



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN

**Automatización de procesos de
solicitudes docentes y estudiantiles
para el Sistema de Gestión Académica
CONEST**

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

Ysabella R. Carneiro B.
C.I. 19.739.906

para optar al Título de
Licenciada en Computación

Tutores:
Profa. Jossie Zambrano
Prof. Sergio Rivas

Caracas, octubre 2014

ACTA

Quienes suscriben, miembros del jurado designado por el Consejo de Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela, para examinar el Trabajo Especial de Grado titulado: "Automatización de procesos de solicitudes estudiantiles y docentes para el Sistema de Gestión Académica CONEST", presentado por la bachiller Ysabella R. Carneiro B., C.I.: 19.739.906, a los fines de optar por el título de Licenciada en Computación, dejan constancia de lo siguiente:

Dicho trabajo, leído por cada uno de los miembros del jurado, se fijó el día lunes 20 de octubre de 2014, a las 2:30 p.m., para que su autora lo defendiera de forma pública en la Sala de Postgrado de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondió a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de lo cual se levanta la presente acta, en Caracas a los veinte (20) días del mes de octubre del año dos mil catorce (2014), dejando constancia de que actuó como coordinadora del jurado la Profesora Jossie Zambrano.



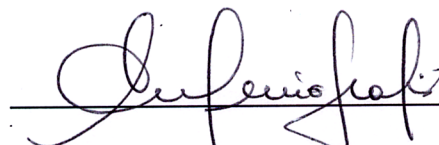
Prof. Jossie Zambrano (Tutor)



Prof. Sergio Rivas (Tutor)



Prof. María Elena Villapol (Jurado)



Prof. Eugenio Scalise (Jurado)

Agradecimientos

En primer lugar, le doy gracias a Dios por haberme permitido alcanzar este logro, proporcionándome la fuerza de voluntad necesaria para superar las dificultades y seguir hasta el final.

A mi familia, en especial a mis padres, por brindarme su comprensión y apoyo incondicional en todo momento.

También agradezco a las personas que me motivaron, de una u otra forma, a tomar decisiones difíciles pero acertadas, que me permitieron continuar y hacer a un lado los obstáculos.

Me gustaría hacer un reconocimiento especial a mis tutores, la Profa. Jossie Zambrano y el Prof. Sergio Rivas, por haber depositado en mí su confianza desde el inicio, por su dedicación, paciencia y disponibilidad a la hora de resolver la infinidad de dudas que surgieron durante el desarrollo de este trabajo, por sus recomendaciones, consejos, así como por motivarme y guiarme durante todo este proceso. Gracias a esta oportunidad adquirí una experiencia invaluable, no pude haber contado con mejores tutores que ustedes. Son merecedores de mi más sincero agradecimiento y aprecio.

Quisiera agradecer también la participación y colaboración del personal de la División de Control de Estudios, a quienes admiro por la ardua labor que realizan a diario, y en especial agradezco al personal de grado por su disposición a la hora de resolver mis inquietudes, y por ayudarme a corregir las situaciones problemáticas que se presentaron durante el desarrollo de este trabajo.

Me siento afortunada de haber tenido la oportunidad de formarme académicamente en la Universidad Central de Venezuela, especialmente en la Facultad de Ciencias, la más hermosa de todas. El presente trabajo es tan sólo una pequeña retribución a mi casa de estudios, y a quienes hacen posible la formación de los futuros profesionales del país.

A todos, les debo mi más profundo agradecimiento por su orientación, colaboración y valioso apoyo durante esta travesía.

Resumen

El presente Trabajo Especial de Grado consiste en el diseño y automatización de procesos relacionados con solicitudes docentes y estudiantiles, entre las que se encuentran las solicitudes de jurado y planillas de notas, solicitudes de modificación de inscripción, abarcando las solicitudes de paralelos, inscripciones extemporáneas y exceso de créditos; incluyendo además las solicitudes de reincorporación. Estos procesos se caracterizan por su complejidad, requiriendo la intervención de diversos roles y el intercambio de información entre los mismos para la gestión de un gran número de solicitudes, las cuales deben ser procesadas ofreciendo tiempos de respuesta apropiados. Por ende, fue necesaria la mejora y simplificación de los procesos administrativos involucrados en el manejo de estas solicitudes, así como la implementación de un sistema flexible, sencillo y usable que brinde soporte a estos procesos. Para el desarrollo de este sistema, se empleó un método de desarrollo iterativo e incremental, aplicando algunas prácticas y técnicas planteadas en metodologías ágiles como Programación Extrema (XP) y AgilUs. El resultado final es un conjunto de funcionalidades para la gestión de solicitudes docentes y estudiantiles, las cuales fueron integradas y puestas en producción en la tercera versión del Sistema de Gestión Académica que presta servicio a la comunidad de la Facultad de Ciencias, conocido como CONEST.

Palabras clave: solicitudes estudiantiles, solicitudes de jurado, reincorporación, modificación de inscripción, División de Control de Estudios, CONEST, *Rails 4*.

Índice general

Introducción	11
1. Solicitudes estudiantiles	19
1.1. Solicitud de jurado y planilla de notas	19
1.2. Solicitudes de modificación de inscripción	22
1.3. Solicitudes de reincorporación	29
2. Tecnologías y herramientas web	31
2.1. Aplicaciones web	32
2.1.1. Arquitectura cliente/servidor	32
2.1.2. Patrón de diseño MVC (Modelo-Vista-Controlador)	34
2.2. Tecnologías empleadas para el desarrollo	35
2.2.1. Tecnologías del lado del cliente	36
2.2.1.1. HTML5	36
2.2.1.2. CSS3	37
2.2.1.3. JavaScript	37
2.2.1.4. JSON	38
2.2.1.5. AJAX	38
2.2.1.6. jQuery y jQuery UI	39
2.2.2. Tecnologías del lado del servidor	40
2.2.2.1. Ruby	40
2.2.2.2. Ruby on Rails	41
2.2.2.3. MySQL	43
2.2.2.4. Git	44
3. Proceso de desarrollo ágil	45
3.1. Manifiesto Ágil	45

Índice general

3.2. Definición del método de desarrollo a utilizar	47
3.2.1. Desarrollo iterativo e incremental	47
3.2.2. Descripción de las fases	49
3.2.2.1. Análisis	49
3.2.2.2. Diseño	50
3.2.2.3. Construcción	51
3.2.2.4. Pruebas	52
3.2.2.5. Despliegue	52
4. Desarrollo del módulo solicitudes	54
4.1. Iteración 1 - Solicitud de jurado para TEG	54
4.1.1. Análisis	55
4.1.1.1. Tormenta de ideas	55
4.1.1.2. Perfiles de usuario	57
4.1.2. Diseño	58
4.1.2.1. Prototipos de interfaz	60
4.1.3. Construcción - Perspectiva del docente	62
4.2. Iteración 2 - Solicitud de jurado para TEG	65
4.2.1. Construcción - Perspectiva del personal de las escuelas	65
4.3. Iteración 3 - Solicitud de jurado para TEG	67
4.3.1. Construcción - Perspectiva del personal de la DCE	68
4.4. Iteración 4 - Solicitud de planilla de notas	69
4.4.1. Construcción - Planilla de notas para seminario	69
4.4.2. Pruebas	71
4.4.3. Despliegue	75
4.5. Iteración 5 - Solicitud de modificación de inscripción	78
4.5.1. Análisis	78
4.5.1.1. Tormenta de ideas	78
4.5.2. Diseño	80
4.5.2.1. Diseño del flujo de comunicación	80
4.5.2.2. Esquema de base de datos	81

Índice general

4.5.2.3. Prototipos de interfaz	83
4.5.3. Construcción - Perspectiva del estudiante	85
4.6. Iteración 6 - Solicitud de modificación de inscripción	88
4.6.1. Diseño	88
4.6.2. Construcción - Perspectiva de la DCE	91
4.6.3. Pruebas	94
4.6.4. Despliegue	96
4.7. Iteración 7 - Solicitud de reincorporación	99
4.7.1. Análisis	99
4.7.2. Diseño	100
4.7.2.1. Esquema de base de datos	100
4.7.3. Construcción	101
4.7.4. Pruebas	104
4.7.5. Despliegue	104
Conclusiones y recomendaciones	105
Referencias	108
Anexos	111

Índice de figuras

1.1. Proceso de solicitud de jurado.	20
1.2. Solicitudes estudiantiles realizadas en la primera versión de CONEST.	24
1.3. Sugerencias para solicitudes estudiantiles en la primera versión de CONEST. . .	24
1.4. Proceso de modificación de inscripción.	25
1.5. Tendencia de las solicitudes estudiantiles realizadas a través de CONEST entre los años 2010 y 2013.	27
1.6. Proceso de reincorporación.	30
2.1. Arquitectura Tres-Capas.	33
2.2. Patrón Modelo-Vista-Controlador.	34
2.3. Estructura básica de una aplicación web.	35
2.4. Comparación de formato <i>JSON</i> vs <i>XML</i>	38
2.5. Tecnologías agrupadas bajo el concepto de <i>AJAX</i>	39
2.6. Tecnologías utilizadas para el desarrollo de la aplicación.	44
3.1. Modelo iterativo e incremental.	48
3.2. Ciclo de vida y actividades del modelo de desarrollo empleado.	53
4.1. Tormenta de ideas para las solicitudes de jurado.	56
4.2. Gestión de solicitudes de jurado para el docente.	59
4.3. Gestión de solicitudes de jurado para el personal de las escuelas.	59
4.4. Gestión de solicitudes de jurado para el personal de la DCE	60
4.5. Conjunto de íconos seleccionados para la interfaz de usuario.	60
4.6. Prototipo de interfaz de consulta de solicitudes de jurado para el personal de la DCE.	61

Índice de figuras

4.7. Prototipo de interfaz de consulta de solicitudes de jurado para el personal de las escuelas.	62
4.8. Formulario para realizar la solicitud de jurado.	64
4.9. Interfaz de consulta de solicitudes de jurado para el docente.	65
4.10. Consulta de solicitudes de jurado para el personal de las escuelas.	66
4.11. Aprobación de solicitud de jurado.	66
4.12. Reenvío de correos con información de la solicitud aprobada.	67
4.13. Consulta de solicitudes de jurado para el personal de la DCE.	68
4.14. Interfaz para la generación de constancias de tutoría para seminario o TEG. . .	69
4.15. Ventana modal para solicitar la planilla de notas para seminario.	70
4.16. Consulta de planillas de notas para seminario.	70
4.17. Escenario de prueba Cucumber - Realizar solicitud de jurado.	72
4.18. Escenario de prueba Cucumber - Aprobar solicitudes.	73
4.19. Escenario de prueba Cucumber - Anular solicitudes.	73
4.20. Solicitudes de jurado y planillas de notas procesadas en el semestre 02-2013. .	77
4.21. Tormenta de ideas para las solicitudes de modificación de inscripción.	79
4.22. Proceso adaptado para las solicitudes de modificación de inscripción.	81
4.23. Esquema de base de datos para el gestor de reglas de las solicitudes de modificación de inscripción.	82
4.24. Prototipo de interfaz del estudiante para realizar solicitudes de modificación de inscripción.	83
4.25. Prototipo de planilla de solicitud de modificación de inscripción.	84
4.26. Interfaz inicial del estudiante para solicitudes de modificación de inscripción. . .	85
4.27. Métodos del gestor de reglas de modificación de inscripción.	86
4.28. Método que permite la validación de una regla general de la normativa de solicitudes estudiantiles.	87
4.29. Observaciones proporcionadas por el gestor de reglas de modificación de inscripción.	87

Índice de figuras

4.30. Interfaz del estudiante para descargar la planilla de modificación de inscripción.	88
4.31. Prototipo de interfaz de consulta de solicitudes de modificación de inscripción para la DCE.	89
4.32. Prototipo de interfaz en papel para el gestor de reglas.	89
4.33. Estructura del objeto <i>JSON</i> para la descripción de valores de las reglas de solicitudes estudiantiles.	90
4.34. Ejemplo de objeto <i>JSON</i> para la descripción de valores de una regla.	90
4.35. Interfaz de consulta de solicitudes de modificación de inscripción para la DCE. .	91
4.36. Ventana modal para la aprobación o rechazo de solicitudes de modificación de inscripción.	92
4.37. Interfaz del gestor de reglas de modificación de inscripción.	92
4.38. Editar valores asociados a una regla de modificación de inscripción.	93
4.39. Gestor de configuración para el proceso de modificación de inscripción.	94
4.40. Tendencia de las solicitudes modificación de inscripción realizadas a través de CONEST en el semestre 01-2014.	97
4.41. Esquema de base de datos para el gestor de reglas de solicitudes de reincorporación.	100
4.42. Interfaz gráfica del estudiante para realizar solicitudes de reincorporación.	101
4.43. Métodos que implementan las reglas de reincorporación.	102
4.44. Interfaz gráfica del gestor de reglas de reincorporación.	102
4.45. Interfaz de consulta de solicitudes de reincorporación para la DCE.	103

Índice de tablas

1.1. Cantidad de solicitudes estudiantiles realizadas a través de CONEST entre los años 2010 y 2013.	26
4.1. Perfil de usuario - Estudiante de pregrado.	57
4.2. Perfil de usuario - Docente.	57
4.3. Perfil de usuario - Personal de las escuelas.	57
4.4. Perfil de usuario - Personal de la DCE.	58
4.5. Cantidad de solicitudes de jurado y planillas de notas procesadas por carrera en el semestre 02-2013.	76
4.6. Cantidad de solicitudes de modificación de inscripción realizadas mediante CONEST 3.0 en el semestre 01-2014.	97

Introducción

Los Sistemas de Gestión Académica, generalmente implementados como aplicaciones web en muchas casas de estudio a nivel nacional, ofrecen un apoyo fundamental a los procesos estudiantiles, docentes y administrativos, a través del almacenamiento, organización y procesamiento de la información concerniente al complejo ámbito académico, automatizando y agilizando los procesos con el objeto de prestar un mejor servicio a la comunidad.

Entre las ventajas de emplear aplicaciones web para la gestión académica se tiene la alta disponibilidad, seguridad y confiabilidad del servicio, permitiendo atender una gran cantidad de tareas y peticiones de los usuarios en un tiempo reducido, muchas veces sin requerir el desplazamiento físico de los mismos hasta la institución. Estos sistemas también permiten la planificación y seguimiento de actividades docentes y administrativas, proporcionando además información relevante para la toma de decisiones.

La Facultad de Ciencias de la Universidad Central de Venezuela, que cuenta con una matrícula a nivel de pregrado de más de 3000 estudiantes y más de 600 docentes, ha promovido el uso y desarrollo de su propio Sistema de Gestión Académica, conocido como CONEST, el cual es un sistema basado en una plataforma de software libre, que ha sido construido con el aporte de estudiantes, docentes y personal administrativo a lo largo del tiempo. La primera versión de este sistema fue puesta en producción en el año 2007, con el propósito de automatizar y dar soporte a las actividades llevadas a cabo por la División de Control de Estudios (DCE). Este organismo es el encargado de gestionar todos los procesos académicos de la Facultad, tales como inscripción, asignación de horarios y aulas, gestión de constancias y solicitudes estudiantiles, calificación y grado, entre otros.

CONEST también ofrece servicios a las cinco escuelas que conforman a la Facultad, las cuales son Biología, Computación, Física, Matemática y Química, cada una de las cuales constituye una carrera, a excepción de la Escuela de Química que dicta las carreras Química y Geoquímica. Además, este sistema automatiza actividades relacionadas con otras dependencias, por lo que se ha convertido en un sistema crítico para el funcionamiento adecuado de la Facultad.

Durante sus inicios, CONEST sufrió un crecimiento desmesurado para soportar una amplia variedad de procesos, llegando a convertirse en un sistema desorganizado, con funcionalidades redundantes que lo hacían extremadamente complejo, difícil de actualizar, extender y mantener. Esta experiencia determinó la necesidad de evolucionar el sistema para solventar sus inconvenientes.

Introducción

Desde el año 2013 se ha estado desarrollando la versión más reciente del sistema CONEST, la versión 3.0, con la visión de reconstruir el sistema por completo, mejorando su diseño, tanto a nivel de arquitectura, como a nivel de base de datos, para resolver las deficiencias y fallas presentes en la versión anterior. Además, se integran nuevas tecnologías y buenas prácticas de desarrollo para construir una aplicación actualizada que preste un servicio más atractivo y eficiente.

CONEST 3.0 nace con el propósito de contar con un sistema que mejore los servicios brindados por la versión anterior a los distintos usuarios, que soporte los diferentes y complejos procesos académicos, optimizando los recursos disponibles, reduciendo costos, ofreciendo a su vez un alto rendimiento. También se busca que el sistema sea escalable, extensible y mantenible con facilidad a lo largo del tiempo, y que además sea lo suficientemente flexible para adaptarse a las necesidades de los entes que hacen vida en la facultad.

Entre las mejoras más resaltantes introducidas en esta versión, se tiene la posibilidad de que los estudiantes seleccionen el horario más conveniente a la hora de realizar la inscripción de asignaturas previo al inicio del semestre. Esto permite reducir los conflictos de horario entre las asignaturas cursadas conjuntamente, y reducir las solicitudes de cambio de horario, entre otros beneficios. De igual manera, esta versión busca ser más comunicativa y brindar más retroalimentación a los usuarios, especialmente a los estudiantes, con el objeto de que los mismos puedan solventar las dudas más comunes con la ayuda del sistema, ofreciendo una experiencia de usuario más agradable y satisfactoria. Otro objetivo primordial de la nueva versión es apoyar los procesos de la DCE y de las distintas escuelas, ajustándose a sus particularidades para mejorar la productividad, en vez de forzar la adaptación o restringir a los usuarios en la forma de realizar sus labores.

Para la puesta en producción de CONEST 3.0 fue necesaria la reconstrucción de las funcionalidades más importantes a fin de garantizar su funcionamiento inicial, y en lo que respecta al presente Trabajo Especial de Grado, estas funcionalidades están relacionadas con la gestión de un conjunto de solicitudes docentes y estudiantiles, tales como solicitudes de jurado y planillas de notas, solicitudes de modificación de inscripción, y solicitudes de reincorporación. Algunas de ellas han sido automatizadas en la versión anterior de CONEST, pero además de su reconstrucción e integración en la nueva versión, requieren ser mejoradas ya que presentan algunas deficiencias.

Las limitaciones o problemas derivados de la versión anterior del sistema requieren un análisis y rediseño de los procesos subyacentes para su resolución efectiva. Existen actividades inherentes a la propia automatización de procesos extensos, constituidos por muchos pasos, que incrementan aun más su complejidad. Es necesario adaptar y simplificar los flujos de información, modificando el comportamiento del proceso para hacerlo más simple, ya que así es posible modelar y construir un sistema que implemente esta simplicidad.

Introducción

En general, la situación problemática está dada por las diferencias particulares de las escuelas a la hora de llevar a cabo los procesos relacionados con la gestión de solicitudes docentes y estudiantiles. Esto quiere decir que los procesos no están estandarizados y se experimenta cierta desorganización. La comunicación frecuente que deben realizar las cinco escuelas con la División de Control de Estudios, a través del constante flujo de documentos, trae como consecuencia que el personal de este organismo deba realizar un esfuerzo adicional para adaptarse a la forma de trabajo de cada una de ellas.

Realizar la unificación de los procesos y estandarizarlos para todas las escuelas no es una tarea sencilla. Existen actividades que no pueden ser modificadas o descartadas porque forman parte integral del proceso particular de cada escuela, o de la misma facultad, pero siempre es posible realizar mejoras y lograr cierta uniformidad general, con el fin de facilitar y mejorar la comunicación con la DCE, ya que es el organismo principal que recibe y procesa la información enviada por todas las escuelas. Para lograrlo, es necesario proponer un flujo de trabajo que contribuya a resolver las fallas comunes y simplifique las tareas de los usuarios.

Respecto a la problemática específica que se aborda en este trabajo, se tienen dos vertientes: la automatización de procesos realizados actualmente de forma manual, y la mejora de aquellos que ya han sido objeto de automatización previa, pero que siguen presentando deficiencias. A continuación se aborda la primera vertiente.

A pesar de que la primera versión de CONEST automatiza una gran cantidad de procesos, existen algunos que siguen realizándose de forma tradicional, tales como las solicitudes de jurado. Estas solicitudes son muy importantes, ya que están relacionadas con el delicado proceso de graduación y egreso de los estudiantes de la facultad.

La solicitud de jurado consiste en la proposición, por parte de los tutores académicos, y posterior aprobación por parte del Consejo de Escuela, de los docentes que se desempeñarán como jurado durante la presentación del Trabajo Especial de Grado (TEG) de los estudiantes. La gestión no automatizada de este proceso conlleva ciertas desventajas, tales como:

- Recepción, gestión y procesamiento manual de un alto número de solicitudes, tanto para las escuelas, como para la DCE.
- Necesidad de transcribir, imprimir y fotocopiar frecuentemente los documentos o memorandos, tanto los de uso interno para la escuela, como los dirigidos a la DCE. Esto ocasiona altos gastos de recursos materiales, sin olvidar el tiempo invertido en la transcripción de información con formatos diferentes entre las escuelas. Esta información debería estar estandarizada y disponible en el sistema de forma oportuna para los roles pertinentes.
- Dificultades en la notificación de respuestas y decisiones. Esto implica que el personal debe invertir tiempo para realizar la entrega personal de las comunicaciones a los involucrados.

