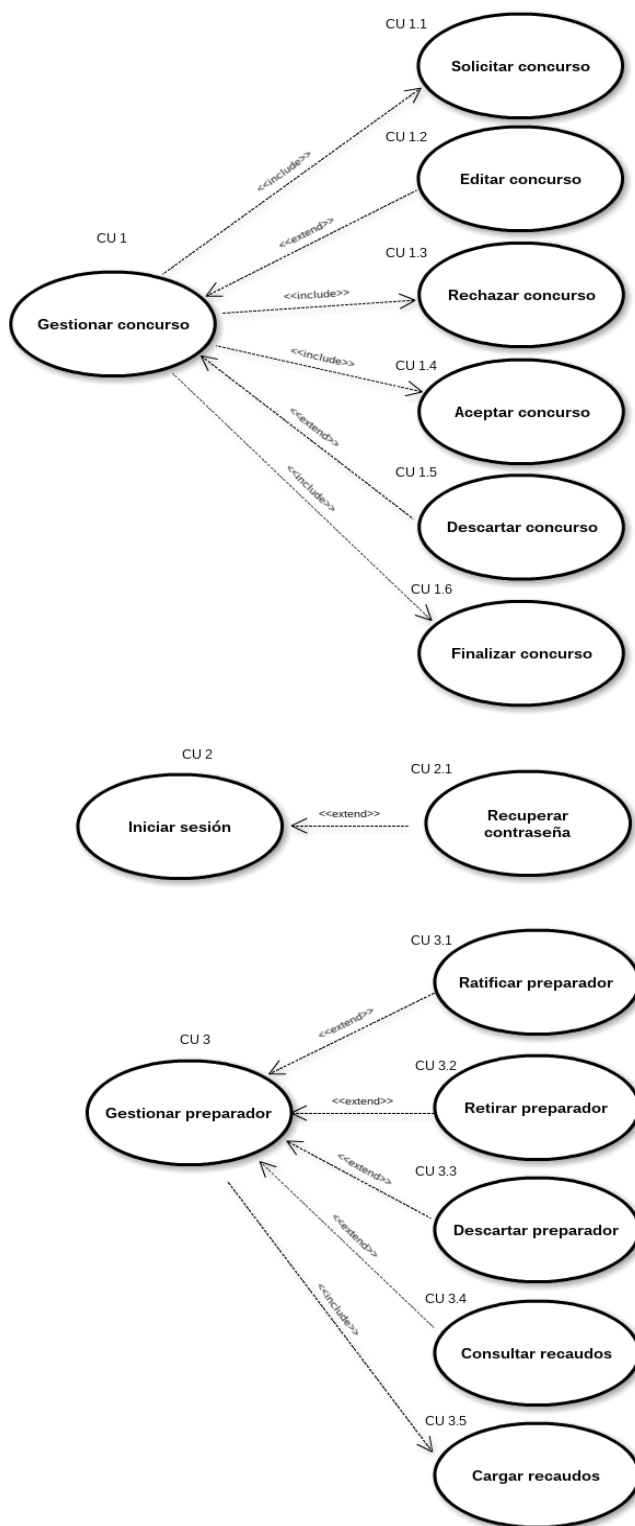


Figura 14. Diagrama de casos de uso - Nivel 2

Especificación de los casos de uso más importantes:

Caso de uso: Solicitar concurso

Actores: Coordinadores

Flujo normal	Flujo alternativo
El coordinador selecciona la opción “Gestión de Concursos” en el menú lateral izquierdo.	
El coordinador escoge un centro o materia (según sea el caso).	No se permite enviar una solicitud de concurso sin especificar para cuál centro o materia es dicho concurso.
3) Caso A: Si el coordinador selecciona una materia, deberá especificar la cantidad de Preparadores I y II. Caso B: Si el Coordinador selecciona un centro, deberá especificar la cantidad de Preparadores I, II y Auxiliares docentes.	Es necesario especificar la cantidad de personal preparador a solicitar de cada tipo para poder completar la solicitud.
4) La solicitud de concurso es enviada al Jefe de Departamento para su posterior aprobación o rechazo.	

Tabla 2. Especificación de casos de uso - Solicitar concurso

Caso de uso: Aceptar concurso

Actores: Jefe de Departamento

Flujo normal	Flujo alternativo
El Jefe de Departamento selecciona la opción “Gestión de Concursos” en el menú lateral izquierdo.	
2) El Jefe de Departamento expande una de las solicitudes con estado “Pendiente” y hace clic sobre el botón “Aceptar”.	
3) El estado del concurso es modificado a “Aceptado” y se reservan las plazas solicitadas por el Coordinador para su concurso.	No hay disponibles suficientes plazas para satisfacer la solicitud, por lo que el Jefe de Departamento rechaza el concurso, con la opción de agregar una observación, para que el Coordinador haga una nueva solicitud adaptándose a la disponibilidad de plazas.

Tabla 3. Especificación de casos de uso - Aceptar concurso

Caso de uso: Rechazar concurso

Actores: Jefe de Departamento

Flujo normal	Flujo alternativo
El Jefe de Departamento selecciona la opción “Gestión de Concursos” en el menú lateral izquierdo.	
2) El Jefe de Departamento expande una de las solicitudes con estado “Pendiente” y hace clic sobre el botón “Rechazar”.	
3) Se despliega una ventana modal de confirmación, permitiéndole al Jefe de Departamento agregar una observación acerca de la razón del rechazo.	Es posible cerrar la ventana modal de confirmación, cancelando así el rechazo del concurso.
4) Se actualiza el estado del concurso, cambiándolo a “Rechazado”. En caso de que se haya agregado una observación, esta es agregada a la información del concurso, y el Coordinador solicitante recibe una notificación.	

Tabla 4. Especificación de casos de uso - Rechazar concurso

Caso de uso: Editar concurso

Actores: Coordinadores

Flujo normal	Flujo alternativo
El Coordinador selecciona la opción “Gestión de concursos” en el menú lateral izquierdo.	
2) El Coordinador expande una de las solicitudes con estado “Rechazado” y hace clic sobre el botón “Editar”.	
3) Se despliega una ventana modal de edición del concurso, que permite modificar la cantidad de plazas solicitadas, y donde se muestran las observaciones emitidas sobre el concurso hasta este punto.	Es posible cerrar la ventana modal de edición.
4) Se actualiza el estado del concurso a “Pendiente”. En caso de que se haya agregado una observación, esta es agregada a la información del concurso y el Jefe de Departamento recibe una notificación.	

Tabla 5. Especificación de casos de uso - Editar concurso

Caso de uso: Descartar concurso

Actores: Coordinadores

Flujo normal	Flujo alternativo
El Coordinador selecciona la opción “Gestión de Concursos” en el menú lateral izquierdo.	
2) El Coordinador expande una de las solicitudes con estado “Rechazado” y hace clic sobre el botón “Descartar”.	
3) Se despliega una ventana modal de confirmación de rechazo, mostrando los detalles de dicho concurso.	Es posible cerrar la ventana modal de rechazo.
4) El concurso es marcado como “Descartado”.	

Tabla 6. Especificación de casos de uso - Descartar concurso

Caso de uso: Finalizar concurso

Actores: Coordinadores

Flujo normal	Flujo alternativo
El Coordinador selecciona la opción “Gestión de Concursos” en el menú lateral izquierdo.	
1) El Coordinador expande una de las solicitudes con estado “Aprobado” y hace clic sobre el botón “Finalizar”.	
3) Se despliega una ventana modal en la cual se solicitan los datos de todos los preparadores que resultaron ganadores del concurso.	
4) Se actualiza el estado del concurso a “Finalizado”.	

Tabla 7. Especificación de casos de uso - Finalizar concurso

Caso de uso: Retirar preparador**Actores:** Coordinadores, Jefe de Departamento

Flujo normal	Flujo alternativo
El usuario selecciona la opción “Gestión de Preparadores” en el menú lateral izquierdo.	
2) El usuario expande el acordeón de uno de los preparadores con estado “Ratificar” o “Activo” y presiona el botón “No renovar - Retirar”.	
3) Se retira el preparador como preparador de esa materia/centro.	
4) Caso A: En caso de que tenga que modificarse su tipo de ID detalle (e.g: Pasa de ser Preparador II a ser Preparador I), se busca un ID detalle de ese tipo, que esté desocupado, se le asigna el nuevo, y se libera el ID detalle anterior. Caso B: Si es preparador de otra materia/centro, con el mismo tipo de preparador, se queda con su ID detalle actual. Caso C: Si no es preparador de otra materia/centro, se libera su ID detalle para posteriores concursos.	

Tabla 8. Especificación de casos de uso - Retirar preparador

Caso de uso: Ratificar preparador**Actores:** Coordinadores, Jefe de Departamento

Flujo normal	Flujo alternativo
El usuario selecciona la opción “Gestión de Preparadores” en el menú lateral izquierdo.	
2) El usuario expande el acordeón de uno de los preparadores con estado “Ratificar” y presiona el botón “Ratificar”.	
3) Se actualiza el estado del preparador a “Activo”.	