Introducción

- La solicitud de planilla de notas, documento donde se registra la calificación obtenida por los tesisistas después de la presentación de los TEGs, está estrechamente relacionada con las solicitudes de jurado, ya que son los miembros del mismo quienes determinan esta calificación y deben firmar esta planilla para dar constancia de ello. No obstante, esta información no está integrada, por lo que se corre el riesgo de cometer errores y tener información inconsistente al manejar un alto número de solicitudes de forma manual. Este punto es delicado, porque se debe garantizar que el jurado aprobado y el que aparece registrado en la planilla de notas coincide, a fin de que el proceso sea veraz y transparente.

Conforme a lo expuesto anteriormente, se desea realizar la automatización de este proceso e integrarlo en CONEST 3.0, con la finalidad de agilizar los tiempos de respuesta, minimizar los costos y garantizar la consistencia de la información.

En el caso de los procesos con antecedentes de automatización, como segunda vertiente, son de especial relevancia para el presente trabajo las solicitudes de modificación de inscripción. Este es un proceso crítico para los estudiantes, ya que les permite modificar, al inicio de cada semestre lectivo, el conjunto de asignaturas previamente inscritas. Los estudiantes tienen la necesidad de realizar estas solicitudes por diversos motivos, ya sea para sobrepasar las restricciones impuestas en el proceso de inscripción en cuanto al número de asignaturas y validación de prelacones, con el objeto de inscribir más asignaturas y adelantar sus estudios; también puede ser a causa de errores cometidos en la inscripción, o para inscribir asignaturas que no fueron ofertadas en ese proceso.

Cada semestre se produce un alto número de solicitudes estudiantiles que deben ser estudiadas en detalle por diversos entes para su aprobación o rechazo. Debido a su naturaleza, se debe ofrecer respuestas de manera oportuna para que los estudiantes estén al tanto del estado de sus solicitudes y puedan cursar con seguridad las asignaturas solicitadas.

Para dar respuesta a los cientos de solicitudes generadas, las unidades de asesoramiento académico (UAA) de las escuelas deben realizar un análisis de la situación académica del estudiante, y verificar el cumplimiento de la normativa que rige a las solicitudes estudiantiles, con la finalidad de determinar la idoneidad de las peticiones y emitir sugerencias al Consejo de Escuela, quien determina las decisiones finales. CONEST, en su primera versión, apoya este proceso proporcionando la información pertinente al personal de las UAA, pero obvia por completo al estudiante, quien debería conocer de antemano la viabilidad de sus peticiones. Entonces, se necesita facilitar el análisis de las solicitudes brindando la información que ya está disponible en el sistema, de forma completa y organizada, al personal encargado de su procesamiento, permitiendo que el estudiante esté al tanto de la factibilidad y de los criterios que serán tomados en cuenta para la aprobación o rechazo de sus peticiones.

Introducción

Todas las solicitudes estudiantiles se rigen por una normativa que puede ser global a la facultad, o presentar ciertas variaciones en las diversas escuelas, y que además puede sufrir actualizaciones con el tiempo. En la versión anterior de CONEST, muchas de estas variaciones no fueron modeladas, por lo cual existe un incumplimiento de estas reglas. Aquellas que sí fueron incluidas, no son modificables a través del sistema, por lo que corren el riesgo de quedar desactualizadas si surgen cambios, generando un impacto significativo en la gestión de solicitudes. Por lo tanto, es imperativo incorporar de forma efectiva el reglamento y facilitar su actualización, para apoyar el cumplimiento de las normas y contribuir al funcionamiento correcto de los procesos.

Sin embargo, también es necesario realizar una reestructuración de las actividades y tareas que constituyen el proceso de modificación de inscripción, ya que la experiencia previa indica que no es lo suficientemente eficiente, sufriendo de constantes retrasos por la secuencialidad de los pasos, implicando una alta carga de trabajo para el personal involucrado. Esta referida secuencialidad de pasos es bastante estricta, por lo que si una acción no es completada, situación que ocurre comúnmente en las escuelas, se requiere que el personal a cargo resuelva el estancamiento para que el proceso pueda continuar.

La DCE, que posee el rol administrativo más alto, debería poder emplear el sistema, con los privilegios que le corresponden, para agilizar la resolución de estos conflictos. Esto no podía hacerse anteriormente porque el sistema no contaba con estas facilidades. Además, en caso de incorporarse personal nuevo al flujo de proceso, se requería de un entrenamiento previo para el manejo del sistema, lo que implicaba sobrellevar un proceso de aprendizaje, que en la versión anterior de CONEST resultaba engorroso debido a sus carencias de usabilidad.

Por otra parte, se tiene la gestión de solicitudes de reincorporación, las cuales están relacionadas con aquellos estudiantes que se encuentran desincorporados, ya sea por estar incursos en el reglamento de las normas de permanencia, o por deserción de sus estudios, y que desean solicitar la reincorporación a la facultad. Estas solicitudes también suelen ser numerosas, deben cumplir una normativa, y requieren un amplio estudio de los casos estudiantiles por parte de varias dependencias, a fin de determinar la aptitud del estudiante para reanudar sus estudios. Las solicitudes de reincorporación requieren un enfoque similar al de las solicitudes de modificación de inscripción en cuanto a su automatización, especialmente para contribuir al cumplimiento de las normas y apoyar la toma de decisiones.

Tomando en cuenta todos estos aspectos, se tiene que además de construir una aplicación usable y eficiente, el diseño y la automatización de estos procesos debe realizarse de forma tal que el sistema final sea adaptable y capaz de soportar los cambios a lo largo del tiempo. Si se debe seguir una secuencia de pasos numerosa y demasiado estricta, todo el proceso es propenso a quedarse estancado al ocurrir conflictos en alguna parte del mismo, independientemente del buen funcionamiento de la aplicación que fue modelada en base a ese proceso. Esto

Introducción

trae como consecuencia una especie de círculo vicioso, porque al tener la voluntad de mejorar o cambiar el proceso subyacente, se presentan dificultades debido a que el funcionamiento del sistema depende de él. Entre otros efectos relevantes a tomar en cuenta en estos casos está el entorpecimiento a la fluidez de las actividades, impacto en la productividad, aumento de esfuerzos, retrasos, frustración de los usuarios, y disminución de la calidad del servicio.

En tal sentido, esta investigación tiene como objetivo general desarrollar una aplicación web que soporte los procesos relacionados con las solicitudes estudiantiles y docentes, para su integración y puesta en producción en el sistema CONEST.

Los objetivos específicos planteados para alcanzar el objetivo general, tomando en cuenta la problemática descrita en párrafos anteriores, son los siguientes:

- Automatizar el proceso de solicitud de jurado para Trabajo Especial de Grado y seminario, con el propósito de reducir la inversión de tiempo, esfuerzo, recursos humanos y materiales para su gestión, integrándolo a su vez con las solicitudes de planillas de notas para garantizar la consistencia de la información.
- Diseñar y automatizar el flujo de comunicación del proceso de solicitudes de modificación de inscripción, con la finalidad de flexibilizarlo y reducir los costos en materia de tiempo, esfuerzo y recursos.
- Automatizar el proceso de solicitudes de reincorporación, ofreciendo a su vez información relevante para el estudio de los casos y agilizar así su procesamiento.
- Aplicar la normativa referente a las solicitudes estudiantiles de modificación de inscripción para cada escuela en el sistema CONEST, así como el reglamento que rige a las solicitudes de reincorporación, para ofrecer una validación y retroalimentación adecuada, informando oportunamente al estudiante sobre la viabilidad de sus peticiones, y facilitando a la vez el análisis y la toma de decisiones por parte de las unidades de asesoramiento académico.
- Desarrollar un gestor para la administración de reglas relacionadas con las solicitudes de modificación de inscripción y solicitudes de reincorporación, de forma que puedan ser ajustadas en base a los cambios en la normativa a lo largo del tiempo, con el fin de brindar flexibilidad, unificar el sistema a nivel de las diferentes escuelas, y favorecer el cumplimiento de la normativa inherente a estas solicitudes.

La consecución de estos objetivos permitiría solventar una problemática importante y recurrente, que tiene un alto impacto sobre los estudiantes, docentes y personal administrativo, especialmente para el que labora en la División de Control de Estudios debido a su fuerte carga de trabajo.

Introducción

Uno de los servicios más demandados por los estudiantes, que son el grupo más numeroso de la comunidad académica, es la gestión de solicitudes estudiantiles. Cada semestre se reciben y procesan cientos de solicitudes de diversos tipos, las cuales deben ser gestionadas en el menor tiempo posible para dar respuesta oportuna a las peticiones de los estudiantes. Para llevar a cabo este proceso, se requiere una variedad de actores involucrados, tales como docentes, personal administrativo de las escuelas conformado por secretarías, diferentes comisiones y unidades de asesoramiento académico, incluyendo a los directores, así como el personal administrativo de la División de Control de Estudios, quienes hacen su mejor esfuerzo para realizar sus labores y hacer cumplir las normativas, sobrellevando las deficiencias derivadas de la problemática actual.

Desde hace varios años se ha estado haciendo un esfuerzo mancomunado para mejorar los múltiples servicios prestados por los diversos entes que hacen vida en la facultad, especialmente para reducir la fuerte carga de trabajo y mejorar la comunicación entre los mismos a través de CONEST, aprovechando las tecnologías disponibles actualmente con el fin de solventar las deficiencias. De esto se deriva la motivación del presente trabajo, el cual proporciona soluciones a un conjunto de situaciones susceptibles a ser mejoradas, mediante el rediseño de procesos complejos, la automatización del trabajo manual, la integración de la información y el desarrollo de herramientas para apoyar la toma de decisiones y el cumplimiento de los reglamentos, con la finalidad de obtener una serie de beneficios, tales como:

- Estandarizar y agilizar la comunicación entre los entes involucrados.
- Flexibilizar los procesos para permitir su adaptación con el paso del tiempo.
- Facilitar el estudio de solicitudes, ofreciendo información oportuna y relevante.
- Simplificar la gestión de solicitudes y disminuir el esfuerzo invertido en su procesamiento.
- Mejorar los tiempos de respuesta.
- Garantizar la consistencia de la información.
- Reducir los costos materiales derivados de la gestión administrativa, minimizando en particular el material impreso.
- Facilitar el trabajo y mejorar el desempeño del personal administrativo en general.

Todo esto se traduce en la mejora de los servicios y en un incremento de la satisfacción de los usuarios: estudiantes, docentes, personal administrativo de las escuelas y de la División de Control de Estudios, lo cual genera un impacto positivo para la comunidad de la Facultad de Ciencias.

Introducción

Para culminar esta sección preliminar, se describe a continuación la estructura del presente Trabajo Especial de Grado, el cual se encuentra organizado de la siguiente manera:

- En el Capítulo 1 - Solicitudes estudiantiles, se describen las solicitudes estudiantiles que son objeto de estudio: las solicitudes de jurado y planilla de notas, las solicitudes de modificación de inscripción, y las solicitudes de reincorporación, explicando los procesos asociados a las mismas, detallando las características más relevantes, resaltando su importancia y las principales dificultades o deficiencias específicas derivadas de la situación actual.
- El Capítulo 2 - Tecnologías y herramientas web, tiene como propósito explicar los conceptos básicos relacionados con el funcionamiento y la implementación de las aplicaciones web. Está dividido en dos secciones principales: la primera trata sobre las aplicaciones web y su relación con la arquitectura cliente/servidor, incluyendo una descripción del patrón de diseño Modelo-Vista-Controlador (MVC). La segunda sección ofrece una reseña del compendio de tecnologías del lado cliente y del lado del servidor, así como las herramientas web seleccionadas para la construcción de la solución propuesta al problema planteado.
- El Capítulo 3 - Proceso de desarrollo ágil, tiene como finalidad presentar el proceso aplicado para la implementación de la aplicación. Como preámbulo, se reseña el Manifiesto Ágil, el cual contiene una serie de principios que han sido heredados por los métodos ágiles existentes actualmente. Posteriormente, se procede a definir el método de desarrollo iterativo e incremental seleccionado, así como cada una de las fases y actividades que lo constituyen.
- El Capítulo 4 - Desarrollo del módulo solicitudes, corresponde a la descripción detallada de las iteraciones que se llevaron a cabo durante el ciclo de desarrollo, producto de la aplicación del método de desarrollo referido anteriormente para la resolución del problema de investigación. En este capítulo, se explican las actividades realizadas en las fases de Análisis, Diseño, Construcción y Pruebas, así como los resultados obtenidos en la fase de Despliegue, para el conjunto de funcionalidades desarrolladas en cada iteración.

Finalmente, se presentan las conclusiones y recomendaciones derivadas de este trabajo, consolidando los resultados y la experiencia general obtenida a lo largo del desarrollo. Además, se presentan las referencias que sustentan la investigación, y un conjunto de anexos que complementan la información expuesta a lo largo del documento.

Capítulo 1. Solicitudes estudiantiles

Durante el desarrollo de sus estudios en la facultad, los estudiantes tienen la necesidad de realizar una variedad de peticiones a los distintos entes de la institución para solventar diversas situaciones. El propósito de este capítulo es describir en detalle los procesos involucrados en la gestión de las solicitudes que son objeto de estudio, resaltar sus características más relevantes, y señalar las carencias o deficiencias asociadas a estos procesos para proporcionar una visión clara y específica de la problemática actual.

La estructura de este capítulo se divide en tres secciones, las cuales corresponden a las solicitudes de jurado y planillas de notas, solicitudes de modificación de inscripción, y solicitudes de reincorporación, las cuales se describen a continuación.

1.1. Solicitud de jurado y planilla de notas

La solicitud de jurado consiste en solicitar al Consejo de Escuela, a través del tutor académico, la aprobación de un conjunto de docentes que conformarán el jurado evaluador de la presentación o defensa del Trabajo Especial de Grado de un estudiante. Algunas escuelas también realizan este proceso para la evaluación de seminarios.

El flujo de proceso involucrado en la solicitud de jurado para TEG varía un poco dependiendo de las escuelas, sin embargo, se toma como referencia el procedimiento llevado a cabo en la Escuela de Computación.

El proceso inicia cuando el tutor académico, en representación del estudiante, llena una planilla de solicitud de nombramiento de jurado, la cual debe contener los datos del estudiante, el título del trabajo, resumen y palabras clave del mismo, así como los nombres de los docentes propuestos para conformar el jurado que evaluará dicho trabajo. El jurado está compuesto por el tutor firmante, dos jurados principales y dos jurados suplentes. Esta planilla es entregada físicamente por el tutor académico en la escuela correspondiente, junto con el expediente curricular (*kardex*) del estudiante. En algunas escuelas, las solicitudes son evaluadas por una comisión de grado antes de que el proceso continúe.

Posteriormente, la solicitud es discutida en Consejo de Escuela, el cual es el ente encargado de asignar y aprobar el jurado propuesto por el tutor académico. Esta asignación se realiza en base a la disponibilidad de los docentes propuestos, área de conocimiento y experiencia de los mismos.

El Consejo de Escuela puede realizar modificaciones a los datos de la solicitud de jurado, tales como el título del trabajo, así como cambiar los docentes que conformarán el jurado evaluador del trabajo correspondiente. Cuando la solicitud es aprobada, el personal de las escuelas debe redactar una carta de aprobación con los datos finales de la solicitud de jurado para imprimir y entregar la copia correspondiente a cada miembro del jurado, es decir, se generan entre cinco y seis copias por solicitud. De igual manera, el Jefe de la División de Control de Estudios es notificado de la aprobación a través de un memorándum que refleja estos datos, con el fin de que esta información conste en el expediente del estudiante. La figura 1.1 ilustra de forma general el proceso anteriormente descrito.

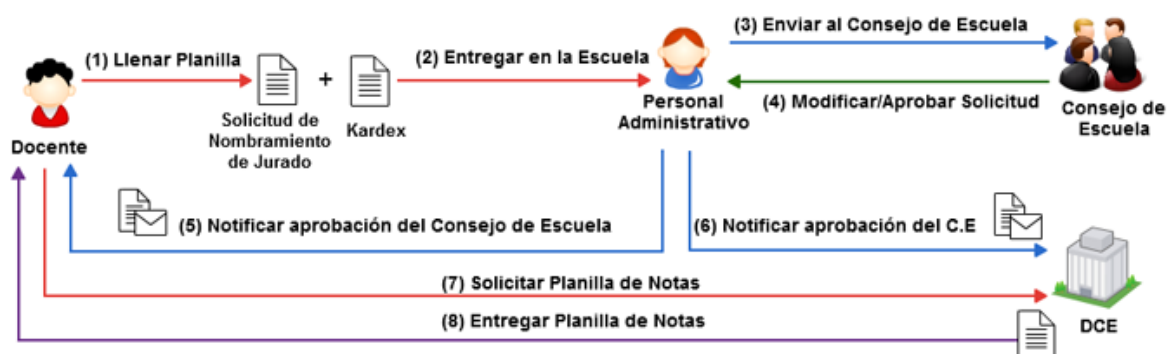


Figura 1.1: Proceso de solicitud de jurado.

Como se puede apreciar en la figura anterior, el flujo de información en todas las escuelas se realiza de forma manual, con la entrega frecuente de documentos en físico. En el caso del seminario, el proceso es más sencillo, puesto que no se requiere la aprobación del jurado en Consejo de Escuela. Por lo tanto, el tutor académico puede seleccionar el jurado, que en este caso está compuesto sólo por dos jurados principales, especificando aquellos que considere adecuados para la evaluación del trabajo.

Por otra parte, existen algunas observaciones a tomar en cuenta respecto a la solicitud de jurado, tales como:

- El jurado del seminario o TEG puede estar conformado por docentes de la facultad o investigadores externos.
- El tutor firmante del seminario o TEG, quien coloca su firma en la planilla de notas, debe pertenecer a la facultad, así como a la escuela correspondiente a la carrera del estudiante, aunque pueden existir excepciones para esta última condición.
- La Escuela de Computación, a diferencia de las demás, permite realizar el seminario o TEG en parejas de estudiantes. Por ende, una solicitud de jurado de la Escuela de Computación puede realizarse en nombre de uno o dos estudiantes.

Capítulo 1. Solicitudes estudiantiles

- Al recibir la aprobación de las solicitudes de jurado, la DCE realiza un chequeo de requisitos de los estudiantes, adicional al realizado por CONEST al momento de la inscripción, para garantizar que cumplen con las condiciones necesarias para presentar el Trabajo Especial de Grado. Este chequeo se basa en verificar que el estudiante cumple con el *pensum* establecido para su carrera, es decir, que cumple con la cantidad de créditos y asignaturas aprobadas para poder presentar el TEG. De no cumplir con los requisitos, las partes involucradas son informadas y el proceso no puede continuar hasta que estos requisitos sean solventados.

Antes de la presentación del trabajo de investigación por parte del estudiante, el docente debe solicitar la planilla de notas, la cual contiene, entre otros datos, la calificación obtenida por el estudiante después de la evaluación realizada por parte del jurado que fue aprobado en el Consejo de Escuela, a través de la solicitud de jurado. Es de vital importancia que la información que contiene la planilla de notas sobre el jurado sea consistente con la información aprobada en la solicitud, para garantizar que el proceso de calificación sea transparente y legal.

Después de la presentación y defensa del trabajo correspondiente en la fecha pautada para ello, el tutor académico debe entregar la planilla de notas del estudiante firmada por los miembros del jurado en la División de Control de Estudios.

Cabe destacar que el personal de la DCE debe introducir manualmente la información de esta planilla para generarla a través del sistema, basándose en la carta de aprobación que debe estar en el expediente del estudiante. Sin embargo, existen casos donde esta carta no se envía de manera oportuna, por lo que se producen retrasos en el trámite, sin mencionar el tiempo y esfuerzo invertido en buscar el expediente de cada estudiante para obtener y verificar la información cuando han pasado varios semestres después de aprobada la solicitud.

Aunque es poco probable debido a la seriedad y profesionalismo del personal, existe el riesgo de que por errores humanos, el jurado especificado no sea el aprobado por el Consejo de Escuela. Para el caso de los seminarios, puede ser más probable que esto ocurra por olvido o confusión, en vista de que un tutor puede tener varios tesis a la vez, y al momento de solicitar la planilla se deben especificar verbalmente los miembros del jurado. La cantidad de tesis que puede tener un tutor también dificulta a los mismos la gestión de solicitudes de jurado y planillas de notas, ya que no se cuenta con un mecanismo que les permita consultar esta información.

Otra desventaja del proceso actual es que el personal administrativo de las escuelas debe gestionar de forma manual la recepción, control y procesamiento de todas las solicitudes de jurado que se realizan en cada semestre, así como el flujo de documentos (solicitudes, cartas, memorandos) entre los actores involucrados en este proceso. Además, los errores o estancamientos que pudieran surgir en el proceso pueden requerir cierto período de tiempo para ser resueltos, y su resolución depende principalmente del ente donde se originó el conflicto.

Sin embargo, debido a la importancia de este trámite, que por lo general se realiza en las últimas semanas del semestre lectivo, y que puede afectar la posibilidad de graduación de los estudiantes, se desea poder brindar soluciones con el menor esfuerzo y tiempo posible.

De la situación descrita anteriormente se deriva la necesidad de automatizar el proceso de solicitud de jurado, con el objeto de facilitar su gestión y control por parte de cada rol involucrado, agilizar el proceso de grado, así como para mantener la integridad y consistencia de la información aprobada entre las solicitudes y las planillas de notas para seminario o TEG.

1.2. Solicitudes de modificación de inscripción

Se trata de un conjunto de peticiones estudiantiles que permiten solicitar la inscripción de asignaturas adicionales, superando las restricciones establecidas en el período de inscripción del semestre, tales como las relaciones de prelación entre ellas y el límite de créditos a inscribir. Son realizadas durante las dos primeras semanas del semestre en curso, y están sujetas a un conjunto de normas establecidas en el reglamento interno de la universidad, y de cada escuela de la Facultad de Ciencias.

A continuación se describen brevemente las solicitudes que componen el proceso de modificación de inscripción.

- **Exceso de créditos:** Cada carrera tiene un límite de unidades de créditos para la inscripción de asignaturas por semestre. A través de esta solicitud, el estudiante tiene la oportunidad de inscribir una cierta cantidad de asignaturas que causen un número de créditos mayor al límite permitido en el semestre en curso.
- **Inscripción de paralelos:** Solicitud en la que el estudiante requiere cursar simultáneamente en un semestre dos asignaturas, una de las cuales tiene una relación de prelación con la otra, y que por lo general ha sido previamente reprobada.
- **Inscripción extemporánea:** Se da cuando el estudiante solicita, fuera del período determinado para realizar la inscripción del semestre, la inscripción de una o más asignaturas, siendo el requisito principal el aval firmado del profesor a cargo de las mismas.

Anteriormente, en la primera automatización de este proceso realizada por Mendez y Boyer en el año 2009, y hasta el año 2011, se permitía la anulación de asignaturas inscritas a través del sistema, lo cual consistía en la cancelación de la inscripción de un conjunto de asignaturas seleccionadas por el estudiante, sin que esto se viera reflejado en su expediente académico. A partir del año 2012, este tipo de solicitud fue desactivada del sistema, y en la actualidad son manejadas a través de peticiones directamente al Consejo de Escuela.

Capítulo 1. Solicitudes estudiantiles

Por otra parte, la normativa de solicitudes estudiantiles de las escuelas de Biología, Computación y Química, vigentes a la fecha, se incluyen en los anexos A, B y C, ubicados en las páginas 112, 113 y 114 respectivamente, como documentación para su consulta. Es importante destacar que, en un principio, sólo la Escuela de Computación contaba con la normativa que rige a estas solicitudes. Para el momento en que se realizó la primera automatización de este proceso, no se contaba con la normativa de las demás escuelas, por lo que sus particularidades no fueron modeladas ni incluidas en ese entonces.

Posteriormente, el resto de las escuelas elaboró sus propias normativas, y esta es una de las razones fundamentales por las que se requiere evolucionar el sistema, a fin de modelar e incorporar las diferentes reglas que rigen a cada escuela de la facultad. Esto permitiría proporcionar una validación y retroalimentación más acertada de las solicitudes estudiantiles.

Como se mencionó anteriormente, ya existe un antecedente respecto a la automatización de estas solicitudes en la primera versión de CONEST, el cual fue estudiado y tomado en cuenta para el desarrollo. Es por ello que este proceso se describe de forma general, y a manera de precedente, a continuación.

El proceso inicia cuando el estudiante, dentro del período correspondiente, realiza sus solicitudes de modificación de inscripción, pudiendo agregar una exposición de motivos con el objeto de explicar o justificar las razones de sus peticiones, a través de CONEST. En este punto, es importante destacar que el sistema es capaz de:

- Mostrar una selección de aquellas asignaturas que el estudiante puede inscribir de acuerdo a su historial académico.
- Determinar por sí mismo qué tipo de solicitud se está realizando al seleccionar cada materia, sin necesidad de que el estudiante lo especifique.
- Realizar validaciones, tales como la cantidad de solicitudes de paralelo permitidas, a fin de cumplir con el reglamento.

Sin embargo, esta versión del sistema no proporciona retroalimentación al estudiante sobre si cumple o no los requisitos establecidos en el reglamento de la escuela a la que pertenece, por lo cual se tiene cierta desinformación. La mayoría de los estudiantes no conocen la normativa, y por lo general desconocen las razones por las cuales sus solicitudes pudieran ser rechazadas.

La figura 1.2 muestra una captura de la interfaz disponible para el estudiante en este sistema, con algunas solicitudes de ejemplo realizadas.

Capítulo 1. Solicitudes estudiantiles

Servicios al Estudiante > Solicitud de Modificación de Inscripción - Paso 2 de 3

Estudiante: DE SOUSA JESUS
Cédula: V-20219212
Licenciatura: COMPUTACION

Solicitud de Modificación de Inscripción - Paso 2 de 3

Paso 3: Resultados de su solicitud

A continuación ud. podrá visualizar sus solicitudes de modificación de inscripción.

CODIGO	NOMBRE	CREDITOS	SOLICITUD	ACCION	OBSERVACION
6002	ORGANIZACION Y ESTRUCTURA DEL COMPUTADOR II	5	MANTENER INSCRITA	NINGUNA	
6203	INGENIERIA DE SOFTWARE	5	MANTENER INSCRITA	NINGUNA	
8207	MATEMATICA II	6	MANTENER INSCRITA	NINGUNA	
6303	BASES DE DATOS	5	INSCRIBIR PARALELO CON INGENIERIA DE SOFTWARE	INSCRIBIR	
0206	ACTIVIDAD CORAL	3	INSCRIBIR	INSCRIBIR	REQUIRE AVAL DEL PROFESOR

TOTAL DE CREDITOS A INSCRIBIR: 24

Justificación (opcional)

A continuación indique una exposición de motivos para realizar esta solicitud, en caso de ser necesaria.

Figura 1.2: Solicitudes estudiantiles realizadas en la primera versión de CONEST.

Una vez confirmadas las solicitudes, el estudiante debe imprimir un documento *PDF*, el cual debe ser firmado por el docente en señal de estar de acuerdo con la solicitud realizada por el estudiante. Este aval es un requisito indispensable para consignar el documento ante el personal administrativo de la escuela y proceder con la solicitud formal.

A medida que se reciben las solicitudes, el personal administrativo registra en el sistema la entrega del aval por parte de los estudiantes, enviando las solicitudes a la unidad de asesoramiento académico (UAA) correspondiente. Este ente se encarga de estudiar las solicitudes y sugerir su aprobación o rechazo, tomando en cuenta el rendimiento del estudiante, su situación actual, así como considerando las sugerencias que el propio sistema CONEST ofrece en base al reglamento interno. Esta última funcionalidad, destinada a apoyar la toma de decisiones, puede apreciarse en la figura 1.3.

Solicitud Estudiantil

Código	Nombre de Materia	Tipo de Solicitud	Sugerencia	Razones	Sugerir
0206	ACTIVIDAD CORAL	INSCRIBIR	Se sugiere su aprobación	Cumple con los requisitos	<input type="radio"/> APROBACION <input type="radio"/> RECHAZO
6303	BASES DE DATOS	PARALELO con INGENIERIA DE SOFTWARE	Se sugiere su rechazo	No se concederán paralelos de materias a estudiantes que hayan aplazado más de una materia en el semestre inmediato anterior	<input type="radio"/> APROBACION <input type="radio"/> RECHAZO

Justificación (Opcional)

Figura 1.3: Sugerencias para solicitudes estudiantiles en la primera versión de CONEST.

Después de que la UAA sugiere las decisiones pertinentes, se genera un reporte que contiene un lote determinado de solicitudes estudiadas, el cual es enviado al Consejo de Escuela para su discusión en una fecha determinada. Este organismo discute las sugerencias y toma las decisiones finales, pudiendo contradecir la opinión de la UAA respecto a la aprobación o rechazo de las solicitudes, o basarse en su criterio para tomar las decisiones. La resolución

final es registrada por el personal administrativo en el sistema CONEST para su posterior ejecución. Además, este personal debe generar e imprimir en físico el lote de solicitudes con la fecha registrada de la resolución del Consejo de Escuela, el cual es enviado a la DCE, ente que verifica que el número de solicitudes procesadas sea igual al físico recibido. Esto debido a que siempre se debe contar con el documento en físico que corrobore el trámite a realizar, ya que el mismo debe ser almacenado en el expediente de cada estudiante.

Finalmente, la DCE activa un botón de procesamiento de solicitudes por lotes, de forma que esta acción ejecuta y actualiza el estado de estas solicitudes en el sistema CONEST, y envía un correo con la respuesta final al estudiante, informándole la aprobación o rechazo de sus solicitudes. La figura 1.4 ilustra el proceso descrito anteriormente.

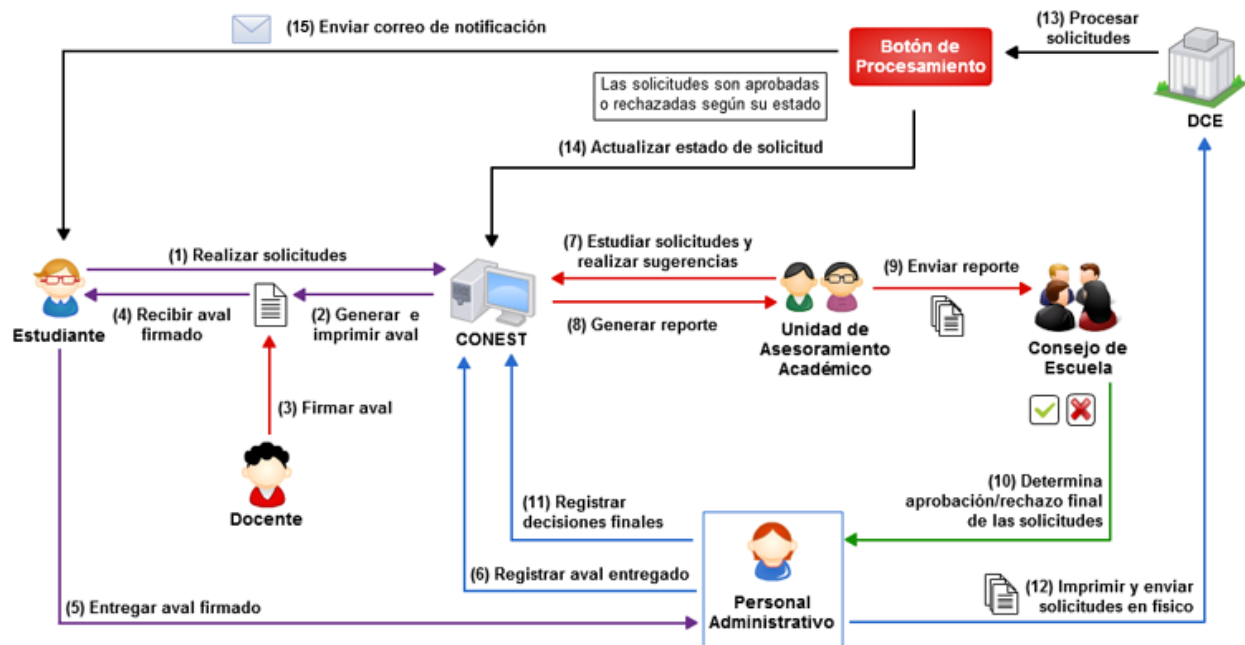


Figura 1.4: Proceso de modificación de inscripción.

Existen algunas observaciones relevantes respecto a este proceso:

- En algunas escuelas como la de Matemática, las solicitudes no requieren el estudio de la UAA, puesto que el Consejo de Escuela realiza esta tarea en su lugar.
- Debido a la complejidad de la normativa de la Escuela de Biología, y debido a la falta de automatización de la misma, el personal administrativo es quien realiza las solicitudes a petición de los estudiantes, lo que implica una mayor carga de trabajo.
- Las escuelas también reciben y gestionan solicitudes estudiantiles de forma tradicional a través de cartas al Consejo de Escuela, como se refirió en las solicitudes de anulación de asignaturas. Una solicitud relevante es la inscripción de asignaturas PCI (Programa de

Cooperación Interfacultades), que no está automatizada por el sistema CONEST. Por ello deben ser procesadas manualmente y además suelen ser numerosas. En caso de fallas del sistema, éste también es el mecanismo alternativo para atender solicitudes.

- Como se puede apreciar en la figura anterior, este proceso es complejo y requiere la participación de varios actores a lo largo del flujo. Cada actor cumple un conjunto de funciones específicas y determinantes en el proceso. Si se presenta algún problema o estancamiento en algún paso del flujo, el mismo se ve afectado por completo. Estas situaciones pueden darse por errores humanos o fallas en el sistema, produciendo un impacto en el tiempo y esfuerzo invertido para resolver la situación.
- Por la razón anterior, se maneja una gran cantidad de estados a la hora de realizar seguimiento de las solicitudes. Se tienen aproximadamente veinte (20) estados registrados para esta tarea, lo cual denota la complejidad del proceso.
- En promedio, se generan cuatrocientas (400) solicitudes de modificación de inscripción por semestre entre todas las escuelas a través de CONEST, el cual es un número elevado. La Tabla 1.1 presenta la cantidad de solicitudes realizadas por los estudiantes de las diferentes carreras de la facultad mediante este sistema, entre los años 2010 y 2013.

Tabla 1.1: Cantidad de solicitudes estudiantiles realizadas a través de CONEST entre los años 2010 y 2013.

Carrera	01-2010	02-2010	01-2011	02-2011	01-2012	02-2012	01-2013	02-2013
Biología	82	53	52	39	25	76	46	65
Computación	168	194	230	209	160	116	110	173
Física	29	28	46	42	26	31	54	19
Geoquímica	21	56	49	43	45	26	20	40
Matemática	6	49	45	44	22	16	10	18
Química	47	97	84	114	81	44	88	77
Total	353	477	506	491	359	309	328	392

- Hasta el año 2011 se experimentó un alto volumen de solicitudes debido a que se permitía la anulación de asignaturas a través de CONEST. A partir del año 2012, como resultado de la desincorporación de esta funcionalidad, se aprecia una disminución importante en este número. La figura 1.5 evidencia este comportamiento, permitiendo resaltar que el mayor volumen de solicitudes generadas históricamente corresponde a las escuelas de Computación y Química.

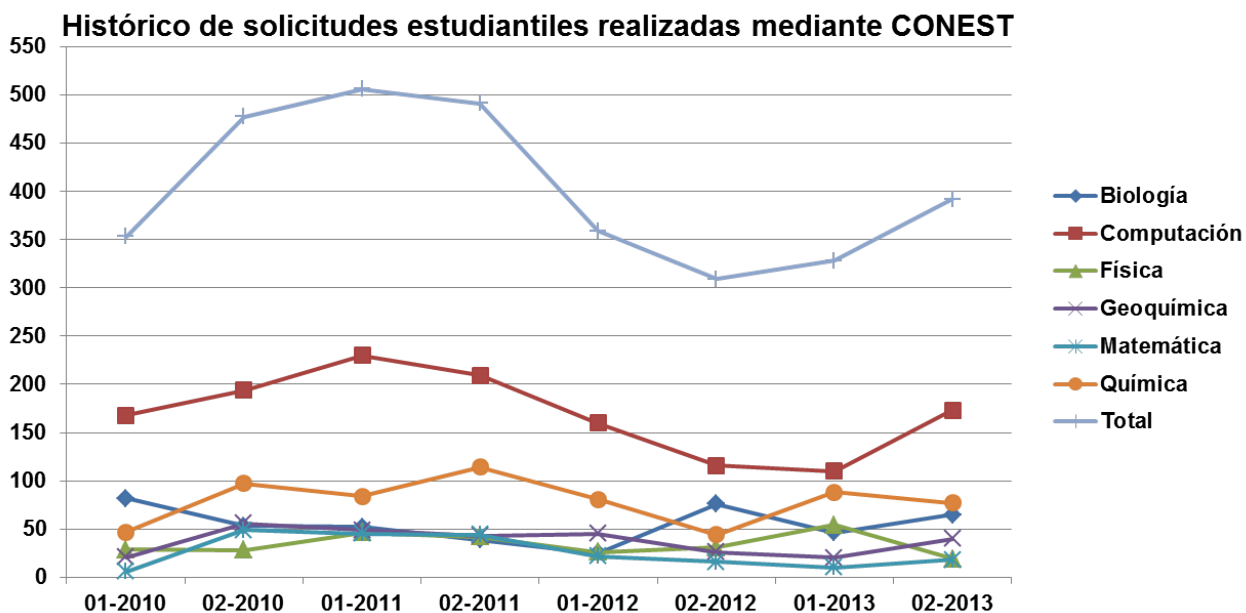


Figura 1.5: Tendencia de las solicitudes estudiantiles realizadas a través de CONEST entre los años 2010 y 2013.

- Como consecuencia del alto número de solicitudes que se generan por semestre, aun cuando se tiene cierto grado de automatización, este proceso implica altos costos en materia de recursos humanos para la recepción, análisis y procesamiento de las solicitudes, así como una cantidad de tiempo considerable para dar resolución a las mismas.
- Debido a la periodicidad de los Consejos de Escuela (cada dos semanas en algunas escuelas), y considerando los retrasos que pudieran surgir, puede tomar más de un mes procesar las solicitudes estudiantiles. Sin embargo, los estudiantes requieren obtener respuestas rápidas a sus peticiones para poder cursar desde el inicio, y con seguridad, las asignaturas solicitadas. Este período, junto con el proceso de inscripción, por lo general conlleva una alta carga de estrés, tanto para los estudiantes, como para el personal administrativo de la facultad.
- Se tiene un gasto muy alto de materiales como papel y tinta, debido a la impresión de reportes y de los lotes de solicitudes que deben ser enviados a la DCE para su procesamiento final. Estos gastos corren por cuenta de cada escuela, y es necesario recordar que se cuenta con un presupuesto limitado. Además, la DCE suele imprimir una lista con el estado final de todas las solicitudes procesadas, la cual se coloca en una cartelera informativa para la consulta de los estudiantes.
- Existe una alta dependencia del sistema, por lo que todo el personal involucrado debe necesariamente tener conocimientos sobre el manejo de CONEST para la gestión de solicitudes. Esto implica que si hay cambios en el personal, se debe capacitar a los nuevos miembros sobre su uso, lo cual implica un proceso de aprendizaje y adaptación.

- Como se refirió anteriormente, el estudiante no obtiene información oportuna sobre el cumplimiento de las reglas que conforman la normativa de su escuela para realizar solicitudes. Este documento no está al alcance del estudiante, por lo que la gran mayoría desconoce el reglamento. De esta forma, el estudiante no puede anticipar el resultado de sus peticiones, y esto incrementa su incertidumbre. Si se proporcionara información al respecto, el proceso para realizar solicitudes por parte del estudiante sería más consciente y sensato, pudiendo llegar a persuadirlo de no realizar ciertas solicitudes si existen pocas probabilidades de que sean aprobadas, o al menos podría brindarle un panorama realista. Esto también pudiera contribuir a reducir el volumen general de solicitudes estudiantiles.
- El sistema no permite la consulta de solicitudes de semestres anteriores, lo cual dificulta obtener esta información, ya que las escuelas no cuentan con el respaldo de la información final sobre las solicitudes. En caso de errores o conflictos en la aprobación de solicitudes, es complicado determinar responsabilidades y solucionar estas situaciones.
- La normativa que rige las solicitudes no se cumple como debería, puesto que el sistema no se ajusta a ella. Por ejemplo, para el caso de los extracréditos, sólo se debería permitir una solicitud de exceso de créditos, pero la realidad es que los estudiantes pueden realizar más solicitudes porque el sistema lo permite, y actualmente no se puede restringir este comportamiento.
- Las reglas que constituyen las normativas de las escuelas son bastante complejas, difíciles de modelar y unificar. Además, son propensas a ser modificadas en el tiempo. El sistema actual no incorpora el reglamento de todas las escuelas, y no ofrece flexibilidad para adaptarse a los cambios que pudieran surgir en el futuro. Las reglas están implementadas de manera fija, requiriendo la modificación del código fuente en caso de ser necesaria su actualización.

El proceso de modificación de inscripción tiene una importancia comparable al proceso de inscripción, en especial para los estudiantes que necesitan realizar solicitudes para inscribir las asignaturas que por múltiples razones no pudieron seleccionar durante el período de inscripción, con la expectativa de obtener una respuesta oportuna que les permita cursar las asignaturas en el semestre que inicia, con el menor retraso posible.

Considerando el alto número de solicitudes que se generan cada semestre, junto al proceso complejo que ellas conllevan, se tiene la necesidad de proporcionar respuestas de la manera más eficiente, en el menor tiempo y con el menor esfuerzo posible. Esto implica mejorar y simplificar el proceso actual, minimizando las deficiencias derivadas del mismo, así como de su propia automatización.

1.3. Solicitudes de reincorporación

Los estudiantes de pregrado de la Universidad Central de Venezuela están sujetos al cumplimiento de las normas sobre rendimiento mínimo y condiciones de permanencia, establecidas por el Consejo Universitario en el año 1982. Este reglamento establece los límites mínimos en cuanto al rendimiento académico que deben mantener los estudiantes, como condición para permanecer con tal carácter en la universidad.

El artículo 3 de esta normativa indica que todo aquel estudiante que no apruebe el 25 % de la carga académica de un período determinado, o que en todo caso, no apruebe al menos una asignatura, deberá participar en un procedimiento especial de recuperación, estando bajo la asesoría de un profesor consejero, quien orientará al estudiante a lo largo del siguiente semestre para apoyar su recuperación.