Tabla 9. Especificación de casos de uso - Ratificar preparador

Caso de uso: Consultar recaudos**Actores:** Coordinadores, Jefe de Departamento, Secretarias, Preparadores

Flujo normal	Flujo alternativo
El usuario selecciona la opción “Gestión de Preparadores”, o “Actualización de Recaudos” para Preparadores, en el menú lateral izquierdo.	
2) El usuario expande el acordeón de uno de los preparadores y selecciona “Descargar archivos”, o en el caso de que el usuario sea Preparador, selecciona uno de los archivos a consultar.	

Tabla 10. Diagrama de casos de uso - Nivel 2

Caso de uso: Actualizar recaudos

Actores: Preparadores

Flujo normal	Flujo alternativo
El Preparador selecciona la opción “Actualización de Recaudos” en el menú lateral izquierdo.	
2) El usuario selecciona la pestaña “Actualizar archivos”.	
3) Dependiendo de qué archivo quiera actualizar, se presiona el botón respectivo al tipo de recaudo.	
4) Se despliega una ventana para seleccionar el archivo a subir. Una vez se seleccione, el archivo se subirá de manera automática.	

Tabla 11. Especificación de casos de uso - Consultar recaudos

Caso de uso: Recuperar contraseña

Actores: Todos los usuarios

Flujo normal	Flujo alternativo
En la pantalla de inicio de sesión, el usuario selecciona la opción “Recuperar contraseña”.	
2) Se introduce el correo electrónico asociado a la cuenta de la cual se quiere recuperar la contraseña.	
3) Es enviado un correo a la dirección de correo electrónico especificada, con una nueva clave generada por el sistema.	Si el correo electrónico no existe en el sistema, se muestra un mensaje de error notificando al usuario.

Tabla 12. Especificación de casos de uso - Recuperar contraseña

Principales interfaces de la aplicación

A continuación se muestran las principales interfaces de la aplicación, en lo referente al manejo del personal preparador:

En la Figura 15, se aprecia la interfaz inicial de la aplicación. El personal docente, administrativo, preparador y auxiliar docente, podrá iniciar sesión, siempre y cuando estén registrados y autorizados.

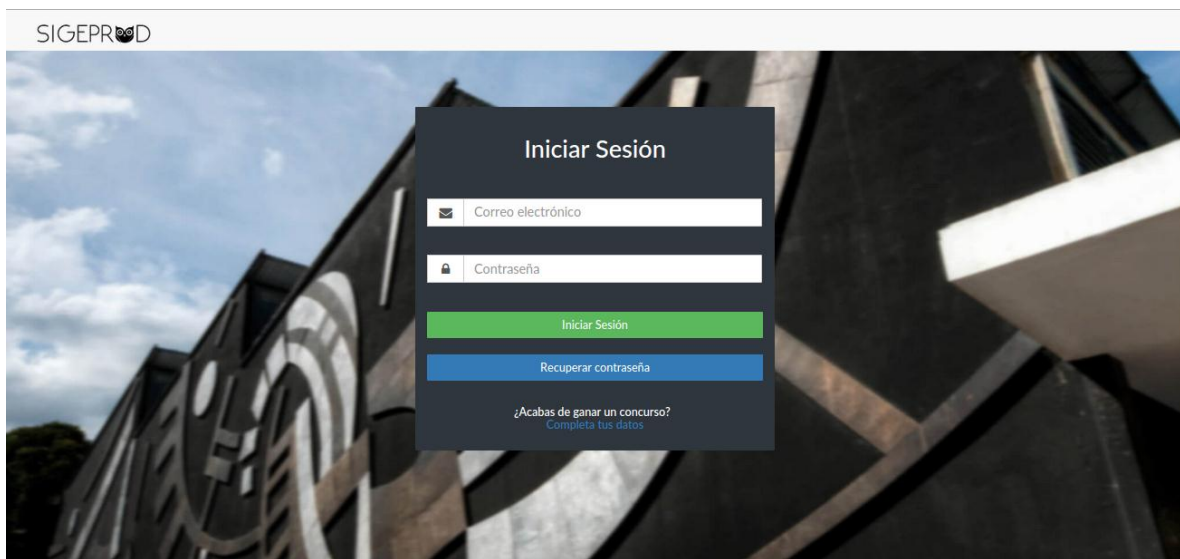


Figura 15. Principales interfaces - Inicio de Sesión

La Figura 16, es la interfaz que permite recuperar una contraseña olvidada.

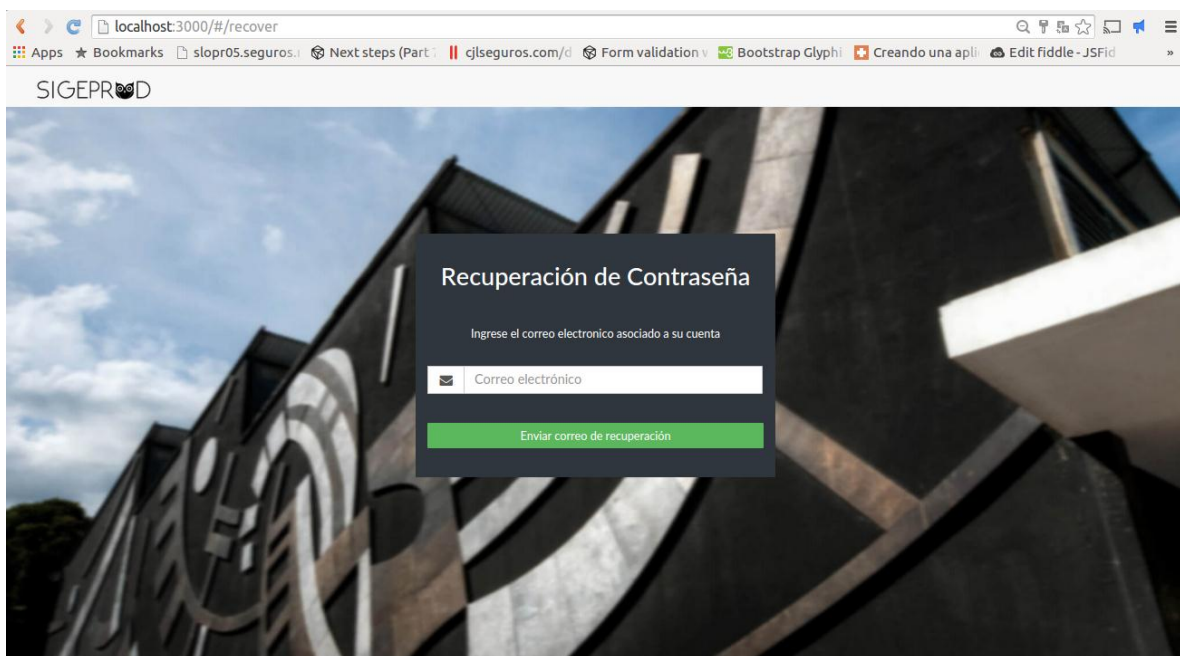


Figura 16. Principales interfaces - Recuperación de contraseña

La Figura 17, es la interfaz principal, después de iniciar sesión satisfactoriamente.



Figura 17. Principales interfaces - Pantalla inicial

En la Figura 18, se muestra la interfaz para que cada usuario pueda consultar y actualizar el perfil del usuario en cualquier momento.

Figura 18. Principales interfaces – Consulta y edición de datos personales

En la Figura 19, se aprecia la interfaz de concurso de preparadores, donde se muestran los datos de cada concurso.



Figura 19. Principales Interfaces - Concursos de Preparadores

En la Figura 20, para cada concurso se cuenta con un cuadro de conversación en donde se especifica la razón de dicho concurso y sus respuestas.

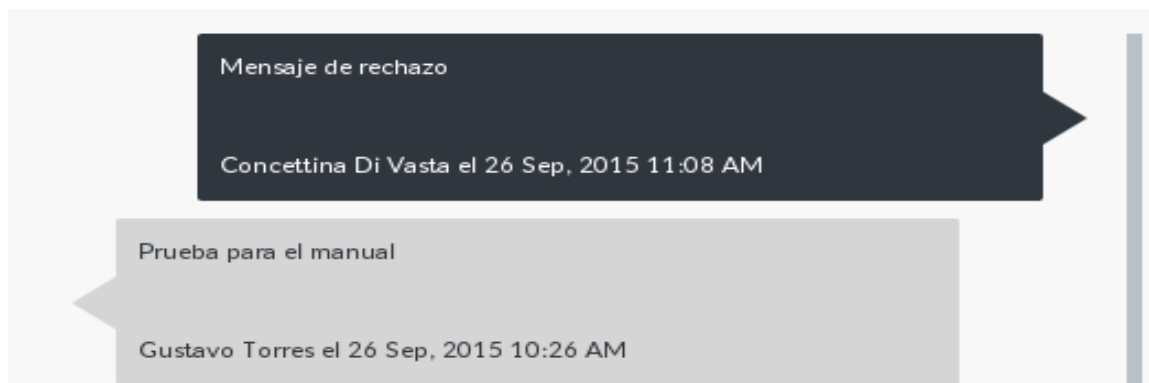


Figura 20. Principales Interfaces - Lista de observaciones

En la Figura 21, en la esquina superior derecha se cuenta con una lista de notificaciones leídas y no leídas, que haciendo clic en alguna de ellas el sistema redirecciona a la interfaz correspondiente de esa notificación.