El artículo 6 de la misma normativa está estrechamente relacionado con el anterior, ya que indica que el estudiante que no alcance a aprobar nuevamente el 25 % de la carga académica, o no apruebe por lo menos una asignatura después de pasar por el procedimiento anterior, no podrá reinscribirse en la universidad en los dos (2) semestres siguientes. Pasados éstos, tendrá el derecho de reincorporarse sin que puedan exigírsele otros requisitos aparte de los trámites administrativos usuales. De este artículo se deriva una gran parte de las solicitudes de reincorporación.

Los estudiantes también pueden desincorporarse de la universidad por motivos distintos a los establecidos en esta normativa, ya sea por deserción de sus estudios, problemas personales, entre otros; de forma tal que existe una diferenciación respecto al motivo de la desincorporación que debe ser tomada en cuenta.

Cuando un estudiante desea reincorporarse a la facultad, ya sea por haber cumplido con lo establecido en el artículo 6 de las normas de permanencia, o para reanudar sus estudios después de cierto tiempo, debe solicitar y posteriormente entregar en la División de Control de Estudios, en el período establecido para ello, una planilla de solicitud de reincorporación que contiene sus datos personales, junto con su expediente curricular actualizado, una carta de solicitud dirigida al Consejo de Escuela, y una constancia de entrevista en el Servicio de Orientación. Este último organismo realiza una evaluación vocacional y psicológica al estudiante, con el fin de determinar su aptitud para reingresar a la facultad. Este informe es enviado a la unidad de asesoramiento académico de la escuela respectiva, quien estudia el caso y emite las recomendaciones pertinentes al Consejo de Escuela. Finalmente, el Consejo de Facultad emite la resolución final para que la DCE pueda ejecutar la aprobación o rechazo de las solicitudes de reincorporación. En la figura 1.6 se ilustra este proceso.

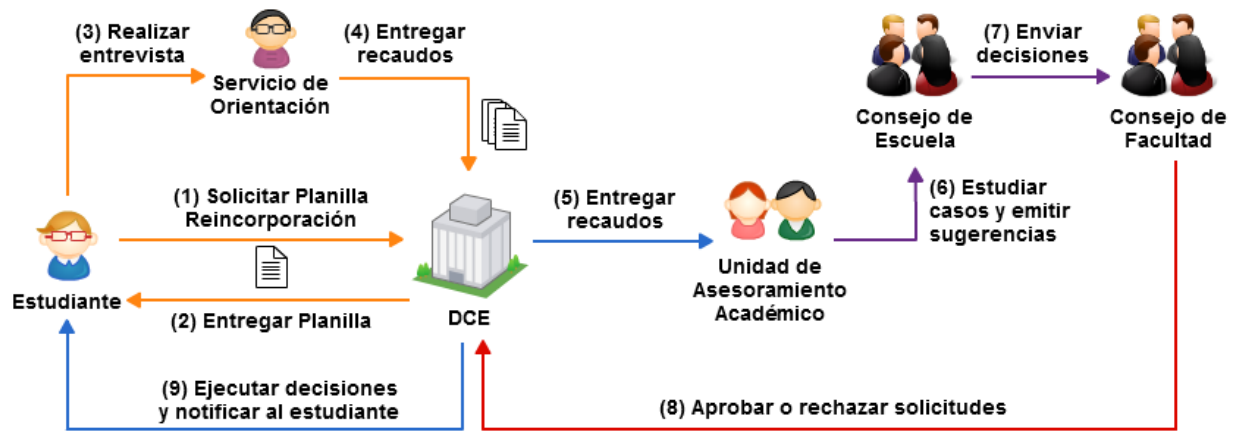


Figura 1.6: Proceso de reincorporación.

La Facultad de Ciencias posee una normativa interna que rige a estas solicitudes, la cual fue aprobada por el Consejo de Facultad en el año 2006. La normativa especifica un conjunto de condiciones que los estudiantes deben cumplir para su reincorporación, las cuales pueden ser consultadas en el anexo D, ubicado en la página 115.

Como se puede observar, este proceso requiere la acción de un conjunto numeroso y variado de actores, e involucra a una serie de organismos heterogéneos que deben comunicarse e intercambiar información para dar respuesta oportuna a estas solicitudes, que pueden llegar a ostentar un número muy elevado, y que necesitan un estudio multidisciplinario. Además, se requiere contar con un mecanismo que permita discriminar la causa de la reincorporación, bien sea por cumplimiento de la sanción especificada en el artículo 6 del reglamento, o por desincorporación voluntaria, con el fin de priorizar la gestión de las solicitudes y ofrecer respuesta oportuna a aquellas que se derivan del reglamento para garantizar la inscripción del estudiante en el período que le corresponde.

Como consecuencia de la puesta en producción de CONEST 3.0, este proceso se lleva a cabo de forma manual, a pesar de que la versión anterior de CONEST lo automatizaba de cierta manera. Por lo tanto, la necesidad de este proceso se basa en su automatización de forma general, y en brindar la información más relevante sobre la normativa y el rendimiento académico del estudiante para apoyar la toma de decisiones entre los entes respectivos. Asimismo, se requieren mecanismos para la administración de las reglas que conforman la normativa, ya que la misma puede variar con el paso del tiempo.

Capítulo 2. Tecnologías y herramientas web

Una selección apropiada de las tecnologías de desarrollo web es vital para la implementación ágil de una aplicación que satisfaga tanto los requerimientos de los usuarios, como los atributos de calidad que debe poseer el software, tales como usabilidad, eficiencia, seguridad, portabilidad, confiabilidad y robustez.

El propósito de este capítulo es presentar los conceptos y tecnologías web que sirven de fundamento para el desarrollo del presente Trabajo Especial de Grado. Este capítulo se divide en dos secciones principales, las cuales se describen a continuación.

La primera sección corresponde a las aplicaciones web, donde se describe su concepto y su relación con la arquitectura cliente/servidor, explicando además el funcionamiento de la arquitectura de Tres-Capas. También se describe el patrón de diseño MVC (Modelo-Vista-Controlador). Los conceptos tratados en esta sección son fundamentales para comprender el diseño y funcionamiento básico de las aplicaciones web.

En la segunda sección se resumen las tecnologías empleadas para el desarrollo de este trabajo, las cuales están subdivididas en tecnologías del lado del cliente y del lado del servidor.

Como tecnologías del lado del cliente se describen las siguientes: el lenguaje de marcado *HTML5*, las hojas de estilo *CSS3*, el lenguaje de *scripting JavaScript*, el formato de texto *JSON*, además del compendio de tecnologías *AJAX*, la librería *jQuery*, y el *framework* para interfaces web *jQuery UI*.

En las tecnologías del lado del servidor, se describe el lenguaje *Ruby* y sus características principales, así como el *framework* para aplicaciones web *Ruby on Rails*, y sus componentes básicos. También se hace una síntesis del Sistema Manejador de Base de Datos Relacional, *MySQL*, y de la herramienta de integración de código *Git*. Como cierre del capítulo, se presenta una ilustración a manera de resumen sobre las tecnologías mencionadas anteriormente.

2.1. Aplicaciones web

En general, una aplicación web se refiere a “cualquier programa que es accedido a través de una conexión de red usando el Protocolo de Transferencia de HiperTexto (*HyperText Transfer Protocol, HTTP*), comúnmente a través de un navegador web.” (Janssen, nd)

Pressman (2002) señala los atributos que caracterizan a las aplicaciones web. Algunos de ellos se listan a continuación.

- *Intensivas de Red*: Estas aplicaciones residen en una red y deben dar servicio a las necesidades de una comunidad diversa de clientes. Una aplicación web puede residir en Internet, haciendo posible así una comunicación abierta para todos los usuarios. De forma alternativa, una aplicación se puede ubicar en una Intranet, implementando la comunicación a través de redes de una organización, o en una Extranet, empleando una comunicación entre redes.
- *Evolución continua*: A diferencia de las aplicaciones convencionales, que evolucionan mediante una serie de versiones planificadas y cronológicamente espaciadas, las aplicaciones web están en constante evolución.
- *Inmediatez*: Se refiere al corto periodo de tiempo empleado en el desarrollo de aplicaciones web, comparado con las aplicaciones convencionales.
- *Seguridad*: Dado que las aplicaciones web están disponibles a través del acceso por red, es difícil limitar la población de usuarios finales que pueden acceder a la aplicación. Por ello, se debe implementar medidas de seguridad con el objeto de proteger el contenido confidencial y de proporcionar formas seguras de transmisión de datos.
- *Estética*: Las aplicaciones web destacan principalmente por su apariencia e interactividad, por lo que las mismas deben resultar atractivas y usables.

Entre las ventajas de las aplicaciones web se tiene la disponibilidad continua, independientemente de la ubicación geográfica, la practicidad de los navegadores web, la independencia del sistema operativo subyacente, la variedad de dispositivos a través de los cuales pueden ser accedidas, y la facilidad de mantener y actualizar aplicaciones web sin la necesidad de distribuir e instalar el software a miles de usuarios potenciales.

2.1.1. Arquitectura cliente/servidor

Las aplicaciones web son un tipo especial de arquitectura cliente/servidor. Esta arquitectura representa un modelo de aplicación distribuida, donde las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los entes solicitantes, llamados clientes.

Un cliente realiza peticiones a otro programa, el servidor, quien atiende estas solicitudes y les da respuesta. El cliente está representado generalmente por el navegador web, que realiza las peticiones en nombre del usuario a un servidor web, a través del protocolo *HTTP*.

La arquitectura cliente/servidor puede ser categorizada dependiendo del número de capas que la compone. Una de las variantes más utilizada en la actualidad es la de Tres-Capas. Grove (2010) describe esta arquitectura de la siguiente forma: La primera capa o nivel es la del cliente, la cual se encarga de la presentación de la interfaz de usuario. En la segunda capa reside el servidor web, el cual se comunica con los clientes y maneja todo el procesamiento de la aplicación, es decir, contiene la lógica de negocio. La tercera capa consiste en un servidor de base de datos que proporciona la data necesaria para la aplicación. Este servidor suele presentar una interfaz de base de datos estándar, tal como el Lenguaje de Consulta Estructurado (*Structured Query Language, SQL*), la cual se accede a través de un manejador de base de datos apropiado para el lenguaje de programación web en uso. La figura 2.1 representa gráficamente esta arquitectura.



Figura 2.1: Arquitectura Tres-Capas.

Las ventajas de emplear una arquitectura de Tres-Capas, según Grove (2010) son:

- Mayor rendimiento, puesto que cada capa o plataforma puede ser configurada dependiendo de sus funciones.
- Bajo acoplamiento, ya que la relación entre la aplicación y el manejo de los datos está estandarizado, y cada componente puede ser reemplazado con facilidad.
- Escalabilidad mejorada, porque cada capa puede ser replicada cuando sea necesario.

Además de la arquitectura de Tres-Capas explicada anteriormente, es posible organizar los componentes de software de la aplicación web utilizando el patrón de diseño Modelo-Vista-Controlador (MVC), el cual se explica a continuación.

2.1.2. Patrón de diseño MVC (Modelo-Vista-Controlador)

El patrón Modelo-Vista-Controlador (*MVC*) es una arquitectura de diseño de software que permite separar los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Según la Universidad de Alicante (2012), se trata de un modelo muy utilizado y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo. A continuación, se describe brevemente cada componente planteado por este patrón.

- **Modelo:** Representa y encapsula los datos, el estado y las reglas de negocio, es decir, el dominio de la aplicación. Este componente se encarga de gestionar las peticiones de acceso y manipulación de los datos, las cuales provienen del *Controlador*. En algunas variantes, el *Modelo* es capaz de actualizar la *Vista*, mientras que en otras variantes esta comunicación no existe.
- **Vista:** Presenta los datos del *Modelo* en un formato adecuado para la interacción a través de la interfaz de usuario. La *Vista* suele emplear plantillas que permitan unificar el *look and feel* o apariencia del conjunto de páginas de una aplicación web.
- **Controlador:** Responde a eventos, que por lo general son originados por las acciones del usuario, y realiza invocaciones al *Modelo* para solicitar o manipular los datos, los cuales serán presentados a través de la *Vista*. También puede enviar órdenes a la *Vista* asociada si se solicita un cambio en la forma en que se presenta el *Modelo*. De esta forma, el *Controlador* es el componente intermediario entre la *Vista* y el *Modelo*.

La figura 2.2 muestra cómo se relacionan estos componentes entre sí.

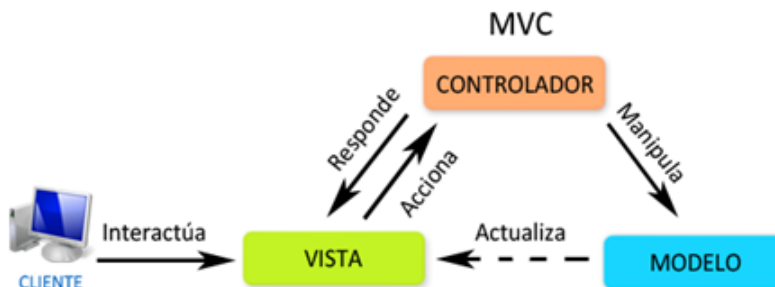


Figura 2.2: Patrón Modelo-Vista-Controlador.

Aunque originalmente el patrón MVC fue desarrollado para aplicaciones de escritorio, ha sido ampliamente adaptado como arquitectura para diseñar e implementar aplicaciones web en los principales lenguajes de programación. Se han desarrollado multitud de *frameworks* MVC comerciales y no comerciales, que implementan este patrón, tales como *Spring* (Java), *Ruby on Rails* (Ruby), *Symfony* (PHP), y *Django* (Python), entre muchos otros.

Algunas de las ventajas de usar este patrón de diseño son:

- Permite el desarrollo independiente y en paralelo de cada componente, favoreciendo la reutilización de código, la ejecución de pruebas de los componentes por separado, y la escalabilidad de la aplicación.
- Facilidad de representar la información del *Modelo* de distintas formas, modificando la interfaz de usuario (*Vistas*), sin afectar la lógica de negocio de la aplicación.
- El sistema de almacenamiento de datos, al estar encapsulado por el *Modelo*, puede ser reemplazado con facilidad.
- Simplificación del mantenimiento del sistema debido a la modularidad y la independencia entre el *Modelo*, la *Vista* y el *Controlador*.

2.2. Tecnologías empleadas para el desarrollo

El funcionamiento básico de una aplicación web consiste en generar dinámicamente páginas descritas en un formato estándar, tal como el *Lenguaje de Marcado de Hipertexto* (*HyperText Markup Language*, HTML). El uso de lenguajes de *scripting* como *JavaScript*, permite agregar interactividad a estas páginas, mientras que el uso de *Hojas de Estilo en Cascada* (*Cascading Style Sheets*, CSS) permite describir cómo se presentará visualmente la información en la página.

El servidor web es el encargado de atender y procesar las peticiones *HTTP* de los clientes, enviadas a través del navegador. En el servidor se ejecutan tecnologías que hacen posible recibir y dar respuesta a estas peticiones, tales como los *frameworks* y lenguajes de programación que permiten procesar y generar la información dinámica. Además, se necesitan tecnologías que permitan la persistencia de los datos de la aplicación, tales como los servidores de bases de datos, los cuales son imprescindibles para el almacenamiento y recuperación de la información. En la figura 2.3 se refleja la estructura básica de una aplicación web.

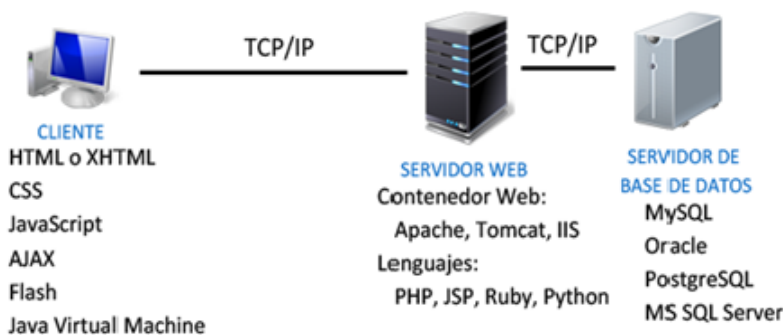


Figura 2.3: Estructura básica de una aplicación web.

2.2.1. Tecnologías del lado del cliente

Las tecnologías del lado del cliente comprenden aquellos lenguajes y programas que se ejecutan en el computador del usuario, específicamente en el navegador web, por lo que son independientes del servidor.

Uno de los principales problemas de la ejecución de aplicaciones distribuidas a través de Internet se refiere a la velocidad de transmisión, así como a la cantidad de información que circula por la red. Esta velocidad puede traducirse en lentitud de respuesta a las acciones realizadas por el cliente a través de la aplicación. (Da Silva, nd) Las tecnologías del lado del cliente surgen para reducir los problemas de saturación y lentitud en la transmisión y procesamiento de la información, de forma tal que, al maximizar el número de operaciones que ocurren en el lado del cliente, sin interactuar con el servidor, estas tomarán menos tiempo, se usará menos ancho de banda, y aumentará la seguridad de la aplicación.

La rápida respuesta ofrecida por las tecnologías del lado del cliente permiten brindar una sensación realista dentro del navegador, delegando al servidor únicamente los aspectos relacionados con el envío y procesamiento de información. En las siguientes secciones, se describen las tecnologías del lado del cliente más relevantes para el desarrollo del proyecto.

2.2.1.1. HTML5

HTML5 (HyperText Markup Language) es una colección de estándares para el diseño y desarrollo de páginas web. Esta colección define la manera en que se presenta la información en el navegador web y la forma de interactuar con ella. (TrazosWeb, 2010)

Según Walker (2011), la quinta revisión de *HTML* ofrece herramientas para la administración efectiva de datos, dibujo, video y audio. Facilita el desarrollo de aplicaciones para diferentes navegadores web, así como para dispositivos portátiles. Entre las mejoras que introduce *HTML5* se encuentran las siguientes:

- Simplificación: Se dispone de nuevas etiquetas que describen claramente su contenido.
- Contenido multimedia: Capacidad de reproducir audio, video y animaciones sin necesidad de instalar *plugins* de terceros.
- Soporte para hojas de estilo *CSS3*: Muchas de las cosas que sólo podían lograrse a través de imágenes, podrán realizarse con código. Esto no sólo se traduce en una mejora en la velocidad y rendimiento de una aplicación web, sino también en nuevas e ilimitadas opciones de diseño.

- Formularios mejorados que reducen la necesidad de emplear código *JavaScript* para realizar validaciones, permitiendo una comunicación más eficiente entre dispositivos móviles y servidores en la nube.
- Mejoras en la velocidad de almacenamiento y carga de páginas.

En cuanto al soporte de navegadores, se tiene que *HTML5* todavía no es un estándar oficial, por lo que los navegadores web aún no ofrecen un soporte completo. Sin embargo, todos los principales navegadores (*Safari*, *Chrome*, *Firefox*, *Opera*, *Internet Explorer*) continúan agregando nuevas características de *HTML5* a sus últimas versiones.

2.2.1.2. CSS3

Las *Hojas de Estilo en Cascada* (*Cascading Style Sheets*, *CSS* por sus siglas en inglés), según el W3C, organismo encargado de su estandarización, son "un mecanismo simple que describe cómo se mostrará un documento en la pantalla, cómo se va a imprimir, o incluso cómo será pronunciada la información presente en ese documento a través de un dispositivo de lectura". *CSS* se utiliza para dar estilo a documentos *HTML* y *XML* (*eXtensible Markup Language*), separando el contenido de la presentación, con el fin de facilitar el mantenimiento de las aplicaciones web.

La versión *CSS3* es la sucesora de la actual versión 2.1, e incluye mejoras tales como la posibilidad de crear bordes redondeados, textos con sombras, la capacidad de asignar múltiples fondos, un mejor manejo de tablas, inclusión de nuevas fuentes tipográficas, posicionamiento de elementos en pantalla, entre otras. (Lemus, 2007)

Los navegadores *Firefox*, *Chrome*, *Safari* y *Opera* son los más avanzados en el soporte de *CSS*, ya que incluyen muchos elementos de la versión *CSS3*, y un soporte casi perfecto de la actual versión 2.1. Por su parte, el navegador *Internet Explorer* sólo ofrece un soporte adecuado para *CSS* a partir de su versión 7. (Eguiluz, 2013b)

2.2.1.3. JavaScript

Es un lenguaje de *scripting* interpretado, dialecto del estándar *ECMAScript*, cuya última versión de su especificación es la 5.1 (Eguiluz, 2013c). Se define como un lenguaje orientado a objetos, basado en prototipos, imperativo, débilmente tipado (es decir, las variables no necesitan tener un tipo de dato especificado), y dinámico. Sintácticamente, *JavaScript* es similar a los lenguajes *C*, *C++* y *Java*. A pesar de su nombre, *JavaScript* no guarda ninguna relación directa con el lenguaje de programación *Java*.

Mientras *HTML* se utiliza para almacenar el contenido y formato de una página web, y *CSS* codifica el estilo de cómo se debe presentar visualmente la información, *JavaScript* se utiliza para añadir dinamismo, efectos e interactividad a las aplicaciones web. Este lenguaje se utiliza principalmente en el ámbito del lado del cliente, siendo interpretado y ejecutado por el motor *JavaScript* del navegador web, pero también existe una variante de *JavaScript* del lado del servidor. Su compatibilidad con la mayoría de los navegadores modernos, lo posiciona como el lenguaje de programación del lado del cliente más utilizado.

2.2.1.4. JSON

JSON (*JavaScript Object Notation*, traducido como *Notación de Objetos de JavaScript*) es un formato ligero de intercambio de datos. La sintaxis definida por este formato es simple de leer y escribir, mientras que para las máquinas su generación e interpretación es sencilla, en comparación con otros formatos tales como *XML*. La figura 2.4 (W3schools, ndb) presenta una comparación entre ambos formatos.

JSON es un formato de texto completamente independiente del lenguaje de programación y ampliamente soportado, puesto que utiliza convenciones de la familia de lenguajes *C*, incluyendo otros tales como *Java*, *JavaScript* y *Python*. Estas propiedades hacen que *JSON* sea un lenguaje ideal para el intercambio de datos. (JSON.org, nd)

JSON	XML
<pre>"empleados":[{"nombre": "José", "apellido": "Pérez"}, {"nombre": "María", "apellido": "López"}, {"nombre": "Pedro", "apellido": "Durán"}]</pre>	<pre><empleados> <empleado> <nombre>José</nombre><apellido>Pérez</apellido> </empleado> <empleado> <nombre>María</nombre><apellido>López</apellido> </empleado> <empleado> <nombre>Pedro</nombre><apellido>Durán</apellido> </empleado> </empleados></pre>

Figura 2.4: Comparación de formato *JSON* vs *XML*.

2.2.1.5. AJAX

AJAX, traducido como *JavaScript Asíncrono + XML*, es una técnica que permite crear páginas web dinámicas y rápidas, brindando la posibilidad de que las páginas se actualicen asincrónicamente al intercambiar pequeñas cantidades de datos con el servidor. Esto significa que es posible actualizar partes de una página web sin recargar la página completa. Las páginas web clásicas (que no utilizan *AJAX*) deben recargar la página si el contenido cambia. (W3schools, nda)

AJAX está basado en un conjunto de estándares de Internet. Según AdaptivePath (2005), *AJAX* no es una tecnología como tal, sino que emplea una combinación de tecnologías, tales como: *XHTML* (*eXtensible HyperText Markup Language*) y *CSS*, para crear una presentación basada en estándares, *DOM* (*Document Object Model*), para la interacción y manipulación dinámica de la presentación, *XML* o *JSON* para el intercambio y la manipulación de información, *XMLHttpRequest*, para el intercambio asíncrono de información, y *JavaScript* para unir todas las demás tecnologías. La figura 2.5 muestra las tecnologías que se combinan a la hora de utilizar *AJAX*. (Eguiluz, 2013a)

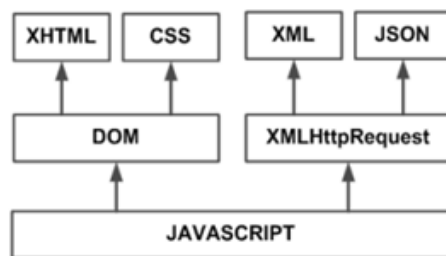


Figura 2.5: Tecnologías agrupadas bajo el concepto de *AJAX*.

2.2.1.6. jQuery y jQuery UI

jQuery es una librería *JavaScript* rápida, pequeña, compacta y con características enriquecidas, creada originalmente por John Resig y presentada en el año 2006. Permite abstraer la manipulación de documentos *HTML* a través del *DOM*, el manejo de eventos, el uso de *AJAX*, la utilización de animaciones, entre otras tareas, de una manera mucho más simple a través de un *API* (*Application Programming Interface*) que funciona en los navegadores web usados popularmente, por lo que simplifica la programación en *JavaScript*, siendo fácil de usar y aprender. (jQuery, 2013)

Por otra parte, *jQuery UI* es un conjunto de componentes en forma de *plugin* para la interacción con la interfaz de usuario de una aplicación web, y permite hacer uso de efectos, *widgets* y temas gráficos construidos sobre la librería *jQuery*. (jQueryUI, 2013)

Esta librería de código abierto es compatible con los navegadores web más usados, y facilita la implementación de interacciones y funcionalidades, favoreciendo la reutilización de código, así como disminuyendo el tiempo de desarrollo de la aplicación web al permitir la inserción de componentes de la interfaz de usuario (botones, menús, pestañas, ventanas de diálogo, calendarios, elementos de formularios, entre otros), y efectos (animaciones, *drag and drop*, autocompletado, entre otros). Además, es posible personalizar los componentes de *jQuery UI*, y aplicar temas para que sean acordes al estilo de la interfaz de usuario de la aplicación.

2.2.2. Tecnologías del lado del servidor

Las tecnologías del lado del servidor web permiten recibir y procesar una petición de usuario mediante la interpretación de un *script* en el servidor web para generar páginas *HTML* o *XHTML* dinámicamente como respuesta al cliente. Los lenguajes de programación del lado del servidor son aquellos que son reconocidos y ejecutados por el propio servidor antes de que se envíe la página web al cliente. Para estos lenguajes existen diversos *frameworks* que facilitan el desarrollo de las aplicaciones web.

En esta sección se describe el lenguaje *Ruby*, así como el *framework* de desarrollo web *Ruby on Rails*, además se ofrece una breve descripción sobre el Sistema Manejador de Base de Datos Relacional *MySQL*, y por último, se reseña la herramienta de control de versiones, *Git*.

2.2.2.1. Ruby

Es un lenguaje de programación interpretado, dinámico, reflexivo y orientado a objetos, creado por el programador japonés Yukihiko "Matz" Matsumoto y presentado públicamente en 1995. Su implementación oficial es distribuida bajo una licencia de software libre. Combina una sintaxis inspirada en *Python* y *Perl* con características de programación orientada a objetos (Rails.mx, nd). La versión 2.0 de este lenguaje fue liberada en febrero de 2013, y entre sus características generales se encuentran las siguientes (EcuRed, 2012):

- Soporta múltiples paradigmas de programación, incluyendo el paradigma funcional, orientado a objetos, e imperativo.
- Es orientado a objetos, no existen tipos de datos básicos o fundamentales. (Rivas et al., 2010)
- Es dinámicamente tipado, los tipos de datos son asignados en tiempo de ejecución y pueden cambiar de acuerdo con la asignación de cada variable. (Rivas et al., 2010)
- Tiene cuatro niveles de ámbito de variable: global, clase, instancia y local.
- Soporta alteración de objetos en tiempo de ejecución, permite el manejo de excepciones, y proporciona recolección de basura automática.
- Soporta el manejo de hilos (*threading*) independientemente del sistema operativo.
- Es altamente portable.

Ruby on Rails es uno de los *frameworks* basados en *Ruby* más populares, el cual se describe a continuación.

2.2.2.2. Ruby on Rails

También conocido como *RoR* o *Rails*, es un *framework* de aplicaciones web de código abierto, escrito en el lenguaje de programación *Ruby*. Fue creado por David Heinemeier Hansson, siendo liberada la primera versión en julio del 2004 (Rails.mx, nd).

Este *framework* enfatiza el uso de principios y buenas prácticas ampliamente conocidas, tales como el patrón *ActiveRecord*, el paradigma *Convención sobre configuración* (*Convention over configuration, CoC*), donde el desarrollador sólo debe definir aquella configuración que no es convencional, y el principio *No te repitas* (*Don't repeat yourself, DRY*) basado en mantener la información ubicada en un solo lugar, evitando su duplicación; siguiendo a la vez el paradigma de la arquitectura Modelo-Vista-Controlador.

Rails resalta entre la amplia variedad de *frameworks* web existentes, puesto que permite desarrollar aplicaciones complejas en un menor tiempo, escribiendo menos código y requiriendo un mínimo de configuración.

Componentes de Rails

Ruby on Rails está estructurado en varios paquetes o módulos, algunos de los cuales son descritos por Belfort y Ojeda (2009) de la siguiente forma:

- **ActiveRecord:** Un sistema de Mapeo-Objeto-Relacional (*ORM* por sus siglas en inglés) que brinda abstracción en el acceso a la base de datos. *ActiveRecord* proporciona al usuario de *Rails* ciertas funcionalidades como relaciones entre los objetos, agregación, herencia, métodos de búsqueda, validaciones, transacciones, entre otras.
- **ActionPack:** Es un componente de *Rails* que provee la separación necesaria para escribir de manera clara y delimitada el código destinado al control y a la presentación sin que esta afecte la alta interacción que existe entre las vistas y los controladores.
- **ActionView:** Es la capa responsable de crear toda o parte de la página que será desplegada en el navegador a través de plantillas.
- **ActionController:** Es la capa responsable de recibir las peticiones web y tomar decisiones con respecto a qué se debe ejecutar o si se debe redirigir la petición a otra acción.
- **ActionMailer:** Este componente permite enviar correos electrónicos desde la aplicación que se está utilizando usando vistas y modelos especiales de la clase *Mailer*.
- **ActiveResource:** Es la capa responsable de la implementación de servicios web *REST* (*Representational State Transfer*) del lado del cliente. A través de *ActiveResource* es

posible consumir servicios *REST* mediante el uso de objetos que actúan como un *proxy* para los servicios remotos.

Otro componente importante, y ampliamente utilizado en *Rails* son las *gemas*, las cuales son *plugins* o librerías distribuidas a través del gestor de paquetes *RubyGems*, las cuales pueden añadirse a los proyectos *Ruby on Rails*, para agregar o ampliar funcionalidades específicas, así como también para modificar el código original de *Rails*. Las siguientes son algunas de las gemas empleadas para el desarrollo del proyecto.

- **Haml**: El lenguaje de marcado *Haml* (*HTML Abstraction Markup Language*) permite describir el código *HTML* de un documento web de manera limpia y simple, usando la indentación y sin el uso de código embebido tradicional. *Haml* funciona como reemplazo de sistemas de plantillas de páginas embebidas como *PHP*, *RHTML* y *ASP* (Haml-Documentation, 2013). Está basado en cuatro principios: el marcado debería ser bonito, no debería repetirse, debería tener buen sangrado, y la estructura *XHTML* debería ser clara.
- **Scss**: También conocido como *Sassy CSS*, es un lenguaje de *scripting* interpretado y que genera código *CSS*. Consiste en dos sintaxis: la primera, llamada *sintaxis indentada*, que usa un estilo similar a *Haml*. En ella se usa la indentación para separar bloques de código y el caracter nueva línea para separar reglas. La sintaxis más reciente, *SCSS*, emplea el formato de bloques como *CSS*, utilizando llaves para denotar bloques de código, y punto y coma para separar las líneas dentro de un bloque. (Sass, nd)
- **CoffeeScript**: Es un lenguaje de programación que se compila a *JavaScript*, con el objeto de proporcionar una sintaxis más breve y legible, empleando la indentación para denotar bloques de código, reduciendo así el uso de símbolos como punto y coma, paréntesis y llaves. El código compilado como resultado tiende a ejecutarse igual o más rápido que el código equivalente escrito a mano (CoffeeScript, nd).
- **Kaminari**: Es un paginador limpio, poderoso, configurable y sofisticado para aplicaciones web modernas y *ORMs* (Matsuda, 2011). Al instalar esta gema, los modelos están listos para ser paginados, sin necesidad de realizar alguna configuración adicional. *Kaminari* soporta múltiples *ORMs* (*ActiveRecord*, *DataMapper*, *Mongoid*), múltiples *frameworks*, y múltiples motores de plantillas, tales como *ERB* (usado por defecto en *Rails*) y *Haml*. Además, hace uso de etiquetas *HTML5*, y soporta *AJAX* no obtrusivo.
- **Prawn**: Es una librería escrita en *Ruby*, dedicada a la generación de documentos en formato *PDF*, la cual proporciona muchas funcionalidades, manteniéndose simple y proporcionando un buen rendimiento a la vez (Brown, 2008). *Prawn* soporta, entre otras características, el dibujo de formas vectorizadas (líneas, polígonos, curvas, elipses, entre otros), incluye un soporte extenso para el manejo de textos y formatos, permite el

uso de imágenes *PNG* y *JPG* con opciones de escala, ofrece un buen manejo de tablas complejas, así como facilidades de paginación.

- **Rails 4:** La versión más reciente de *Rails* a la fecha es la número 4, la cual fue liberada en su versión final el 25 de junio de 2013. En esta versión, aunque no hay modificaciones a nivel de la estructura del *framework*, se requiere el lenguaje *Ruby* en su versión 1.9.3 o mayor, y prefiere la versión 2.0, por lo que ya no hay soporte para la versión 1.8 (RailsGuides, 2013). Además, se introduce el uso de *turbolinks*, que pueden transformar la aplicación *Rails* en una aplicación *JavaScript* de una sola página, es decir, no se cargan nuevas páginas sino que se reemplaza la página actual con el contenido del servidor en cada petición (Stocker, 2013).

2.2.2.3. MySQL

Es un Sistema Manejador de Bases de Datos Relacional (SMBDR), multihilo y multiusuario. Este servidor de base de datos de código abierto es muy popular, siendo muy rápido, confiable, escalable y fácil de usar. La siguiente lista describe algunas de sus características más importantes (MySQL, nda):

- Está escrito en *C* y en *C++*, y funciona en diferentes plataformas, tales como *Microsoft Windows*, *Linux*, *Mac OS X*, *Solaris* y *FreeBSD*.
- Existen *APIs* para el acceso a las bases de datos disponibles para los lenguajes *C*, *C++*, *Java*, *Perl*, *PHP*, *Python*, *Ruby*, entre otros.
- Uso de multihilos mediante hilos del *kernel*, lo que permite aprovechar la ventaja de usar procesamiento multihilos con varios procesadores.
- Soporta grandes cantidades de datos. *MySQL Server* tiene bases de datos de hasta 50 millones de registros.
- Proporciona un sistema seguro de contraseñas y privilegios mediante verificación basada en el *host*, y además, el tráfico de contraseñas está cifrado al conectarse al servidor de base de datos.

MySQL es usado por muchos sitios web grandes y populares, como *Wikipedia*, *Facebook*, *Twitter*, *Flickr*, *LinkedIn* y *YouTube*, entre otros. (MySQL, ndb). También, es comúnmente empleado en proyectos de código abierto tales como *Drupal* y *Joomla*.

2.2.2.4. Git

Es un software de control de versiones distribuido, libre y de código abierto, diseñado para manejar desde proyectos pequeños hasta proyectos muy grandes, con rapidez y eficiencia. Los sistemas de control de versiones permiten registrar los cambios realizados al código de la aplicación a lo largo del tiempo, para la recuperación posterior de versiones específicas, así como para su integración remota y continua con el resto del sistema (Git, nd). De esta forma, el equipo de desarrolladores puede mantener un control sobre los proyectos y trabajar siempre con la última versión, evitando problemas al emplear versiones desactualizadas del código.

Esta herramienta también facilita en gran medida la migración del código elaborado en el ambiente de desarrollo, una vez que ha sido lo suficientemente probado y validado, hacia el ambiente de producción, lo cual reduce las complicaciones del proceso y agiliza la puesta en producción de los incrementos de software en el servidor web.

A lo largo de este capítulo se han descrito los conceptos y tecnologías más relevantes para el desarrollo del presente Trabajo Especial de Grado. La figura 2.6 sintetiza las tecnologías involucradas en la implementación del mismo, mientras que en el siguiente capítulo se describe el método de desarrollo empleado para ello.

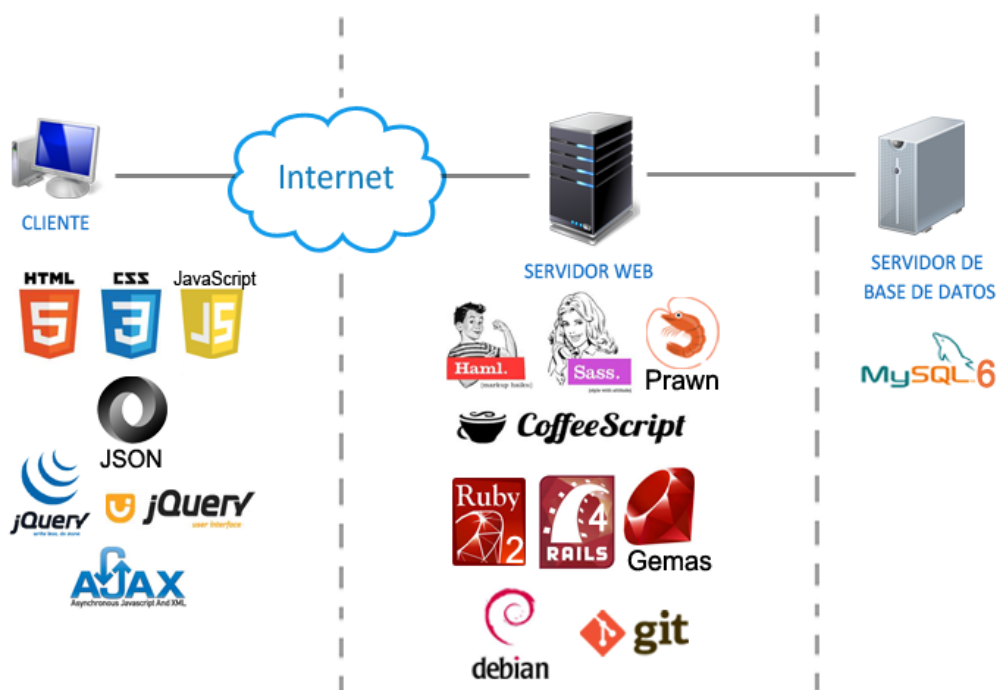


Figura 2.6: Tecnologías utilizadas para el desarrollo de la aplicación.

Capítulo 3. Proceso de desarrollo ágil

Los métodos de desarrollo de software proponen procedimientos para lograr la producción de software de calidad. Es allí donde radica la importancia de estudiar, seleccionar, adaptar y aplicar un método acorde con las necesidades del proyecto a ejecutar. El objetivo de este capítulo es presentar el método de desarrollo seleccionado y puesto en práctica durante la implementación del presente Trabajo Especial de Grado.

A manera de introducción, la primera sección de este capítulo trata sobre el Manifiesto Ágil, donde se enumeran los doce principios que han adoptado, en general, los procesos ágiles de desarrollo de software, y que han sido tomados en cuenta en el método seleccionado.

En la segunda sección, se explica el modelo de desarrollo iterativo e incremental, considerando sus aspectos positivos y negativos. El capítulo culmina definiendo el ciclo de vida utilizado para orientar el proceso de desarrollo del presente trabajo, el cual consta de cinco fases: Análisis, Diseño, Construcción, Pruebas y Despliegue, constituidas a su vez de un conjunto variado de tareas y actividades.

3.1. Manifiesto Ágil

Los procesos de desarrollo ágiles o livianos surgen como una alternativa a los procesos tradicionales, también llamados *pesados*, con el fin de responder de forma más rápida a los cambios que por lo general surgen en los proyectos de software, reducir los tiempos de desarrollo y la documentación generada a lo largo del mismo, así como controlar los riesgos, produciendo a la vez software de calidad.

Los procesos ágiles comúnmente adoptan una serie de principios y valores enumerados en el Manifiesto Ágil (Beck et al., 2001), el cual surgió a partir de un acuerdo entre un grupo de expertos de la industria del software, en febrero del año 2001. Según el Manifiesto Ágil, se valora:

- **Al individuo y sus interacciones** sobre los procesos y herramientas.
- **Al software que funciona** sobre la documentación extensiva.
- **La colaboración con el cliente** sobre la negociación contractual.
- **La respuesta ante el cambio** sobre el seguimiento de un plan.

Esto significa que aunque se valoran los elementos de la derecha, los de la izquierda tienen mayor peso. Estos valores sirven de inspiración a los siguientes doce principios, también enunciados en el Manifiesto Ágil.

1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Entregamos software funcional frecuentemente, en periodos entre dos semanas y dos meses, con preferencia al período de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
7. El software funcionando es la medida principal de progreso.
8. Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para ajustar y perfeccionar su comportamiento en consecuencia.

Algunos de los métodos ágiles de desarrollo de software que incorporan estos valores y principios son *eXtremme Programming (XP)*, *Agile Unified Process (AUP)*, Scrum y AgilUS.

3.2. Definición del método de desarrollo a utilizar

Según Pressman (2002), “los métodos de la Ingeniería de Software indican cómo construir técnicamente el software”. Estos métodos suelen estar compuestos de un conjunto de tareas agrupadas en actividades de Análisis de Requerimientos, Diseño, Construcción y Pruebas, que permiten organizar y llevar el control del proceso de desarrollo, con el fin de producir software de calidad. Durante estas actividades se obtienen productos de trabajo o artefactos, tales como documentos, modelos y prototipos. Existe una amplia gama de métodos de desarrollo de software, los cuales a grandes rasgos, varían en términos de flexibilidad, agilidad, énfasis en cada una de sus fases y documentación generada a lo largo del ciclo de vida. Sin embargo, en el marco de los métodos ágiles de desarrollo, es comúnmente aceptado que el desarrollador puede adaptar el método a las necesidades, tanto propias, como del proyecto.

En base a esto, tomando en cuenta los principios enunciados en el Manifiesto Ágil, y las características resaltantes de los métodos ágiles estudiados en el seminario elaborado como investigación previa al presente trabajo (Carneiro y Pérez, 2013), se decidió adoptar un método de desarrollo iterativo e incremental, el cual está conformado por las fases de Análisis, Diseño, Construcción y Pruebas, incluyendo la fase de Despliegue. En algunas de ellas se aplica un conjunto de buenas prácticas promovidas por *eXtremme Programming* (Beck, 2000), así como algunas actividades y artefactos propuestos en AgilUs (Acosta, 2011). Este último método hace especial hincapié en aspectos relacionados con la usabilidad, por lo cual sirvió de base para definir y adaptar el método empleado a lo largo del desarrollo.