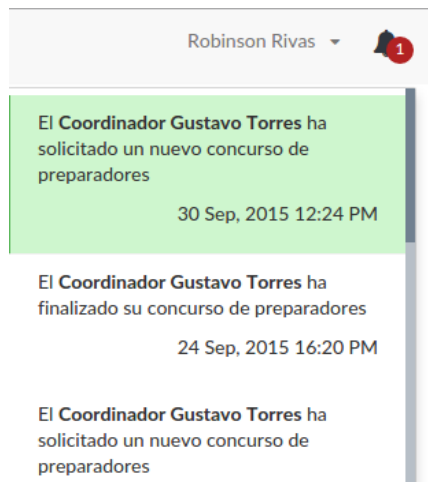


Figura 21. Principales Interfaces - Lista de notificaciones

Las Figuras 22 y 23, muestran el modal que se despliega, ya sea coordinador de centro o materia, para generar un nuevo concurso.

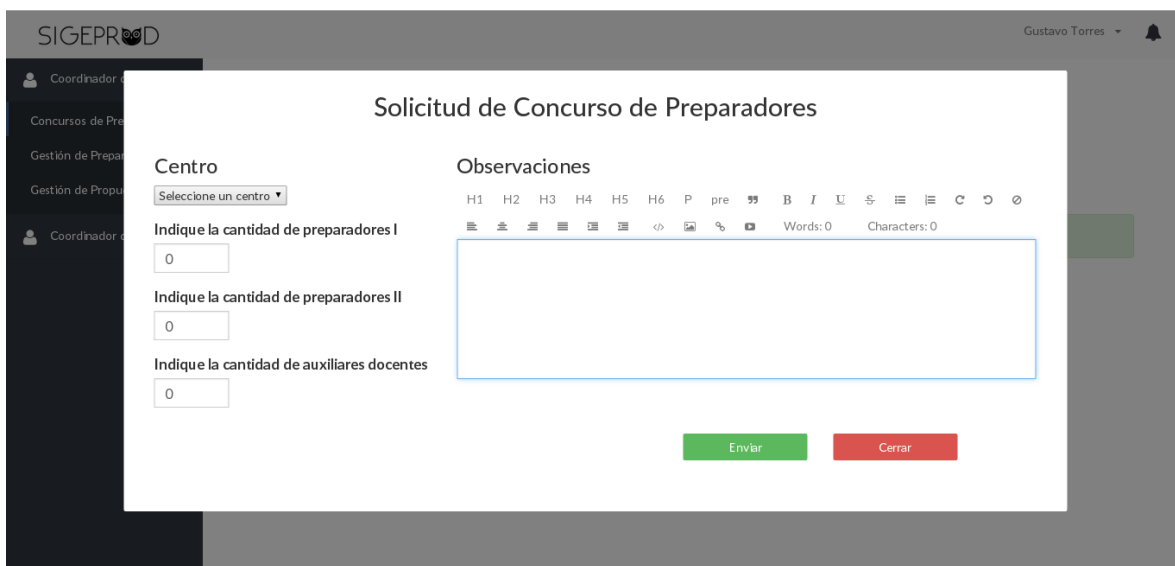


Figura 22. Principales Interfaces - Modal de Solicitud de Concurso de Preparadores para Centros

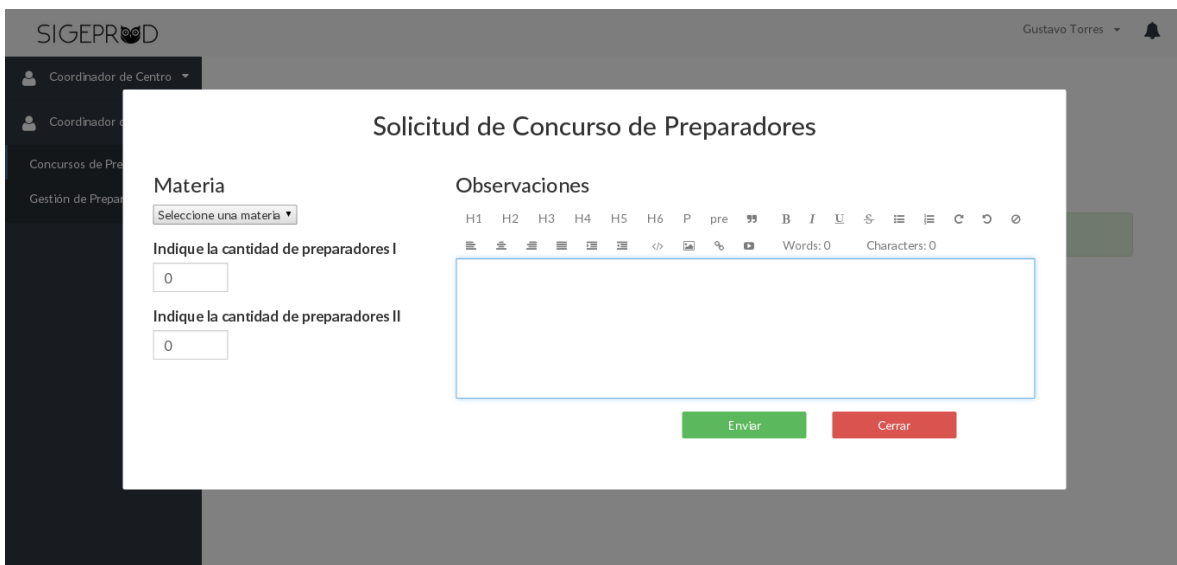


Figura 23. Principales Interfaces - Modal de Solicitud de Concurso de Preparadores para Materias

En la Figura 24, se muestra el modal de confirmación para descartar un concurso previamente rechazado por el Jefe de Departamento.

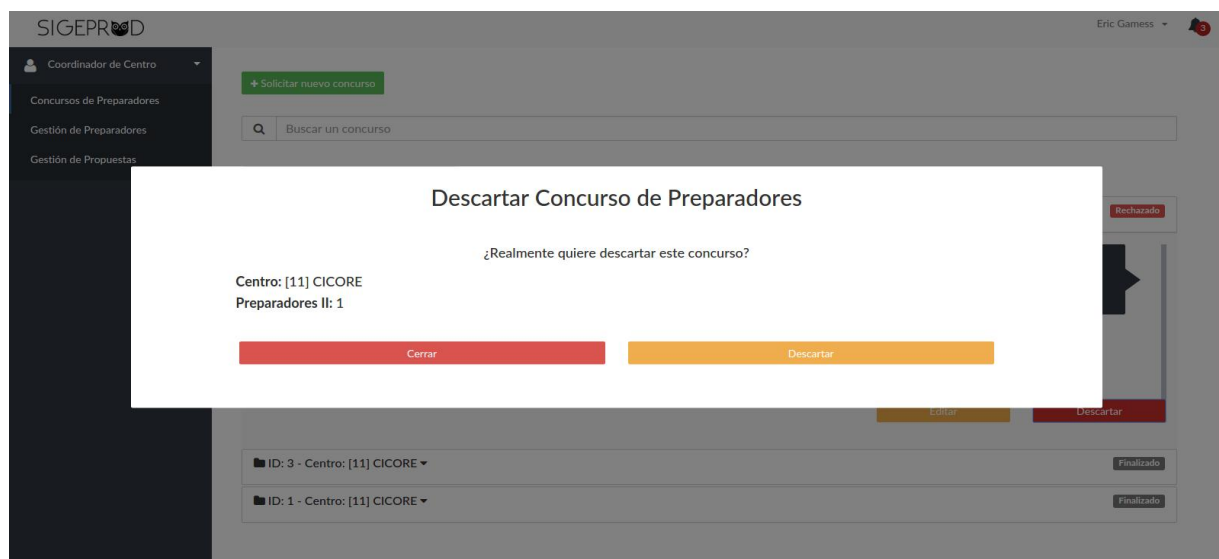


Figura 24. Principales Interfaces - Descartar concurso de preparadores

En la Figura 25, se observa el modal de editar un concurso previamente rechazado por el Jefe de Departamento.