Cabe destacar que el sistema sobre el que se centra este TEG se caracteriza por presentar altos niveles de interactividad con los usuarios finales, por lo que la usabilidad es un aspecto fundamental a ser tomado en cuenta. En consecuencia, se le brindó una importancia considerable al diseño de interfaces de usuario usables, sin dejar a un lado la naturaleza cambiante de los requerimientos y la necesidad de un desarrollo ágil del sistema.

A continuación se describe de manera general el concepto del desarrollo iterativo e incremental, y se enumeran algunas de sus ventajas y desventajas.

3.2.1. Desarrollo iterativo e incremental

Según Ortiz (2012), el desarrollo incremental se enfoca en desarrollar partes o módulos del software en cada iteración, de forma tal que el sistema crece y evoluciona al incorporar nuevas funciones en cada ciclo de desarrollo hasta obtener el producto final; mientras que el desarrollo iterativo consiste en la revisión y mejoramiento del producto obtenido en cada ciclo, por lo que el sistema mejora a través de iteraciones sucesivas, sin añadir nuevas funcionalidades.

A su vez, Larman (2003), indica que un ciclo de vida iterativo e incremental se basa en el crecimiento y perfeccionamiento secuencial de un sistema a través múltiples ciclos de desarrollo, donde se aborda un conjunto relativamente pequeño de requerimientos en cada iteración, pasando por las fases de Análisis, Diseño, Construcción y Pruebas. Al unir ambos enfoques, se obtiene un modelo que ofrece las siguientes ventajas:

- Permite priorizar los requisitos según el criterio de las partes interesadas, lo cual sirve de base para guiar el proceso de desarrollo y obtener resultados de forma incremental.
- Se obtiene retroalimentación frecuente por parte de los usuarios en base a los incrementos operativos, lo cual permite ajustar oportunamente el software a sus necesidades.
- Se reduce el costo en cuanto al tiempo y esfuerzo que conllevan los cambios que pueden producirse a lo largo del desarrollo.
- Es más fácil realizar pruebas, depurar y ajustar cada incremento de software.

En cuanto a las restricciones o posibles desventajas de este modelo se tienen las siguientes:

- Se requiere una alta disponibilidad de los usuarios, dado que se requiere su participación continua durante todo el desarrollo del proyecto (Albaladejo, nd).
- Al no contar con todos los requerimientos definidos desde el inicio del desarrollo, pueden surgir dificultades a la hora de diseñar la arquitectura que dará soporte al proyecto si no se tiene una visión global del mismo. (Ortiz, 2012)
- Aunque se promueve la poca documentación para agilizar los tiempos de respuesta, Joskowicz (2008) indica que es necesario pensar en el posterior mantenimiento del sistema, y lo que sucederá cuando ocurran cambios de personal en el equipo de desarrollo, por lo que siempre conviene mantener la mínima documentación posible.

La figura 3.1 ilustra el desarrollo iterativo e incremental para un proyecto de software.

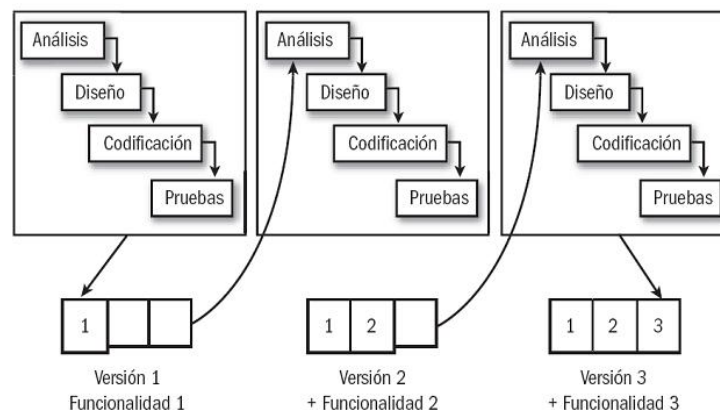


Figura 3.1: Modelo iterativo e incremental.

3.2.2. Descripción de las fases

El ciclo de vida del modelo iterativo e incremental aplicado en el presente TEG está compuesto por las fases de Análisis, Diseño, Construcción, Pruebas y Despliegue. Por otra parte, las iteraciones están orientadas al cumplimiento de objetivos, es decir, no están restringidas a un límite tiempo determinado, por lo que una iteración culmina cuando se han implementado los objetivos o requerimientos definidos en la planificación.

Además, debido a la alta interactividad que caracteriza a este sistema, se toman como referencia algunas actividades, técnicas y artefactos del método AgilUs, con el objeto de mantener un enfoque centrado en la usabilidad de la aplicación, a fin de contribuir con su calidad.

A continuación se describen las fases y actividades que constituyen este modelo de desarrollo.

3.2.2.1. Análisis

El objetivo de esta fase es comprender el alcance del problema, identificar los perfiles de usuario, roles involucrados y sus necesidades, e identificar y analizar los requerimientos funcionales y no funcionales del sistema. Cabe recordar que, de acuerdo a los principios ágiles, no es necesario conocer todos los requerimientos en detalle al inicio del ciclo de desarrollo, puesto que se trabaja sobre un conjunto reducido de los mismos en cada iteración. Para llevar a cabo esta fase, las actividades y artefactos elaborados se describen a continuación:

- Reuniones y sesiones de discusión con los usuarios para determinar los requerimientos funcionales y no funcionales del sistema, así como la priorización de los mismos. De estas reuniones también surge la planificación del proyecto, la cual puede ser controlada a través de herramientas colaborativas en línea, junto con la entrega de avances.
- Tormenta de ideas, que consiste en la generación de ideas por parte del equipo de desarrollo, estimulando la creatividad. El resultado de una sesión es un conjunto de buenas ideas que pueden aplicarse para resolver un problema. Esta técnica por lo general se aplica en las etapas tempranas del proceso de desarrollo, a fin de determinar los requerimientos de usabilidad del sistema a desarrollar. (Acosta, 2011)
- Perfiles de usuario, que están compuestos por datos sobre los usuarios del sistema, para brindar servicios de forma personalizada. Entre los datos que componen un perfil de usuario básico se tienen características personales que pueden influir en la interacción con el sistema, tales como la edad y el sexo, nivel de instrucción, nivel de experiencia en el manejo de aplicaciones web, destrezas, entre otros.

Estas actividades y artefactos pueden realizarse de manera opcional en cada iteración, dependiendo de las necesidades de información y de lo que sea más apropiado para el conjunto de requerimientos a desarrollar en una iteración en particular. Por ejemplo, los perfiles de usuario se realizan una vez, en la primera iteración, por lo que esta información aplica para el resto de las iteraciones subsiguientes.

3.2.2.2. Diseño

La fase de Diseño, según Pressman “traduce requisitos en una representación del software donde se pueda evaluar su calidad antes de que comience la codificación”. El diseño permite abordar diversas soluciones a los requerimientos derivados de la fase de Análisis, y el objetivo es proporcionar la arquitectura base sobre la que se implementará el sistema en la fase de Construcción. Contar con un buen diseño facilita en gran medida el proceso de codificación, puesto que las decisiones cruciales se toman y validan en esta fase, haciendo más sencilla la tarea de traducir el diseño aprobado por las partes interesadas en un producto terminado.

Entre las decisiones a tomar en esta fase, se encuentran: la estructura de datos que soportará a la aplicación, el diseño de interfaces de usuario estandarizadas y reutilizables, la disposición de sus elementos, metáforas y colores, tomando en cuenta aspectos de usabilidad, el formato de los reportes o documentos a generar mediante el sistema, y las tecnologías más apropiadas para implementar los requerimientos. Debido a ello, esta es una fase de suma importancia, donde para la toma de algunas decisiones se requiere la participación activa de los usuarios.

Respecto a la documentación, se busca generar la mínima posible, haciendo uso de herramientas que faciliten y agilicen esta tarea, tales como las herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora). Los artefactos que se producen en esta fase pueden ser:

- Elaboración de gráficos sencillos para diseñar el flujo de comunicación entre los actores que participan en los procesos. Existen diversas aplicaciones en línea gratuitas que facilitan la elaboración de diagramas, un ejemplo de ellas es la herramienta web *Cacoo*.
- En cuanto al diseño de la interfaz de usuario, se elaboran prototipos en papel, los cuales permiten realizar las primeras evaluaciones de usabilidad, así como su posterior refinamiento. Debido a ello, esta fase puede tener un carácter iterativo, con el objeto de lograr el diseño de un prototipo lo más cercano posible al producto que se desea obtener. Existen muchas herramientas en línea y programas para elaborar los prototipos, pero en ocasiones el prototipado con lápiz y papel resulta más práctico.
- Esquemas de estructuras de datos para el diseño del almacenamiento persistente de la información. Existen herramientas como *MySQL Workbench* que facilitan esta tarea y

generan código *SQL* en base al diseño previamente elaborado, listo para ser importado en la base de datos.

Una vez que se tienen los artefactos mencionados anteriormente, es posible evaluar algunos atributos de calidad antes de iniciar el proceso de codificación, tales como la seguridad, usabilidad, eficacia, así como determinar el grado de reusabilidad y extensibilidad de la aplicación. Por ende, se pueden realizar oportunamente los ajustes pertinentes para obtener un diseño simple y sólido, procediendo así a la fase de Construcción.

3.2.2.3. Construcción

Pressman indica que esta fase consiste en “traducir el diseño de forma tal que sea legible por la máquina. Si se lleva a cabo el diseño de una forma detallada, la generación de código se realiza mecánicamente”. Al usar un enfoque iterativo e incremental, se busca producir prototipos funcionales rápidamente para que los mismos sean evaluados por el usuario y recibir la retroalimentación pertinente con el objeto de realizar mejoras y ajustes.

En esta fase, se proceden a codificar los requerimientos de acuerdo a las prioridades determinadas por el usuario, añadiendo y extendiendo las funcionalidades del sistema de manera incremental. La participación de los usuarios es mínima durante la construcción del software. Entre algunas de las actividades propuestas (Jaldín, 2010) y buenas prácticas a tomar en cuenta, se encuentran las siguientes:

- Contar con un entorno de desarrollo similar al entorno de producción para facilitar el despliegue. Asimismo, instalar y configurar adecuadamente las herramientas de trabajo, tales como: editores de texto o IDE (Entorno de Desarrollo Integrado), el *framework* a emplear, herramientas CASE, servidores de base de datos, entre otros; para que el proceso de desarrollo sea sencillo, rápido y fluido.
- Definir y aplicar estándares de codificación, ya que esto permite producir código consistente, legible, y se facilita la colaboración con el resto del equipo de desarrolladores. Esta práctica es ampliamente recomendada en métodos ágiles como XP.
- Es importante utilizar un sistema de control de versiones, como *Git*, para la integración del código.
- Mantener el código lo más simple posible, y dejar la optimización para el final, puesto que cuando se tenga la funcionalidad terminada será posible evaluar lo que requiere ser optimizado.
- A lo largo del desarrollo, y después de varias iteraciones, puede ser necesario refactorizar el código del sistema para eliminar redundancia, facilitar los cambios y simplificar el mantenimiento de la aplicación, sin modificar su comportamiento.

3.2.2.4. Pruebas

La fase de pruebas consiste en evaluar el comportamiento del software, para verificar que satisface los requerimientos de los usuarios. Para llevar a cabo esta actividad, pueden realizarse pruebas de aceptación dirigidas por el usuario, así como pruebas automatizadas, con el objeto de asegurar el funcionamiento apropiado de la aplicación y constatar que las interfaces son usables para todos los roles involucrados en el manejo del sistema. Estas técnicas se describen a continuación:

- Pruebas de aceptación: Son conducidas por el usuario para verificar que el sistema satisface los requerimientos funcionales y no funcionales (Acosta, 2011). Los resultados de estas pruebas pueden recolectarse a través de reuniones con los usuarios, haciendo demostraciones del comportamiento del sistema en diversos escenarios y registrando las observaciones o comentarios de los mismos. La retroalimentación de los usuarios determinará los cambios o ajustes a realizar, y las situaciones a corregir para el cumplimiento adecuado de los requerimientos del sistema.
- Las pruebas automatizadas a través de herramientas como Cucumber permiten escribir pruebas de aceptación en un lenguaje que puede ser comprendido tanto por los desarrolladores como por los usuarios, y ejecutarlas de forma automatizada, sirviendo además, sirven como documentación del sistema (Suárez, 2013). Las pruebas automatizadas resultan útiles tanto en la fase de Construcción, así como en las posteriores a ella, permitiendo la ejecución de diversos escenarios, invirtiendo a su vez poco tiempo y esfuerzo.

3.2.2.5. Despliegue

Cuando la aplicación (o una parte de ella) ha sido desarrollada y suficientemente probada, se procede a su liberación en el entorno de producción. Esta fase es crítica, puesto que determina si la aplicación se comportará como está previsto en el ambiente real para el cual está destinada.

Incluso si el desarrollo se realizó en un entorno similar al de producción, es posible que surjan imprevistos, tales como casos o escenarios no contemplados, mayor tráfico y volumen de información que puede ralentizar o afectar el servicio brindado por la aplicación, consultas que consumen mayor tiempo debido a la cantidad de registros en las bases de datos en producción, entre otros. En consecuencia, es posible que se deban optimizar operaciones para que la aplicación ofrezca las mejores prestaciones a los usuarios finales, así como realizar ciertos ajustes, añadir o extender funcionalidades, pudiendo iniciar un nuevo ciclo de desarrollo, lo que denota la naturaleza iterativa del proceso. Además, en esta fase se debe brindar un seguimiento y soporte para asegurar la estabilidad de la aplicación a lo largo del tiempo.

Las actividades más relevantes a realizar durante el despliegue son:

- Instalación y configuración de la aplicación en el servidor de producción. Esta tarea se simplifica a través de herramientas como *Git* para la integración del código. También puede ser necesaria la migración de las estructuras de datos que soportan la aplicación hacia las bases de datos de producción.
- Capacitar a los usuarios de la aplicación, especialmente a aquellos que no estuvieron involucrados en el proceso de desarrollo, para que hagan buen uso de la misma, explicando cómo realizar las tareas correspondientes utilizando el nuevo sistema, resolver las dudas que puedan surgir al respecto, y señalar los beneficios para que la transición sea más fácil.
- Brindar soporte y seguimiento del desempeño de la aplicación para responder ante las contingencias de forma oportuna, a fin de ofrecer soluciones en el menor tiempo posible.

A manera de resumen, la figura 3.2 sintetiza las fases y actividades descritas en esta sección.



Figura 3.2: Ciclo de vida y actividades del modelo de desarrollo empleado.

En el siguiente capítulo se procede a detallar cada una de las iteraciones llevadas a cabo para el desarrollo del presente trabajo, así como los resultados obtenidos al aplicar el modelo de desarrollo iterativo e incremental explicado en este capítulo.

Capítulo 4. Desarrollo del módulo solicitudes

Durante la puesta en práctica del modelo iterativo e incremental especificado en el capítulo anterior, se ha realizado una serie de iteraciones para llevar a cabo la implementación de las funcionalidades relacionadas con las solicitudes de jurado para seminario y Trabajo Especial de Grado, solicitudes de planillas de notas, solicitudes de modificación de inscripción y solicitudes de reincorporación.

El propósito del presente capítulo es describir las actividades y artefactos realizados en cada fase del modelo de desarrollo seleccionado, así como presentar los resultados obtenidos al final de cada iteración. Cabe recordar que las iteraciones en este modelo son orientadas a objetivos, por lo cual no están basadas en el tiempo de su duración, sino que el enfoque es mostrar el producto final obtenido en cada una de ellas.

Es importante acotar que las iteraciones están divididas de acuerdo a los procesos y de acuerdo a la perspectiva de los roles involucrados en los mismos. Las fases de Análisis y Diseño fueron ejecutadas de forma general para todos los roles, recabando la información relevante desde la primera iteración correspondiente a cada proceso. En las iteraciones intermedias, se resalta la fase de Construcción, ya que es la fase más variable, donde se implementan los requerimientos específicos para cada rol, mientras que en la iteración final de cada proceso es donde se incluyen las fases de Pruebas y Despliegue, aplicadas una vez que el conjunto de funcionalidades relativas a un proceso han sido desarrolladas.

4.1. Iteración 1 - Solicitud de jurado para TEG

Se decide iniciar la implementación del sistema a partir de las solicitudes de jurado para TEG debido a la menor complejidad de las funcionalidades y del comportamiento general de sistema, como primer acercamiento y para tener un período de adaptación a la dinámica de desarrollo. Además, cronológicamente, fue más conveniente la implementación de estas funcionalidades, puesto que se requería realizar la puesta en producción de la versión 3.0 del sistema CONEST con estos requerimientos integrados. A continuación, se describen las actividades realizadas en esta iteración.

4.1.1. Análisis

En la fase de Análisis, se realizaron reuniones con los usuarios para describir el proceso de las solicitudes de jurado, discutir el alcance y las principales funcionalidades que debería satisfacer el sistema, esto con el fin de determinar los requerimientos funcionales y no funcionales de la aplicación. Las fases de Análisis y Diseño se realizaron en conjunto para los roles involucrados en el proceso de solicitudes de jurado y solicitudes de planilla de notas debido a los aspectos comunes entre ellos, tales como los requerimientos y elementos de la interfaz de usuario, con el objeto de recopilar la mayor cantidad de información posible y agilizar el proceso de desarrollo. Por esta razón, las actividades y artefactos concernientes a ambas fases aplican también para las iteraciones subsiguientes.

4.1.1.1. Tormenta de ideas

El resultado de las múltiples reuniones y discusiones llevadas a cabo durante esta fase fue una tormenta de ideas, donde surgieron propuestas y soluciones para solventar los problemas que presenta el proceso actual de las solicitudes de jurado, así como el de la solicitud de la planilla de notas.

Algunas de estas ideas, que a su vez se traducen en requerimientos funcionales y no funcionales, se listan a continuación.

- En cuanto a las funcionalidades básicas, se requieren las siguientes:
 - Creación, consulta, edición, aprobación y anulación de solicitudes de jurado para Trabajo Especial de Grado.
 - Solicitud, impresión y anulación de planillas de notas para seminario y TEG.
 - Registrar jurados externos y modificación de jurados para solicitudes y planillas de notas.
 - Búsquedas por estudiante, envío de notificaciones por correo a las partes involucradas en este proceso, entre otras. Todas estas funcionalidades requieren una permilogía que varía de acuerdo a cada rol.
- Este proceso incluye la generación de documentos en formato *PDF (Portable Document Format)*, tales como:
 - Solicitud de jurado para TEG: Solicitada por el personal docente, a ser entregada en la escuela respectiva para su evaluación y posterior aprobación.

- Carta de aprobación de la solicitud de jurado: Generada por las secretarías de cada escuela con el fin de notificar al jurado y a la División de Control de Estudios sobre la aprobación de la solicitud de jurado.
 - Planilla de notas para seminario y TEG: Solicitada por el docente a la División de Control de Estudios para registrar y consignar la calificación obtenida por un estudiante luego de su evaluación por parte del jurado.
 - Constancia de tutoría: Solicitada por los docentes y emitidas por el personal de la División de Control de Estudios, con el objetivo de certificar los seminarios o Trabajos Especiales de Grado en los que un docente ha participado como tutor.
- Los datos más relevantes a ser almacenados son el autor del trabajo a presentar, el título y el jurado que lo evaluará, así como datos relacionados con la aprobación de la solicitud.
 - Las interfaces de usuario son un aspecto muy importante, ya que deben favorecer el reconocimiento de la información para reducir la carga cognitiva de los usuarios que interactúan con la aplicación. Para la organización de los datos se prefiere el uso de tablas, con información concreta y resumida. A su vez, se deben reunir múltiples funciones en la misma interfaz, utilizando íconos o metáforas intuitivas que reflejen las acciones pertinentes sobre cada solicitud. Los diferentes usuarios deben visualizar una interfaz estandarizada, con las funcionalidades pertinentes de acuerdo a su rol.

La figura 4.1 resume el conjunto de ideas que se originaron durante esta actividad.

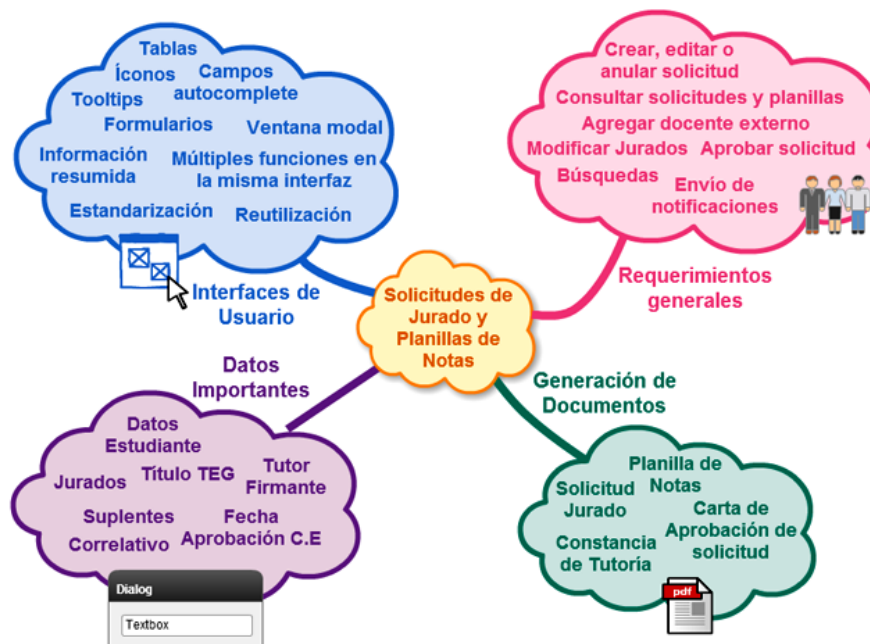


Figura 4.1: Tormenta de ideas para las solicitudes de jurado.