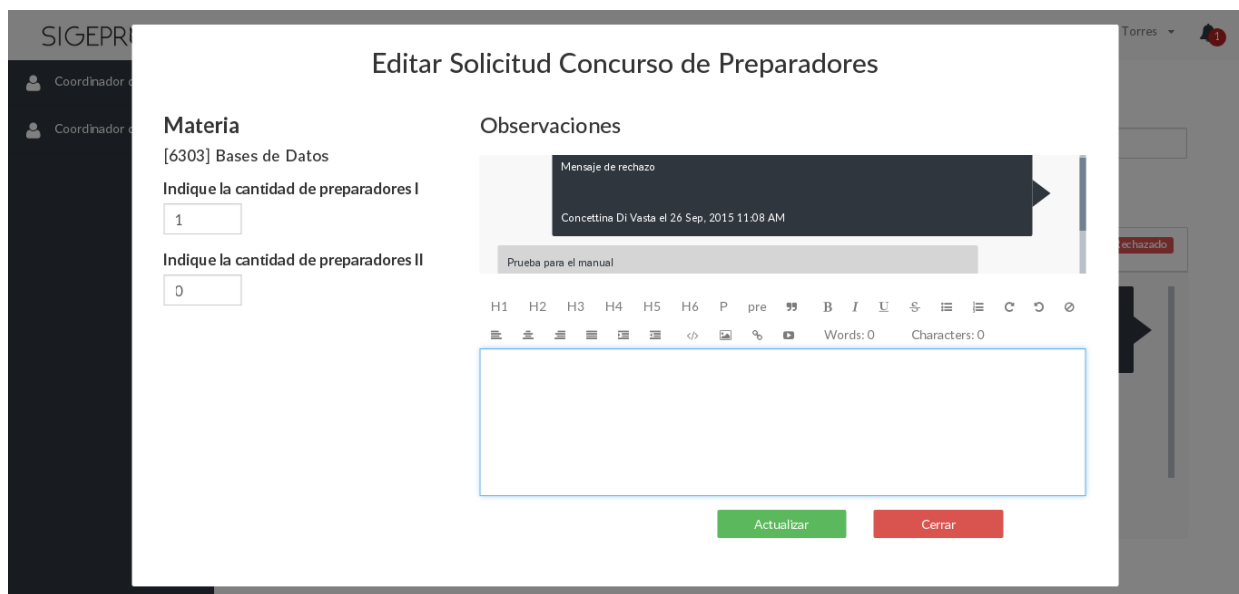


Figura 25. Principales Interfaces - Editar solicitud concurso de preparadores

La Figura 26, se muestra el modal para rechazar un concurso, e indicar la razón del rechazo.

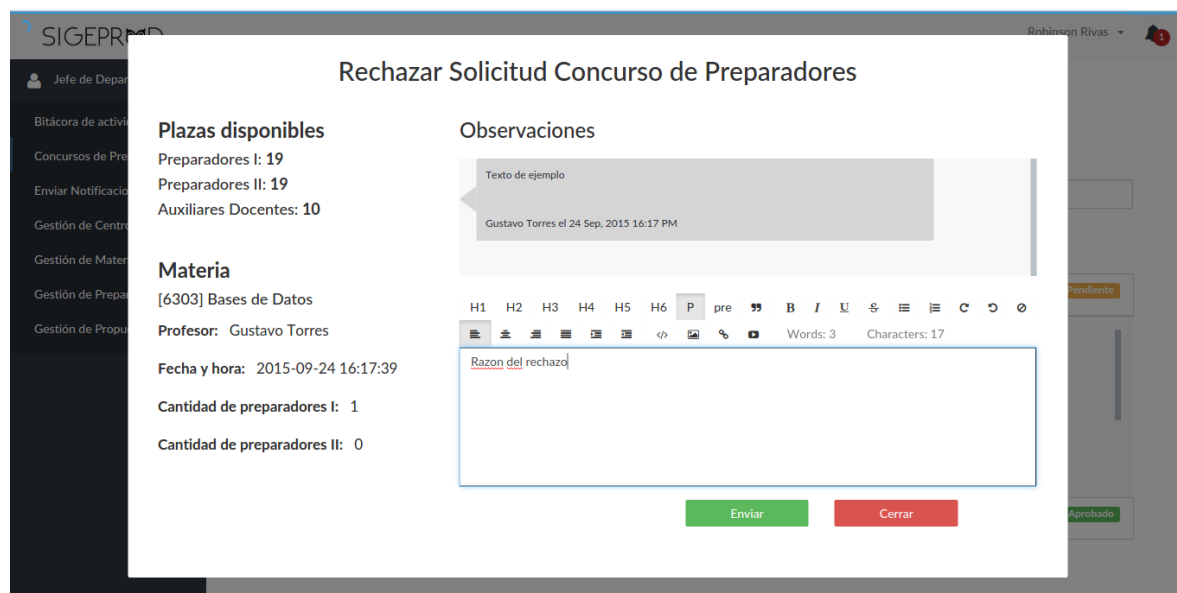


Figura 26. Principales Interfaces - Rechazar solicitud de concurso de preparadores

En la Figura 27, se muestra la interfaz en donde, después de que el concurso se haya aprobado y realizado, el Coordinador debe ingresar los datos de los ganadores del concurso.

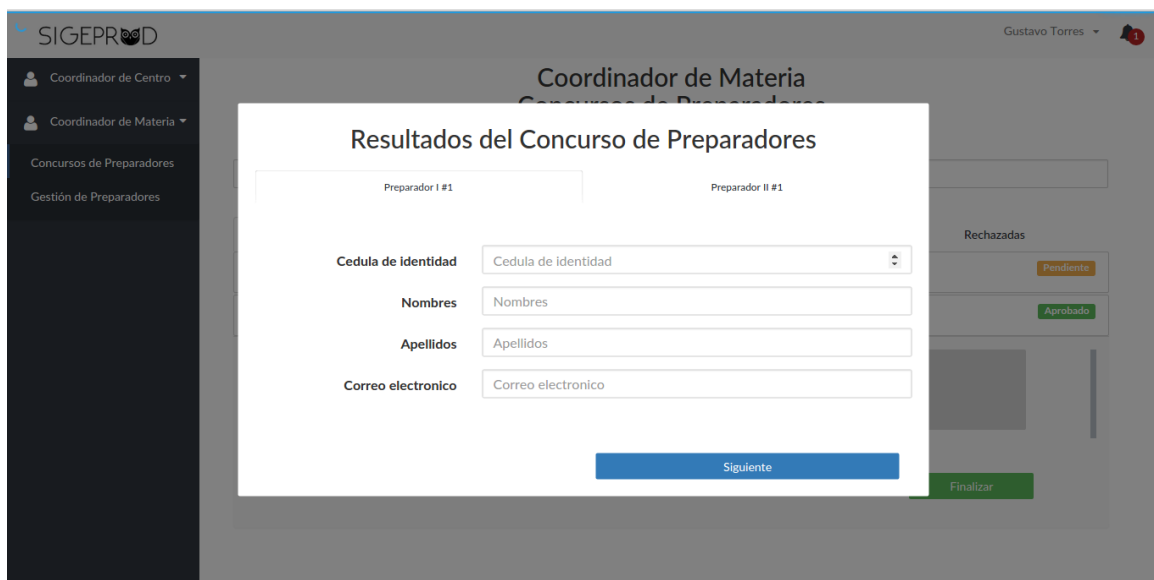


Figura 27. Principales Interfaces - Resultados del concurso de preparadores

La Figura 28, muestra cómo es la interfaz para ratificar o retirar preparadores del sistema.

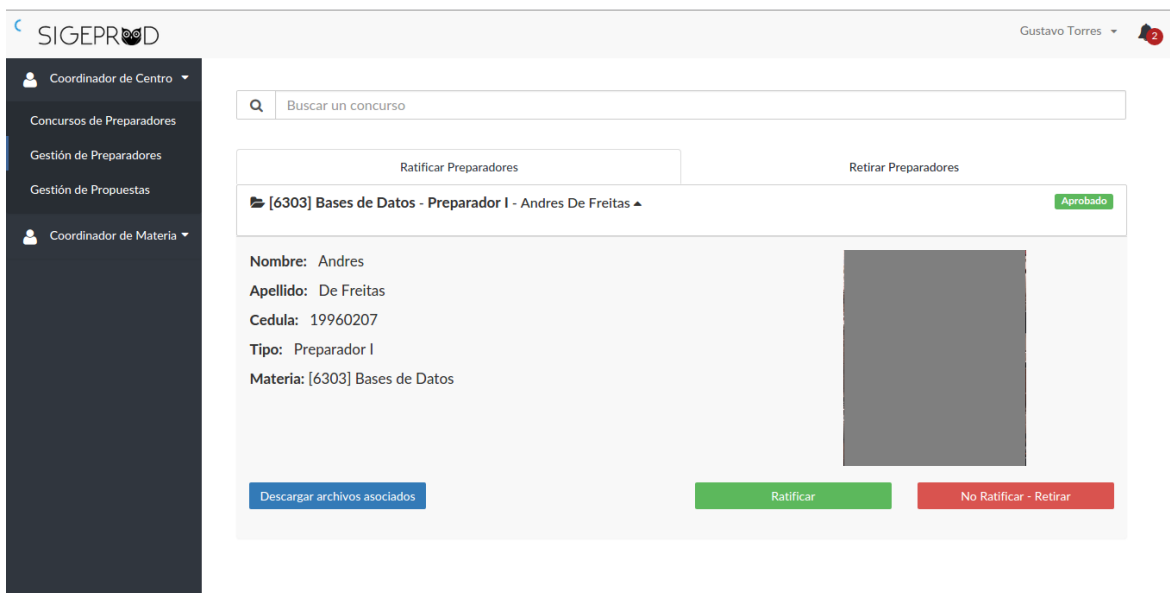


Figura 28. Principales Interfaces - Gestión de preparadores

Las Figuras 29 - 30 - 31 - 32, son las diferentes interfaces que se muestran a lo largo de todo el proceso de registro de un nuevo preparador o auxiliar docente en el sistema.

SIGEPROD

Paso 1 Paso 2 Paso 3 Paso 4

Ingrese sus datos

Cédula de Identidad

Cédula de Identidad

Correo electrónico

Correo electrónico

Continuar

Figura 29. Principales Interfaces - Registro de nuevo preparador. Paso 1: Ingreso de datos

SIGEPROD

Paso 1 Paso 2 Paso 3 Paso 4

Verificación de datos

Nombre

Andres

Apellido

De Freitas

Cédula de Identidad

19960207

Correo electrónico

adefreitas12@gmail.com

Contraseña

Contraseña

Repita su contraseña

Repita su contraseña

Regresar

Continuar

Figura 30. Principales Interfaces - Registro de nuevo preparador. Paso 2: Verificación de datos

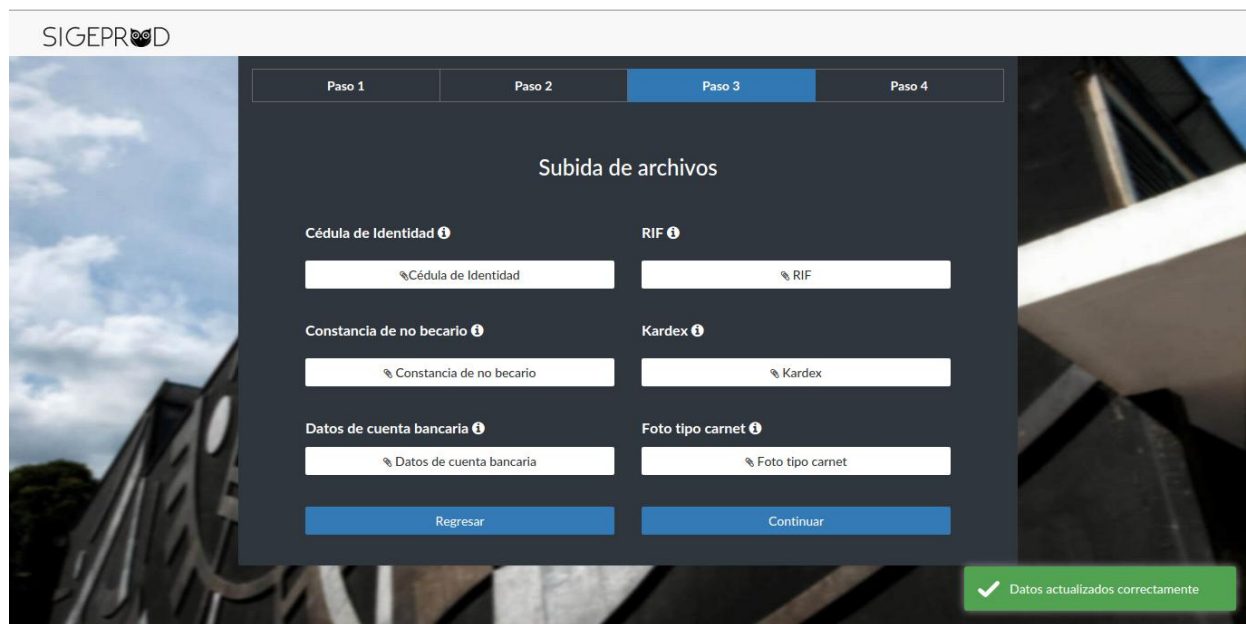


Figura 31. Principales Interfaces - Registro de nuevo preparador. Paso 3: Subida de archivos

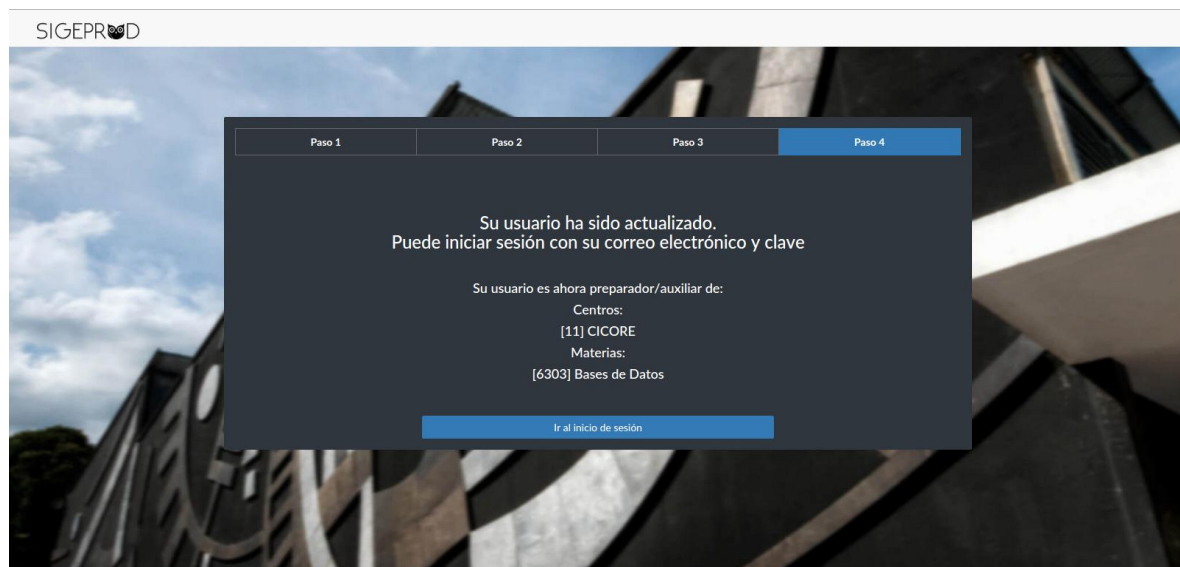


Figura 32. Principales Interfaces - Registro de nuevo preparador. Paso 4: Confirmación

Pruebas funcionales de la aplicación

Para verificar el correcto funcionamiento de la aplicación, se realizaron pruebas de caja negra sobre la misma. Los casos y los resultados de las principales pruebas se presentan en las tablas 13, 14, 15, 16, 17 y 18.

Rango de entradas	Correo	Clave	Resultado esperado	Resultado obtenido
1	Vacío	Vacío	Error: Credenciales incorrectas	Error: Credenciales incorrectas
2	test@ciens.ucv.ve	Vacío	Error: Credenciales incorrectas	Error: Credenciales incorrectas
3	Vacío	Clave	Error: Credenciales incorrectas	Error: Credenciales incorrectas
4	cm111@ciens.ucv.ve	asdf (No valida)	Error: Credenciales incorrectas	Error: Credenciales incorrectas
5	cm111@ciens.ucv.ve	cm111 (Valida)	Éxito: Inicio de sesión de manera correcta	Éxito: Inicio de sesión de manera correcta

Tabla 13. Pruebas funcionales – Inicio de sesión

Coordinador de Materia (Solicitud de Concurso)

Rango de entradas	Materia	Preparador I	Preparador II	Resultados esperados	Resultados obtenidos
1	Vacío	Vacío	Vacío	Mensaje de error: Debe seleccionar una materia	Mensaje de error: Debe seleccionar una materia
2	[6303] Bases de Datos	1	Vacío	Éxito: Solicitud enviada correctamente	Éxito: Solicitud enviada correctamente
3	[6303] Bases de Datos	Vacío	Vacío	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
4	[6303] Bases de Datos	-1	Vacío	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
5	[6303] Bases de Datos	Vacío	-1	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
6	[6303] Bases de Datos	2	Vacío	Éxito: Solicitud enviada correctamente	Éxito: Solicitud enviada correctamente
7	[6303] Bases de Datos	1	-1	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
8	[6303] Bases de Datos	-1	1	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
9	[6303] Bases de Datos	a	1	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente

10	[6303] Bases de Datos	1	a	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
11	Vacío	1	1	Mensaje de error: Debe seleccionar una materia	Mensaje de error: Debe seleccionar una materia

Tabla 14. Pruebas funcionales - Coordinador de materia (solicitud de concurso)

Coordinador de Centro - Solicitud de Concurso

Rango de entradas	Centro	Preparadores I	Preparadores II	Auxiliares Docentes	Resultados esperados	Resultados obtenidos
1	Vacío	Vacío	Vacío	Vacío	Mensaje de error: Debe seleccionar un centro	Mensaje de error: Debe seleccionar un centro
2	[11] CICORE	1	Vacío	Vacío	Éxito: Solicitud enviada correctamente	Éxito: Solicitud enviada correctamente
3	[11] CICORE	Vacío	Vacío	Vacío	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
4	[11] CICORE	Vacío	1	Vacío	Éxito: Solicitud enviada correctamente	Éxito: Solicitud enviada correctamente
5	[11] CICORE	-1	Vacío	Vacío	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
6	[11] CICORE	Vacío	-1	Vacío	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
7	[11] CICORE	2	Vacío	Vacío	Éxito: Solicitud enviada correctamente	Éxito: Solicitud enviada correctamente
8	[11] CICORE	1	-1	Vacío	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
9	[11] CICORE	-1	1	Vacío	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
10	[11]	a	1	Vacío	Mensaje de error: Debe ingresar un número	Mensaje de error: Debe ingresar un número