4.1.1.2. Perfiles de usuario

Los perfiles de usuario contienen los datos más relevantes sobre los distintos tipos de usuario que interactúan con el sistema. Entre estos están los estudiantes, los docentes, el personal de las escuelas, y el personal administrativo de la División de Control de Estudios. Estos perfiles se muestran a continuación a través de una serie de tablas, y son globales a todas las iteraciones presentadas en este documento.

Tabla 4.1: Perfil de usuario - Estudiante de pregrado.

Estudiante de Pregrado	
Edad	Puede variar ampliamente, pero se ubica entre los 16 y 25 años de edad.
Sexo	Depende de la Escuela, puesto que en algunas hay predominancia de estudiantes de determinado sexo.
Nivel de Instrucción	Educación Media.
Experiencia en tecnologías	En general, los estudiantes presentan un buen manejo de las tecnologías y aplicaciones web.

Tabla 4.2: Perfil de usuario - Docente.

Docente	
Edad	Varía desde los 23 años hasta adultos mayores.
Sexo	Varía en cada Escuela, ya que en algunas predomina cierto sexo. Por ejemplo, en la Escuela de Biología hay predominancia del sexo femenino, mientras que en la Escuela de Física predomina el sexo masculino.
Nivel de Instrucción	Educación Universitaria. La mayoría de los docentes de la Facultad tiene al menos un Postgrado.
Experiencia en tecnologías	Independientemente de su área de desempeño en la Facultad, los docentes manejan aplicaciones web de manera efectiva.

Tabla 4.3: Perfil de usuario - Personal de las escuelas.

Personal de las Escuelas	
Edad	Comprendida entre los 35 y 45 años de edad.
Sexo	El personal está constituido principalmente por personas del sexo femenino.
Nivel de Instrucción	Educación Media, Técnico Superior Universitario (T.S.U.), Educación Universitaria. También son profesionales en otras áreas de trabajo.
Experiencia en tecnologías	Posee un buen manejo de las aplicaciones de ofimática y aplicaciones web.

Tabla 4.4: Perfil de usuario - Personal de la DCE.

Personal de la División de Control de Estudios (DCE):	
Edad	Comprendida entre los 18 y 45 años de edad.
Sexo	En su mayoría, el personal de la DCE está compuesto por personas del sexo femenino.
Nivel de Instrucción	Educación Media, Técnico Superior Universitario (T.S.U.), Educación Universitaria. El personal de la DCE es muy variado, ya que está compuesto por pasantes, auxiliares docentes, y personal profesional en áreas como Administración, Archivología, Informática, entre otras. Además, este personal tiene un alto dominio de los reglamentos, lineamientos y flujos de todos los procesos académicos de la Facultad.
Experiencia en tecnologías	Posee un manejo avanzado de las aplicaciones web. Debido a razones de seguridad, se deben mantener soportes físicos en todo momento (expedientes, documentos, oficios) para garantizar la integridad y autenticidad de la información en caso de que los sistemas de gestión fallen.

Específicamente en el proceso de las solicitudes de jurado, los roles participantes son el docente, el personal de las escuelas (secretarías) y el personal de la DCE. Aunque se trata de una solicitud estudiantil, el estudiante no participa en el proceso como tal, puesto que el docente es quien realiza la solicitud en nombre del estudiante bajo su tutela.

4.1.2. Diseño

Después de obtener, organizar y analizar la información referente a los procesos llevados a cabo por los roles involucrados en las solicitudes de jurado, se procede a diseñar la secuencia de las acciones y sintetizar las funcionalidades básicas del sistema mediante gráficos simples, con el objeto de que los mismos sirvan para orientar la automatización que se desea implementar en este proceso.

Se toma como punto de partida la necesidad del tutor de realizar una solicitud de jurado para TEG. Para ello, este debe ingresar al sistema en su rol de docente, y realizar la solicitud indicando los datos relevantes a través de un formulario. El tutor podrá editar la solicitud e imprimirla en formato *PDF* a través de la interfaz de consulta correspondiente. Este documento debe ser entregado junto con el *kardex* del estudiante en la escuela pertinente para su aprobación en Consejo de Escuela. Cuando este evento se produzca, tanto el jurado como el estudiante serán informados mediante un correo de notificación. Luego, se habilitará una función que permite al tutor solicitar a través del sistema la planilla de notas del estudiante evaluado, la cual puede retirar en la División de Control de Estudios. La figura 4.2 ilustra de forma general esta secuencia de pasos.

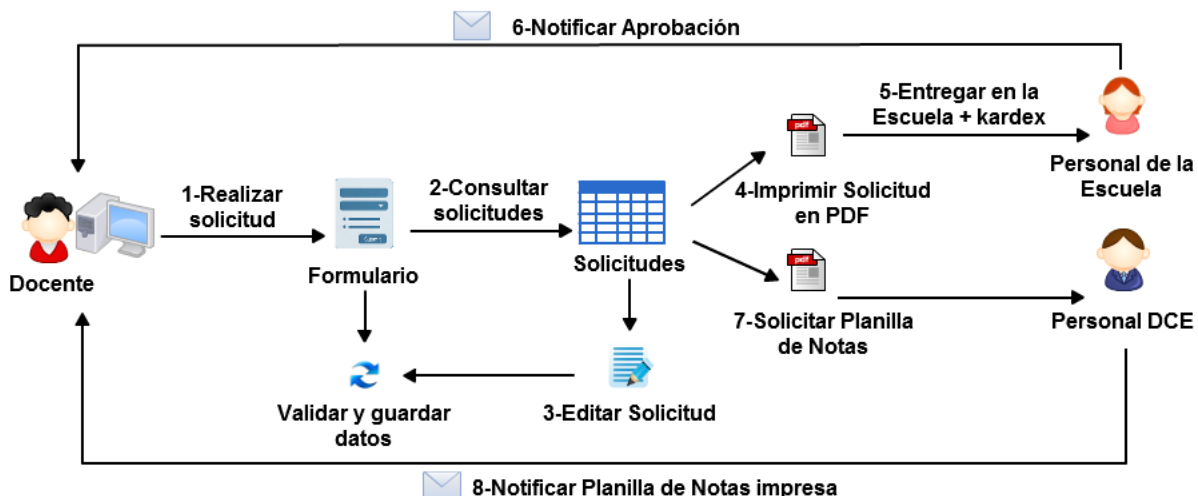


Figura 4.2: Gestión de solicitudes de jurado para el docente.

El personal administrativo de las escuelas es quien recibe y procesa a través del sistema las solicitudes a ser evaluadas en sesión de Consejo de Escuela. Dependiendo de lo que se decida, el personal puede editar los datos de la solicitud para su aprobación final. Posteriormente, las partes interesadas son notificadas vía correo electrónico, y se genera una carta que avala los datos aprobados en la solicitud, con el objeto de ser entregada al Jefe de la División de Control de Estudios para que sea archivada en el expediente del estudiante. Es necesario resaltar que el personal administrativo de cada escuela también puede iniciar el proceso de solicitud de jurado a petición del docente. La figura 4.3 resume el proceso anteriormente descrito.

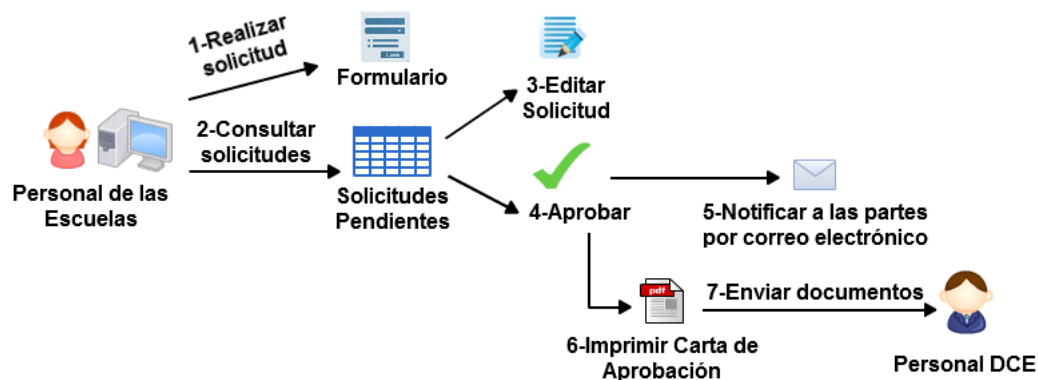


Figura 4.3: Gestión de solicitudes de jurado para el personal de las escuelas.

El personal administrativo de la División de Control de Estudios, al poseer un rol con altos privilegios, puede consultar todas las solicitudes, y ejecutar sobre ellas las acciones correspondientes al docente y al personal de las escuelas. Además, puede imprimir (verificando los requisitos de *pensum* necesarios para ello), o anular las planillas de notas solicitadas por los

docentes. El tutor es notificado vía correo electrónico cuando la planilla de notas esté lista para ser retirada. La figura 4.4 representa este conjunto de funcionalidades.

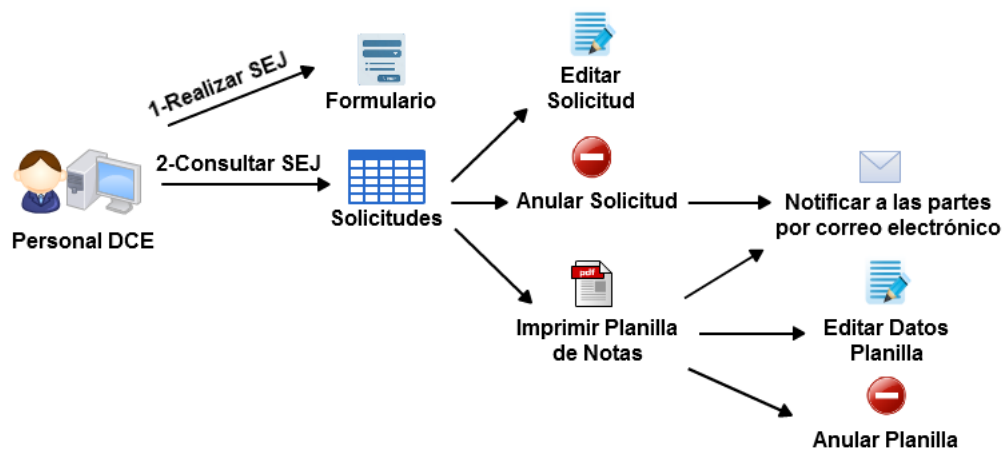


Figura 4.4: Gestión de solicitudes de jurado para el personal de la DCE .

Respecto al diseño del modelo de datos, como ya se cuenta con una base de datos existente, sólo es necesario extender los campos de algunas tablas relacionadas con las solicitudes para poder almacenar datos como la fecha de aprobación de las solicitudes y su correlativo. También se agregan nuevos estados para el manejo de las mismas.

Para la identificación rápida de funcionalidades relacionadas con las solicitudes de jurado y el resto de los procesos tratados en este TEG, se ha seleccionado un conjunto de íconos gratuitos de licencia libre a través de la página <https://www.iconfinder.com/> (ver figura 4.5). El uso de íconos para representar acciones facilita la agrupación de las funcionalidades y su unificación en la interfaz de usuario.



Figura 4.5: Conjunto de íconos seleccionados para la interfaz de usuario.

4.1.2.1. Prototipos de interfaz

Durante esta fase se elaboran algunos prototipos de interfaz de usuario para evaluar los principales aspectos de usabilidad a considerar en el desarrollo del sistema. A continuación se muestra un prototipo de consulta de solicitudes para el personal administrativo de la DCE, ya que es la más compleja y engloba las funcionalidades de los roles de menor permisología.

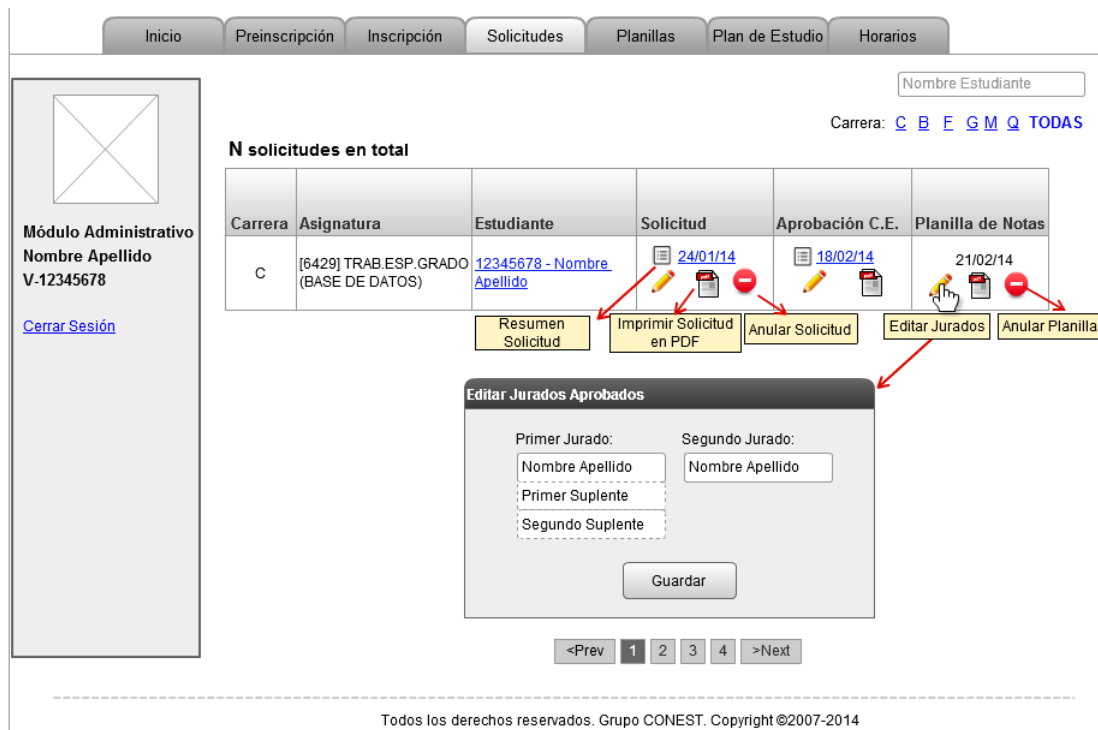


Figura 4.6: Prototipo de interfaz de consulta de solicitudes de jurado para el personal de la DCE.

En la figura 4.6 se aprecia la interfaz de consulta de solicitudes, presentada a través de una tabla cuyas columnas reflejan el flujo de procesos de las solicitudes de jurado, y donde se organiza la siguiente información:

- Datos sobre la carrera y la materia correspondiente a la solicitud, así como datos del estudiante.
- En la columna *Solicitud* se tiene un conjunto de acciones a ejecutar sobre la misma. Se puede consultar un resumen de la solicitud a través de una ventana modal, editar la solicitud, imprimirla en formato *PDF*, o anularla. El rol del docente es quien presenta mayor interacción sobre las funcionalidades que contiene esta columna, puesto que representa el paso inicial del proceso.
- La columna *Aprobación C.E.*, cuyas funcionalidades conciernen principalmente al personal administrativo de las escuelas, permite aprobar la solicitud indicando datos como la fecha de la sesión en que se llevó a cabo el Consejo de Escuela, editar estos datos, o generar una carta de notificación de aprobación de la solicitud en formato *PDF*.
- La última columna, *Planilla de Notas*, permite al personal administrativo de la DCE imprimir en formato *PDF* aquellas planillas que han sido previamente solicitadas por los docentes. Tanto el docente como el administrador tienen la posibilidad de editar los jurados que han sido aprobados en Consejo de Escuela, sustituyendo a los jurados principales

Capítulo 4. Desarrollo del módulo solicitudes

por suplentes en caso de ser necesario. Además, también es posible anular la solicitud de la planilla de notas si el caso lo amerita.

En la parte superior derecha de la figura 4.6, referida anteriormente, se muestra un campo a través del cual es posible realizar búsquedas de solicitudes por estudiante. Un poco más abajo se aprecia un selector de carreras, el cual permite filtrar las solicitudes dependiendo de la carrera seleccionada. Esta última funcionalidad sólo es visible para el personal administrativo de la DCE.

Por otra parte, la figura 4.7 muestra un detalle del conjunto de acciones disponibles para el personal de las escuelas o secretarías.

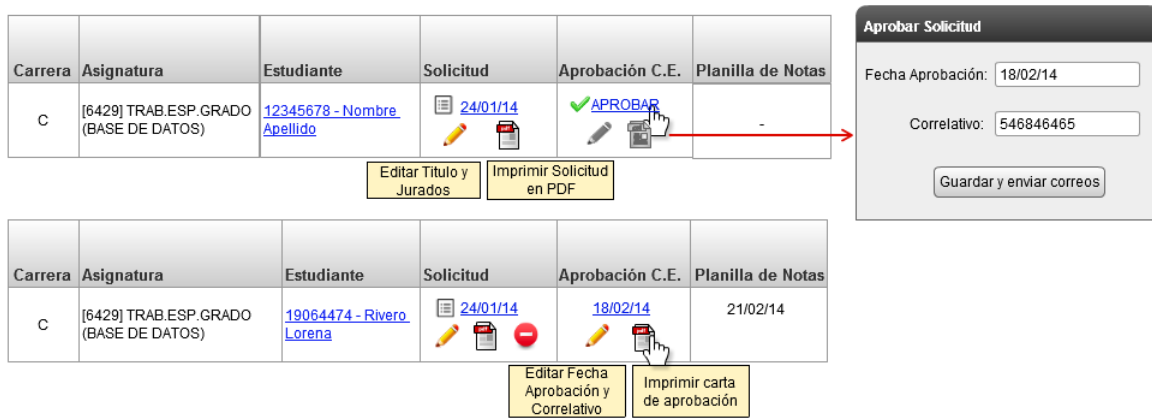


Figura 4.7: Prototipo de interfaz de consulta de solicitudes de jurado para el personal de las escuelas.

Con la aceptación de estos prototipos por parte de los usuarios, se procedió a la fase de Construcción, presentada a continuación.

4.1.3. Construcción - Perspectiva del docente

Para la construcción de las funcionalidades relativas a los procesos tratados en este trabajo, se sigue un conjunto de estándares de codificación común para todo el equipo de desarrollo de CONEST. Entre ellos, se encuentran los siguientes:

- Evitar el uso de caracteres especiales y acentos en nombres de variables, métodos, clases, nombres de tablas y atributos, así como en nombres de archivos. Además, estos nombres deben estar en español y deben ser lo suficientemente descriptivos para dar a entender su objetivo o comportamiento.
- Uso razonable de comentarios para la documentación del código.

Capítulo 4. Desarrollo del módulo solicitudes

- Respetar las convenciones de *Ruby on Rails* en cuanto a la ubicación de las vistas, controladores, modelos, y demás archivos en los directorios correspondientes del proyecto para facilitar su mantenimiento.
- Utilización de herramienta para el control de versiones e integración del código a través de *Git*, enviando porciones de código de forma periódica a lo largo del desarrollo.

A continuación, se describe el producto obtenido al poner en práctica estos aspectos y después del refinamiento de sucesivos prototipos funcionales desarrollados durante el proceso de codificación.

Cuando un tutor desea realizar una solicitud de jurado, ingresa al sistema bajo el rol docente, y accede a un campo de búsqueda a través de un menú dispuesto para ello. En este campo, el docente puede ingresar la cédula de identidad o el nombre del estudiante a nombre de quien se realiza la solicitud. Entre los requisitos que debe cumplir un estudiante, se tiene:

- Haber inscrito la asignatura Trabajo Especial de Grado en el semestre actual.
- La asignatura inscrita debe estar sin calificar.
- No poseer una solicitud de jurado en curso, con la excepción de que la misma esté anulada. Si el estudiante ya posee una solicitud pendiente realizada por el docente, se mostrará un formulario para su edición.

En el caso de la Escuela de Computación, un seminario o Trabajo Especial de Grado puede ser realizado por una pareja de estudiantes, por lo cual los requisitos anteriores deben cumplirse para ambos. Una vez validados estos requisitos, el docente puede llenar los datos de la solicitud a través de un formulario, donde deberá ingresar el título del TEG y la proposición de jurados a evaluar el trabajo. Si se desea incluir un jurado externo a la facultad, el docente dispone de un enlace que permite agregar al docente externo, luego de introducir algunos datos básicos (ver figura 4.8).

Solicitud de Jurado para TEG

Buscar

Estudiante

18760518 - CORDOVA ARANGUREN AF

Carrera: COMPUTACION
Asignatura: (6420) TRAB.ESP.GRADO(APLI.INTE)

Datos del Trabajo Especial de Grado

Los campos con asterisco (*) son obligatorios.