	CICORE				mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
11	[11] CICORE	1	a	Vacío	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
12	Vacío	1	1	Vacío	Mensaje de error: Debe seleccionar un centro	Mensaje de error: Debe seleccionar una materia
13	[11] CICORE	1	Vacío	1	Éxito: Solicitud enviada correctamente	Éxito: Solicitud enviada correctamente
14	[11] CICORE	Vacío	Vacío	1	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
15	[11] CICORE	Vacío	1	1	Éxito: Solicitud enviada correctamente	Éxito: Solicitud enviada correctamente
16	[11] CICORE	-1	Vacío	1	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
17	[11] CICORE	Vacío	1	-1	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
18	[11] CICORE	Vacío	Vacío	2	Éxito: Solicitud enviada correctamente	Éxito: Solicitud enviada correctamente
19	[11] CICORE	1	1	-1	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
20	[11] CICORE	-1	1	1	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de

					preparador o Auxiliar Docente	preparador o Auxiliar Docente
21	[11] CICORE	a	1	1	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
22	[11] CICORE	1	a	Vacío	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
23	[11] CICORE	1	1	a	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
24	Vacío	1	1	1	Mensaje de error: Debe seleccionar un centro	Mensaje de error: Debe seleccionar un centro

Tabla 15. Pruebas funcionales - Coordinador de centro (solicitud de concurso)

Coordinador de Centro - Edición de Concurso

Rango de entradas	Preparadores I	Preparadores II	Auxiliares Docentes	Resultados esperados	Resultados obtenidos
1	1	Vacío	Vacío	Éxito: Concurso actualizado correctamente	Éxito: Concurso actualizado correctamente
2	Vacío	Vacío	Vacío	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
3	Vacío	1	Vacío	Éxito: Concurso actualizado correctamente	Éxito: Concurso actualizado correctamente
4	-1	Vacío	Vacío	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
5	Vacío	-1	Vacío	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
6	2	Vacío	Vacío	Éxito: Concurso actualizado correctamente	Éxito: Concurso actualizado correctamente
7	1	-1	Vacío	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
8	-1	1	Vacío	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
9	a	1	Vacío	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
10	1	a	Vacío	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente

11	1	Vacío	1	Éxito: Solicitud enviada correctamente	Éxito: Concurso actualizado correctamente
12	Vacío	Vacío	1	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
13	Vacío	1	1	Éxito: Solicitud enviada correctamente	Éxito: Concurso actualizado correctamente
14	-1	Vacío	1	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
15	Vacío	1	-1	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
16	Vacío	Vacío	2	Éxito: Concurso actualizado correctamente	Éxito: Concurso actualizado correctamente
17	1	1	-1	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
18	-1	1	1	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
19	a	1	1	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
20	1	a	Vacío	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
21	1	1	a	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente

Tabla 16. Pruebas funcionales - Coordinador de centro (edición de concurso)

Coordinador de Materia - Edición de Concurso

Rango de entradas	Preparadores I	Preparadores II	Resultados esperados	Resultados obtenidos
1	1	Vacío	Éxito: Concurso actualizado correctamente	Éxito: Concurso actualizado correctamente
2	Vacío	Vacío	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
3	Vacío	1	Éxito: Concurso actualizado correctamente	Éxito: Concurso actualizado correctamente
4	-1	Vacío	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
5	Vacío	-1	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
6	2	Vacío	Éxito: Concurso actualizado correctamente	Éxito: Concurso actualizado correctamente
7	1	-1	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
8	-1	1	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
9	a	1	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
10	1	a	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
11	1	Vacío	Éxito: Solicitud enviada correctamente	Éxito: Concurso actualizado correctamente
12	Vacío	Vacío	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente

13	Vacío	1	Éxito: Solicitud enviada correctamente	Éxito: Concurso actualizado correctamente
14	-1	Vacío	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
15	Vacío	1	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente	Mensaje de error: Debe solicitar al menos un tipo de preparador o auxiliar docente
16	Vacío	Vacío	Éxito: Concurso actualizado correctamente	Éxito: Concurso actualizado correctamente
17	1	1	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
18	-1	1	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
19	a	1	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
20	1	a	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente
21	1	1	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente	Mensaje de error: Debe ingresar un número mayor o igual a ceo en el tipo de preparador o Auxiliar Docente

Tabla 17. Pruebas funcionales - Coordinador de materia (edición de concurso)

Usuario no autenticado - Registro de resultados de concursos - Paso 1				
Rango de entradas	Cédula	Correo	Resultados esperados	Resultados obtenidos
1	Vacío	Vacío	Mensaje de error: Debe especificar su cedula de identidad y correo electrónico	Mensaje de error: Debe especificar su cedula de identidad y correo electrónico
2	1	Vacío	Mensaje de error: Los datos especificados no coinciden con ningún registro	Mensaje de error: Los datos especificados no coinciden con ningún registro
3	Cedula valida	Correo invalido	Mensaje de error: Los datos especificados no coinciden con ningún registro	Mensaje de error: Los datos especificados no coinciden con ningún registro
4	Cedula invalida	Correo valido	Mensaje de error: Los datos especificados no coinciden con ningún registro	Mensaje de error: Los datos especificados no coinciden con ningún registro
5	Cedula valida	Correo valido	Avanza al siguiente paso	Avanza al siguiente paso

Tabla 18. Pruebas funcionales - Usuario no autenticado - Registro de resultados de concursos - Paso 1

RESULTADOS DEL TRABAJO ESPECIAL DE GRADO

El presente Trabajo Especial de Grado consiste en desarrollar el módulo de Gestión de Trámites del Personal Docente Preparador y Auxiliar Docente que, basándose en los procesos de negocios de la Escuela de Computación, permita ejecutar los procesos referentes a los trámites del personal docente preparador y auxiliar docente, a fin de proveer un servicio automatizado y eficaz. El módulo, elaborado en Laravel y AngularJS, aporta las siguientes ventajas:

- Se minimiza el uso del papel ya que el docente no tendría que llevar los documentos requeridos físicamente al Departamento de la Escuela, y estos pueden ser vistos en línea.
- Se facilita el proceso de solicitar concursos.
- El docente no tiene que dirigirse a la Escuela de Computación para poder realizar una solicitud, además puede estar informado del estado de la solicitud y la fase en que se encuentra desde cualquier dispositivo móvil o PC fija con acceso a internet y soporte HTML5, CSS3 y JavaScript.
- El sistema mantendrá un orden y asignación de plazas, ya que cuenta con el manejo de asignación, ratificación y retiro de preparadores.
- El sistema está sujeto a seguir creciendo, agregar nuevas funcionalidades que faciliten las tareas del día a día.

El módulo incluye:

- Manejo de concursos para preparadores y auxiliares docentes.
- Gestión de preparador y auxiliar docente (ratificar y retirar).
- Bitácora de actividades.
- Envío de notificaciones masivas.
- Registro de preparadores.
- Carga y consulta de documentos asociados a preparadores y auxiliares docentes.

CONCLUSIONES Y RECOMENDACIONES

El Trabajo Especial de Grado “Implementación del Módulo de Gestión de Preparadores del Sistema de Gestión de la Programación Docente para el Departamento de la Escuela de Computación de la Facultad de Ciencias” surge como una investigación para desarrollar una posible una solución a la necesidad de actualización tecnológica y carencia de automatización en los servicios que ofrece el Departamento de la Escuela de Computación.

Procesos asociados a los concursos de preparadores y auxiliares docentes no se hacen actualmente de forma automatizada. Esto causa retrasos en el proceso de asignación de preparadores y auxiliares docentes a las distintas materias y centros de la Escuela, y no favorece a que se mantenga un orden en el número de plazas y la cantidad de personas postuladas. Con la realización de este proyecto se buscó centralizar la mayoría de los servicios relacionados con el personal docente preparador y auxiliar docente, que presta el Departamento de Computación.

Para esto se llevó a cabo la implementación de un módulo que pudiera gestionar y manejar vía web las solicitudes de estos concursos, haciendo uso de la metodología Scrum, que permitió la evolución de las distintas funcionalidades a lo largo del proceso de desarrollo a medida que surgían nuevos problemas que resolver, como es el caso de los “ID detalle”, los cuales identifican a un preparador en la nómina de la escuela. Estos no fueron tomados en cuenta al principio del proceso de desarrollo pero, al presentarse la necesidad de agregarlos a la aplicación, el proceso iterativo de Scrum facilitó analizar, diseñar, y desarrollar esta funcionalidad en etapas avanzadas del proceso de desarrollo. También se utilizaron herramientas web de vanguardia, como lo son los *frameworks* Laravel y AngularJS.

Laravel, el *framework* utilizado para el desarrollo de la aplicación del lado del servidor, presentó una curva de aprendizaje más pronunciada de lo esperado, debido a la falta de conocimientos y experiencia previa sobre el mismo por parte del equipo antes del desarrollo de esta aplicación, mientras que la curva de aprendizaje del *framework* del lado del cliente, AngularJS, fue menos pronunciada debido a experiencias previas del equipo con dicho *framework* u otros similares a este.

Se espera que estas funcionalidades permitan al Departamento de Computación facilitar el manejo de las solicitudes y los procesos asociados al personal preparador y auxiliar docente, así como agilizar y optimizar el desarrollo de dichos procesos.

Con la utilización de este sistema, se espera un funcionamiento más eficiente de los servicios del Departamento en lo que se refiere a preparadores y auxiliares docentes.

Como recomendaciones para futuros desarrollos sobre este sistema:

- Permitir que se ingrese al sistema con los datos del correo institucional, administrado por el Centro de Computación.
- Hacer pruebas de seguridad y vulnerabilidad tanto del lado del servidor, como del lado del cliente.

REFERENCIAS BIBLIOGRÁFICAS DIGITALES

Alegsa, L. (2012). *Definición de API*. Recuperado el 7 de Octubre de 2014. Disponible en <http://www.alegsa.com.ar/Dic/api.php>

Cobo, A. (2009). *Diseño y Programación de Bases de Datos*. España. Visión Libros.

EcuRed. (s.f). *Aplicación Web*. Recuperado el 30 de Septiembre de 2014. Disponible en http://www.ecured.cu/index.php/Aplicacion_web

EcuRed. (s.f). *Framework*. Recuperado el 30 de Septiembre de 2014. Disponible en <http://www.ecured.cu/index.php/Framework>

GitHub. (s.f). *Angular/angular.js*. Recuperado el 30 de Septiembre de 2014. Disponible en <https://github.com/angular/angular.js>

Jaramillo, S., Cardona, S. & Villa, D. (2008). *Programación Avanzada en Java*. Colombia. Ediciones Elizcom.

Jeston, J. & Nelis, J. (2008). *BPM: Business Process Management Fundamentos y Conceptos de Implementación*. (Tercera edición). Chile. Bernhard Hitpass.

Llanos, D. (2010). *Fundamentos de informática y programación en C*. España. Editorial Paraninfo.

Maestro de la web. (s.f). *Laravel: ¿Qué es?, Ventajas y ¿Por qué utilizar este Framework?*. Recuperado el 10 de Octubre de 2015. Disponible en <http://www.maestrolaweb.com/blog/desarrollo-web/php/laravel-que-es-ventajas-por-que-utilizar-este-framework>

Mendoza, I. (s.f). *Definición de un Framework para aplicaciones Web con Navegación sensibles a concerns*. Recuperado el 30 de Septiembre de 2014 Disponible en http://sedici.unlp.edu.ar/bitstream/handle/10915/4192/Documento_completo.pdf?sequence=1

Mozilla Developer Network. (s.f). *CSS | MDN*. Recuperado el 26 de Enero de 2015. Disponible en <https://developer.mozilla.org/es/docs/Web/CSS>

Mozilla Developer Network. (s.f). *CSS media queries*. Recuperado el 22 de Octubre de 2014. Disponible en https://developer.mozilla.org/es/docs/CSS/Media_queries

Mozilla Developer Network. (s.f). *JavaScript*. Recuperado el 30 de Septiembre de 2014. Disponible en <https://developer.mozilla.org/es/docs/Web/JavaScript>

PHP. (s.f). *PHP: What is PHP?*. Recuperado el 30 de Septiembre de 2014. Disponible en <http://php.net/manual/en/intro-what-is.php>

Postgresql. (s.f). *Sobre PostgreSQL*. Recuperado el 26 de Enero de 2015. Disponible en http://www.postgresql.org.es/sobre_postgresql

Sass Lang. (s.f). *Sass Reference*. Recuperado el 30 de Septiembre de 2014. Disponible en http://sass-lang.com/documentation/file.SASS_REFERENCE.html

Schwaber, K. & Sutherland, J. (2013). *La guía de Scrum. La guía definitiva de Scrum: Las reglas del juego*. Recuperado el 12 de noviembre de 2014. Disponible en: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>

Scrum Manager. (s.f). *Scrum Manager BoK*. Recuperado el 6 de Octubre de 2014. Disponible en: http://www.scrummanager.net/bok/index.php?title=Main_Page

Sistema Summa. (2011). *Ciclo de vida clásico de los sistemas*. Recuperado el 10 de Octubre de 2015. Disponible en: <http://sistemasumma.com/2011/02/21/ciclo-de-vida-clasico-de-los-sistemas/>

Softeng. (s.f). *Metodología Scrum*. Recuperado el 5 de Octubre de 2014. Disponible en: <http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>

Solís, J. (s.f). *Bootstrap, ¿Qué Es Bootstrap y Cómo Funciona En El Diseño Web?*. Recuperado el 10 de Octubre de 2015. Disponible en: <http://www.arweb.com/chucherias/editorial/%C2%BFque-es-bootstrap-y-como-funciona-en-el-diseno-web.htm>

W3C. (s.f). *HTML & CSS*. Recuperado el 30 de Septiembre de 2014. Disponible en: <http://www.w3.org/standards/webdesign/htmlcss>

WikiMonagas. (s.f). *Metodologías para el desarrollo de software*. Recuperado el 3 de Octubre de 2014. Disponible en: http://wiki.monagas.udo.edu.ve/index.php/Metodolog%C3%ADas_para_el_desarrollo_de_software

Wikipedia. (s.f). *Git*. Recuperado el 30 de Septiembre de 2014. Disponible en <http://es.wikipedia.org/wiki/Git>

ANEXO

A continuación, se muestran los diagramas BPM que ayudaron a modelar la solución para la gestión de los concursos de preparadores y auxiliares docentes, y el formulario de ingreso de un estudiante ganador de un concurso.

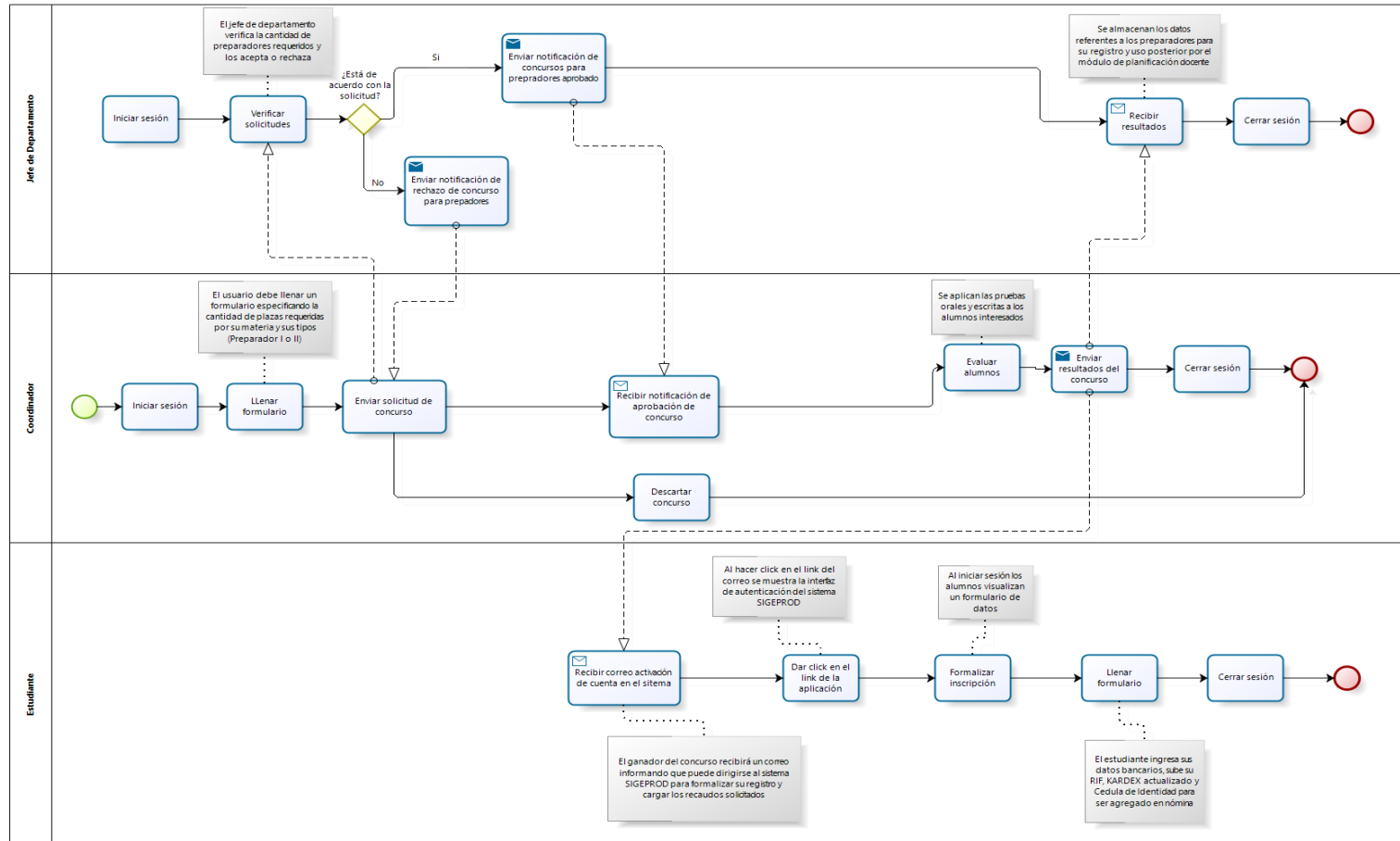


Figura 33: Diagrama BPM Gestión de Personal Preparador

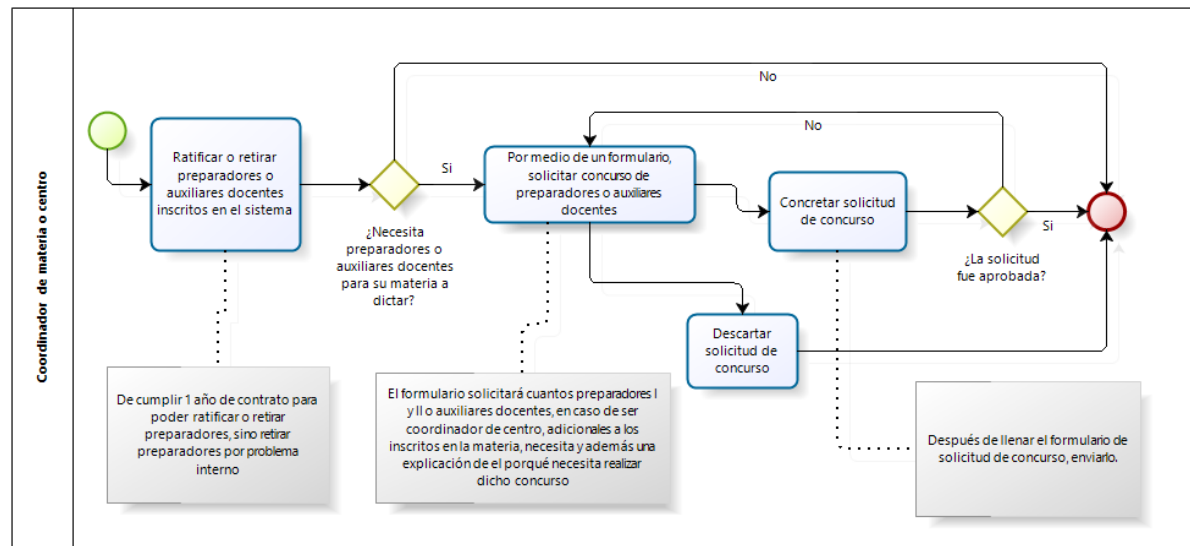


Figura 34: Diagrama BPM Llenar Formulario de Coordinador de Materia

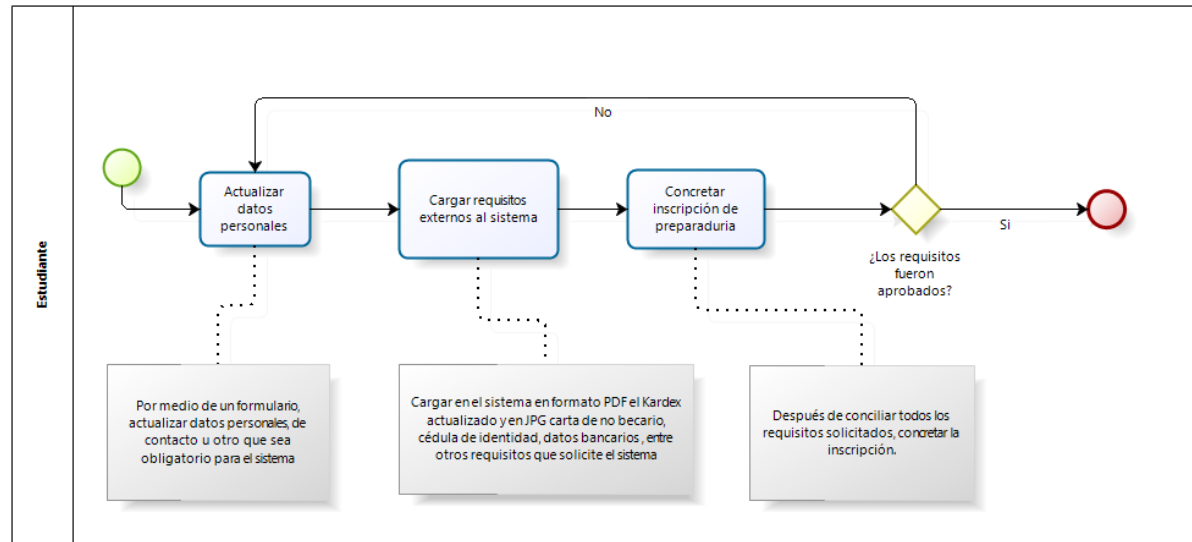


Figura 35: Diagrama BPM Llenar Formulario de Estudiante

La Figura 36, se muestra el esquema de la base de datos definido para la aplicación. Además de las tablas para dar soporte a las funcionalidades que proveerá la aplicación.

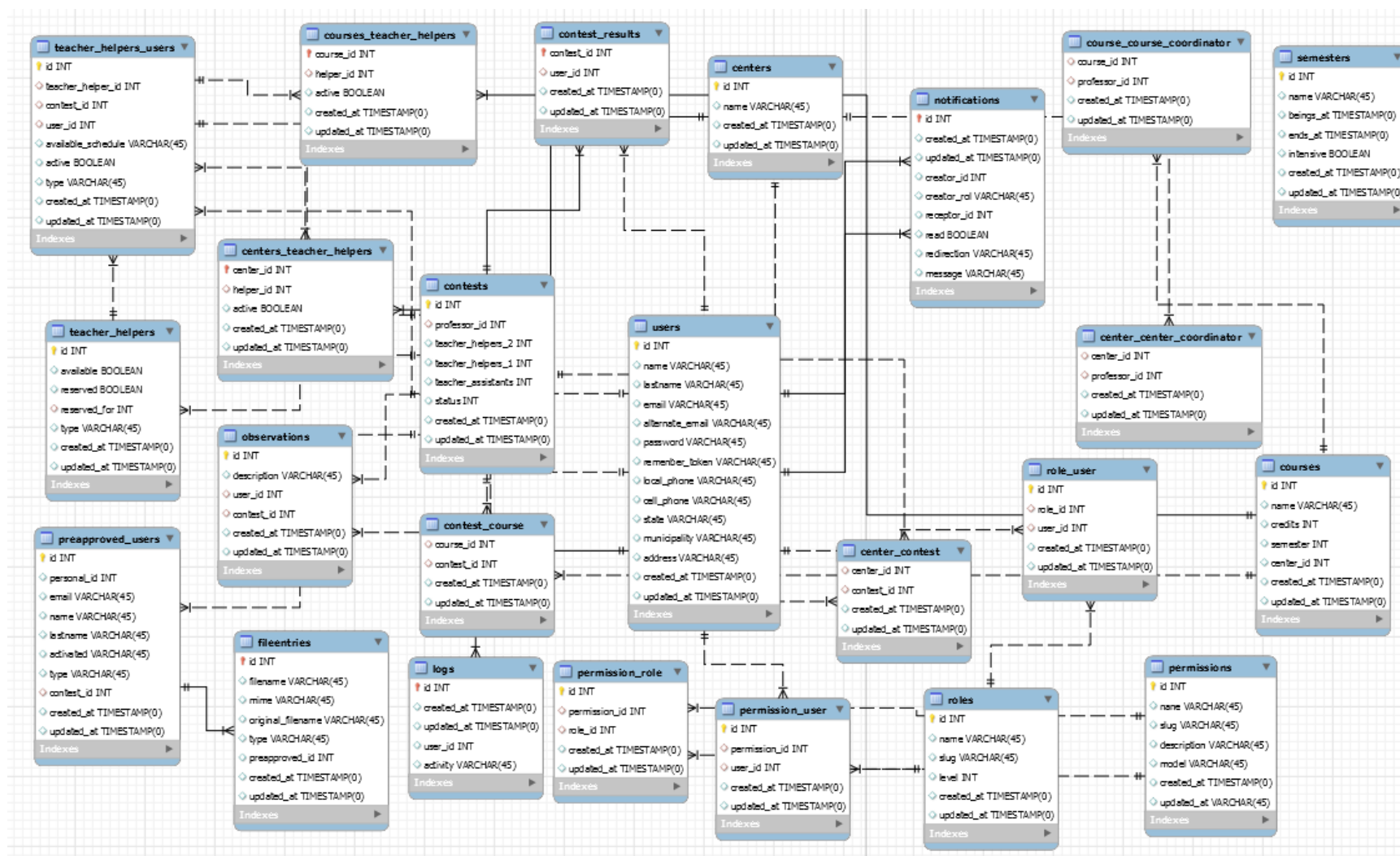


Figura 36: Modelo de base de datos – Gestión de Personal Preparadores