Titulo del Trabajo *

Resumen

Palabras clave

Proposición de Jurado

Agregar Calificador Externo

Tutor Firmante * Tutor

Cédula o Nombre del Docente Cédula o Nombre del Docente

Primer Jurado * Segundo Jurado *

Cédula o Nombre del Docente Cédula o Nombre del Docente

Primer Suplente * Segundo Suplente *

Cédula o Nombre del Docente Cédula o Nombre del Docente

Agregar Calificador Externo

Nacionalidad * Sexo *

VENEZOLANO MASCULINO

Cédula *

Cédula sin puntos

Nombres *

Primer y Segundo Nombre

Apellidos *

Primer y Segundo Apellido

Correo *

algo@ejemplo.com

Teléfono Oficina *

0212-1234567

Teléfono Celular

0412-1234567

Procedencia

Nombre Organización

Título

Título

Guardar

Enviar

Figura 4.8: Formulario para realizar la solicitud de jurado.

En este punto del proceso, se le otorga especial atención a la validación de los datos de entrada del formulario, así como a los mensajes de retroalimentación para que sean lo suficientemente informativos y breves. Luego de validar y procesar los datos, el docente visualiza la interfaz de consulta a través de una tabla (ver figura 4.9), reflejando la solicitudes en orden de creación. Una solicitud de jurado tiene tres estados posibles: nueva, aprobada o anulada, siendo el primero el estado inicial.

En la columna *Solicitud*, se puede ver mediante un enlace, el resumen de los datos de la misma, así como editar e imprimir la solicitud en formato *PDF* a través de los íconos dispuestos para ello. Cuando la solicitud es aprobada, la columna *Aprobación C.E.* refleja la fecha de aprobación, y además, en la columna *Planilla Notas*, se habilita un enlace por estudiante para

Capítulo 4. Desarrollo del módulo solicitudes

solicitar la planilla de notas, la cual luego de ser solicitada, proporciona un ícono que permite sustituir los jurados principales aprobados por aquellos jurados suplentes especificados. Es necesario destacar que después de la aprobación de una solicitud de jurado, el docente no podrá editarla, por lo que esta función se deshabilita para garantizar la integridad de los datos aprobados por el Consejo de Escuela.



Carrera	Asignatura	Estudiante	Solicitud	Aprobación C.E.	Planilla Notas
C	(6420) TRAB.ESP.GRADO(APLI.INTE)	19993937 - CAMEJO JUAN	29/06/2014 	02/07/2014	07/07/2014
C	(6420) TRAB.ESP.GRADO(APLI.INTE)	20174430 - PARDO ALEJANDRO	29/06/2014 	02/07/2014	07/07/2014
C	(6414) TRA.ESP.GRADO(ING.SOFTW.)	17387508 - MARTIN ADRIAN	17/06/2014 	18/06/2014	20/06/2014

Figura 4.9: Interfaz de consulta de solicitudes de jurado para el docente.

En el anexo E, ubicado en la página 116, se encuentra el formato final de la solicitud de jurado que el docente debe imprimir en formato *PDF* a través del sistema, el cual se basa en una plantilla que se empleaba anteriormente, y cuyo diseño fue aprobado por las diversas escuelas.

4.2. Iteración 2 - Solicitud de jurado para TEG

Durante esta iteración se implementan las funcionalidades relativas al personal de las escuelas, con el fin de que el mismo pueda recibir y procesar las solicitudes de jurado realizadas por los docentes. Además, este personal está autorizado para ejecutar las acciones correspondientes al docente, descritas en la iteración anterior.

4.2.1. Construcción - Perspectiva del personal de las escuelas

Los requerimientos implementados durante esta iteración consisten en la consulta, edición y aprobación de las solicitudes de jurado según las decisiones tomadas por el Consejo de Escuela. En la figura 4.10 se aprecia la interfaz de consulta de solicitudes para el personal de las escuelas.

Capítulo 4. Desarrollo del módulo solicitudes

Consulta de Solicitudes de Jurado					
Cédula o Nombre del Estudiante					
Solicitudes encontradas: 8 registros en total					
Carrera	Asignatura	Estudiante	Solicitud	Aprobación C.E.	Planilla Notas
C	(6420) TRAB.ESP.GRADO(APLI.INTE)	18760518 - CORDOVA ARTHUR	08/08/2014	✓ APROBAR	-
C	(6412) TRAB.ESP.GRADO(SIST.INF.)	7993641 - ALVAREZ EDUARDO	15/07/2014	16/07/2014	Solicitar
C	(6412) TRAB.ESP.GRADO(SIST.INF.)	16495102 - PORCO CARMELA	15/07/2014	16/07/2014	Solicitar
C	(6420) TRAB.ESP.GRADO(APLI.INTE)	18738592 - ROSALES ROGER 18761047 - DIAZ SARAI	07/07/2014	16/07/2014	Solicitar Solicitar
C	(6420) TRAB.ESP.GRADO(APLI.INTE)	19993937 - CAMEJO JUAN	29/06/2014	02/07/2014	07/07/2014
C	(6420) TRAB.ESP.GRADO(APLI.INTE)	20174430 - PARDO ALEJANDRO	29/06/2014	02/07/2014	07/07/2014

Figura 4.10: Consulta de solicitudes de jurado para el personal de las escuelas.

La aprobación de una solicitud se realiza a través de un enlace que despliega una ventana modal, mostrada en la figura 4.11, donde se deben ingresar datos como la fecha de sesión del Consejo de Escuela donde se decidió la aprobación, y el correlativo de la solicitud, siendo este último un dato opcional.

Aprobar Solicitud

Se enviará un correo de notificación de aprobación al estudiante y jurado involucrado en la solicitud.

Datos de Aprobación

Fecha Sesión C.E.: *

Correlativo

Información a enviar

Datos Generales

Carrera: COMPUTACION Período Académico: 01-2014

Estudiante: 18760518 - CORDOVA ARTHUR Asignatura: (6420) TRAB.ESP.GRADO(APLI.INTE)

Datos TEG

Título TITULO DE PRUEBA

Jurado Aprobado

Tutor Firmante 13736933 - RIVAS SERGIO
 Jurado Principal 13872341 - ZAMBRANO JOSSIE Jurado Principal 82147710 - RAMIREZ ESMITT
 Jurado Suplente 12950975 - COTO ERNESTO Jurado Suplente 6814548 - ACOSTA ALECIA

Figura 4.11: Aprobación de solicitud de jurado.

Al realizar este procedimiento, las partes interesadas (jurados y estudiantes) son notificadas automáticamente por correo electrónico. Además, se habilita la función que permite descargar la carta de aprobación de solicitud de jurado en formato *PDF*, la cual es entregada posteriormente al Jefe de la División de Control de Estudios para avalar los datos aprobados de la solicitud. En el anexo F, en la página 117, se encuentra el formato final de esta carta, el cual fue diseñado y aprobado con la participación de las escuelas, permitiendo la estandarización de este documento con los datos requeridos para todas ellas.

Además, se incluye una funcionalidad que permite el reenvío de los correos con la información de las solicitudes aprobadas a las partes interesadas en caso de que se requiera realizar esta acción por errores en el envío, o por otros motivos. La figura 4.12 muestra la ventana modal, que simula un gestor de correos, y que permite la selección de un conjunto de destinatarios, constituido por los tutores, jurados y estudiantes involucrados.

Reenviar correo de aprobación

Se enviará un correo de notificación de aprobación con los datos actuales de la solicitud. Seleccione los destinatarios:

Destinatarios

Profa. Hernandez Yosly Prof. Silva Antonio Profa. Castillo Zenaida Prof. Bellorin Alberto
 Prof. Arias Pio Prof. Carmona Rhadames Br. Camejo Juan

Información a enviar

Datos Generales

Fecha C.E:	02/07/2014	Correlativo:	03.13.207
Carrera:	COMPUTACION	Período Académico:	01-2014

Estudiante: 19993937 - CAMEJO JUAN Asignatura: (6420) TRAB.ESP.GRADO(APLI.INTE)

Datos TEG

Título SISTEMA DE ARBITRAJE Y PRODUCCIÓN DE UNA REVISTA CIENTIFICO TECNOLÓGICA. CASO DE ESTUDIO: LACLO.

Jurado Aprobado

Tutor Firmante	15327864 - HERNANDEZ YOSLY	Tutor	9663887 - SILVA ANTONIO
Jurado Principal	5410603 - CASTILLO ZENAIDA	Jurado Principal	8382750 - BELLORIN ALBERTO
Jurado Suplente	7183922 - ARIAS PIO	Jurado Suplente	10804242 - CARMONA RHADAMES

Enviar correo

Figura 4.12: Reenvío de correos con información de la solicitud aprobada.

4.3. Iteración 3 - Solicitud de jurado para TEG

Las funcionalidades implementadas durante esta iteración corresponden a las relacionadas con el personal de la División de Control de Estudios, quien además de estar autorizado para ejecutar todas las funcionalidades descritas en las iteraciones anteriores, así como resolver los conflictos que puedan surgir durante el proceso de solicitud de jurado, está encargado de emitir documentos tales como planillas de notas y constancias de tutoría, a petición de los docentes.

4.3.1. Construcción - Perspectiva del personal de la DCE

Cuando el personal de la DCE desea hacer seguimiento de las solicitudes de jurado pendientes, debe ingresar a la interfaz de consulta presentada en la figura 4.13.

Consulta de Solicitudes de Jurado

Cédula o Nombre del Estudiante

Licenciatura: [B](#) | [C](#) | [E](#) | [G](#) | [M](#) | [Q](#) | [Todas](#)
 Período Académico: [01-2014](#)

Solicitudes encontradas: 10 registros en total

Carrera	Asignatura	Estudiante	Solicitud	Aprobación C.E.	Planilla Notas
C	(6420) TRAB.ESP.GRADO(APLI.INTE)	18760518 - CORDOVA ARTHUR	08/08/2014 	APROBAR 	-
Q	(3498) T. E.G. (OPC. TECNOLOGIA)	15638059 - GUERRA EDUIJS	25/07/2014 	19/01/2011 	25/07/2014
M	(8604) TRAB ESP GRADO MATEMATICA	20413139 - MARTINEZ KERLYNS	19/07/2014 	15/05/2014 	21/07/2014
C	(6412) TRAB.ESP.GRADO(SIST.INF.)	7993641 - ALVAREZ EDUARDO	15/07/2014 	16/07/2014 	Solicitar
C	(6412) TRAB.ESP.GRADO(SIST.INF.)	16495102 - PORCO CARMELA	15/07/2014 	16/07/2014 	Solicitar
C	(6420) TRAB.ESP.GRADO(APLI.INTE)	18738592 - ROSALES ROGER 18761047 - DIAZ SARAI	07/07/2014 	16/07/2014 	Solicitar Solicitar
C	(6420) TRAB.ESP.GRADO(APLI.INTE)	19993937 - CAMEJO JUAN	29/06/2014 	02/07/2014 	07/07/2014

Figura 4.13: Consulta de solicitudes de jurado para el personal de la DCE.

A través de esta interfaz se pueden filtrar las solicitudes por escuela, realizar búsquedas por estudiante, ejecutar las funciones correspondientes al docente y al personal de las escuelas, así como imprimir las planillas de notas solicitadas por los docentes, luego de verificar que los estudiantes cumplen con los requisitos pertinentes. El anexo G, en la página 118, presenta un ejemplo de planilla de notas para TEG.

Además, el personal de la DCE puede imprimir constancias de tutoría para seminario y TEG, las cuales son solicitadas por los docentes (de la facultad o externos a ella), con el fin de avalar aquellos trabajos donde han participado como tutores. Estas constancias pueden generarse opcionalmente con la firma digital del Jefe de la DCE, y también a partir de un semestre en particular. La figura 4.14 muestra la interfaz elaborada en la ficha del docente para este fin.

The image shows a web interface for generating tutorship certificates. It is divided into two main sections, each containing a table with two columns: 'Con firma' and 'Sin Firma'. The first section is titled 'Constancia de Tutoría' and the second is titled 'Por Semestre: 01-2014'. Both tables have two rows of options: 'Constancia de Tutoría TEG' and 'Constancia de Tutoría Seminario'.

Con firma	Sin Firma
Constancia de Tutoría TEG	Constancia de Tutoría TEG
Constancia de Tutoría Seminario	Constancia de Tutoría Seminario

Con firma	Sin Firma
Constancia de Tutoría TEG	Constancia de Tutoría TEG
Constancia de Tutoría Seminario	Constancia de Tutoría Seminario

Figura 4.14: Interfaz para la generación de constancias de tutoría para seminario o TEG.

Los anexos H e I, en la página 119 y 120 respectivamente, presentan un ejemplo de constancia de tutoría para TEG y seminario. Entre las mejoras incorporadas respecto a los formatos que existían anteriormente, se tiene la agrupación de estudiantes que participan en el mismo trabajo para el caso de los TEG, una organización clara de la información y la optimización de espacio para el ahorro considerable de papel.

4.4. Iteración 4 - Solicitud de planilla de notas

La solicitud de la planilla de notas, también llamada acta final de notas, se realiza con el fin de registrar la calificación obtenida por un estudiante en una asignatura determinada. Esta planilla, emitida por la División de Control de Estudios, consta de tres (3) copias, una para el docente, otra para el personal de secretaría, y la última es la copia para la DCE, la cual reside en el expediente del estudiante.

El proceso involucrado en la solicitud de planilla de notas para TEG, debido a su estrecha relación con las solicitudes de jurado, ha sido descrito de acuerdo a cada rol en iteraciones anteriores, por lo que a continuación se reseña el proceso de Construcción de las planillas de notas para seminario. Adicionalmente, se extendieron estas funcionalidades para soportar la generación de planillas de notas para pasantía, cuyo proceso es similar a las referidas anteriormente. Por esta razón, y para evitar redundar, el proceso de estas últimas no será reseñado.

4.4.1. Construcción - Planilla de notas para seminario

El proceso de solicitud de planillas de notas para seminario es ligeramente diferente al relacionado con las planillas de notas para TEG, puesto que el jurado propuesto no requiere ser aprobado previamente en Consejo de Escuela en algunas escuelas. Por ende, el docente

Capítulo 4. Desarrollo del módulo solicitudes

(o el personal administrativo) sólo debe buscar al estudiante en cuestión a través del sistema, e ingresar los tutores y jurados que evaluarán el seminario (ver figura 4.15).

Figura 4.15: Ventana modal para solicitar la planilla de notas para seminario.

El personal administrativo de la DCE puede consultar las solicitudes pendientes a través de la interfaz presentada en la figura 4.16. Cuando se ha validado que el estudiante cumple con los requisitos pertinentes para ser calificado, y cuando la planilla ha sido impresa, el personal de la DCE envía un correo de notificación al docente a través de un ícono dispuesto para ello en cada solicitud. De esta forma, el docente es notificado para que proceda a retirar la planilla solicitada.

Consulta de Planillas de notas para Seminario

Cédula o Nombre del Estudiante

Licenciatura: [B](#) | [C](#) | [E](#) | [G](#) | [M](#) | [Q](#) | [Todas](#)

Período Académico: [01-2014](#)

Planillas encontradas: 5 registros en total

Carrera	Asignatura	Estudiante	Resumen	Planilla Notas
C	(6431) SEMINARIO (INTEL.ARTIFIC)	19163211 - FUENTES JORGE	Resumen Planilla	23/07/2014
C	(6411) SEMINARIO (SIST.INFOR)	17401514 - PEREZ JORMAR 17960873 - DIAZ KARINA	Resumen Planilla	21/07/2014 21/07/2014
M	(8603) SEMINARIO II	20413139 - MARTINEZ KERLYNS	Resumen Planilla	19/07/2014
C	(6415) SEMINARIO (TEC. EDUCATIVA)	17554171 - PALACIOS MAYELA	Resumen Planilla	18/07/2014
C	(6427) SEMINARIO (TEC.COM.RED.)	10352539 - PEREIRA GUSTAVO	Resumen Planilla	13/07/2014

Figura 4.16: Consulta de planillas de notas para seminario.

Es importante destacar que las operaciones llevadas a cabo por todos los usuarios de la aplicación son registradas mediante una bitácora, a fin de llevar el control de las mismas y facilitar posibles auditorías al sistema. La bitácora también es útil para almacenar información descriptiva de los errores que ocurren en tiempo de ejecución, de los cuales el usuario final recibe sólo información general.

La siguiente sección describe la fase de Pruebas llevada a cabo para la comprobación de las funcionalidades implementadas durante estas cuatro primeras iteraciones.

4.4.2. Pruebas

En esta fase se describen los resultados de las pruebas realizadas, en general, para las cuatro iteraciones reseñadas anteriormente, puesto que todas las funcionalidades están estrechamente relacionadas para la totalidad de los roles involucrados.

Para el proceso de solicitud de jurado y planilla de notas se ejecutó una serie de pruebas automatizadas concernientes a cada rol, a través de la herramienta Cucumber. La figura 4.17 muestra el caso de prueba junto con los pasos correspondientes y resultados obtenidos para el escenario de realizar una nueva solicitud de jurado por parte de un conjunto de docentes de las diferentes escuelas.


```
Característica: Funcionalidades de las Solicitudes de Jurado para la Secretaria
Antecedentes:
  Dado que el usuario ha iniciado sesion como "Secretaria"

@javascript @no-txn @no-database-cleaner
Escenario: Aprobar Solicitud de Jurado
  Dado que estoy en la pagina "/webapp/solicitud_jurado/sec_consultar"
  Y espero 5 segundos
  Entonces debería haber al menos 1 registro en total
  Cuando hago click en el enlace ambiguo "Aprobar Solicitud"
  Y coloco en el campo "fecha_aprobacion_sesion_ce" el valor "05/04/2014"
  Y coloco en el campo "correlativo" el valor "E0888888"
  Y hago click en el boton "Aceptar y enviar correos"
  Y espero 2 segundos
  Entonces debería ver el mensaje "La solicitud ha sido aprobada y actualizada"

1 escenario (1 passed)
10 steps (10 passed)
0m55.746s
```

Figura 4.18: Escenario de prueba Cucumber - Aprobar solicitudes.

Del conjunto de funcionalidades relativas al personal de la DCE, se ejecutaron varios casos de prueba para comprobar su buen funcionamiento, incluyendo la gestión de solicitudes, la generación correcta de los documentos *PDF*, la modificación efectiva de jurados, el procesamiento de las planillas de notas, entre otras. En la figura 4.19, se muestra el escenario correspondiente a la anulación de solicitudes de jurado.

```
Escenario: Anular Solicitud de Jurado
  Dado que estoy en la pagina "/webapp/solicitud_jurado/admin_consultar"
  Cuando hago click en el enlace ambiguo "Anular Solicitud"
  Y selecciono del checkbox "solicitudes_" la primera opcion
  Y hago click en el boton "Anular"
  Entonces debería ver el mensaje "La solicitud seleccionada ha sido anulada"

1 escenario (1 passed)
8 steps (8 passed)
0m52.314s
```

Figura 4.19: Escenario de prueba Cucumber - Anular solicitudes.

La ejecución de pruebas automatizadas permite comprobar de forma rápida el comportamiento adecuado del código implementado en la fase anterior, puesto que el proceso de solicitud de jurado incluye el uso frecuente de formularios para obtener y validar los datos de entrada necesarios. De esta forma, es posible ejecutar diferentes casos de prueba de forma rápida y automatizada. Sin embargo, inicialmente es necesario invertir una cantidad considerable de tiempo para escribir las pruebas de los diversos escenarios, encontrándose dificultades a la hora de modelar ciertos escenarios que son difíciles de implementar a través de Cucumber. Para estos casos se realizan pruebas manuales tradicionales.

En cuanto a las pruebas de aceptación, estas se basan principalmente en comprobar que el flujo de proceso diseñado es el correcto, que la interfaz de usuario cumple satisfactoriamente los requerimientos, así como verificar la aceptación del formato de los documentos PDF de la solicitud de jurado, carta de aprobación, constancias de tutoría y planilla de notas, todo esto antes de realizar el despliegue de la aplicación. Para ello, se realizaron sucesivas reuniones

con los usuarios de las diferentes escuelas, explicando el funcionamiento del sistema, realizando demostraciones, resolviendo dudas, recogiendo impresiones y tomando en cuenta las sugerencias brindadas. Los comentarios recibidos al realizar estas actividades fueron bastante positivos, y los usuarios coincidieron en opiniones tales como:

- La interfaz de la aplicación es agradable y fácil de usar, las acciones están representadas por íconos intuitivos. No se requiere invertir mucho tiempo para aprender a utilizar la aplicación.
- Como no existía un sistema previo que permitiera automatizar el proceso de las solicitudes de jurado, los usuarios expresaron que la aplicación implementada contribuye a simplificar ampliamente este proceso, así como facilitar la gestión y control de solicitudes.
- Los usuarios manifestaron que con esta automatización se reduce de forma considerable la carga de trabajo al no tener que realizar la transcripción manual de la solicitud y de los cambios decididos por el Consejo de Escuela, sino que sólo deben editar los datos que ya han sido cargados previamente por los docentes, minimizando los posibles errores, aligerando y agilizando el trabajo.
- El hecho de mantener unificada la información de las solicitudes de estudiantes que realizan el seminario o TEG en pareja, para el caso de la Escuela de Computación, permite evitar la redundancia de acciones a la hora de modificar los datos de dichas solicitudes y facilitan a la vez su consulta.
- Los usuarios se expresaron sobre el beneficio de ahorrar una cantidad significativa de papel y materiales de impresión al emplear correos electrónicos para la notificación de la aprobación de solicitudes para todos los actores involucrados, incluyendo al estudiante, de forma que este pueda tener conocimiento de los datos aprobados de su propio trabajo de investigación. Con esto también se ahorra tiempo y esfuerzo al no tener que buscar y entregar las cartas físicamente a todos los miembros del jurado, simplificando la comunicación, mientras que la información siempre está disponible a lo largo del tiempo para su consulta y reenvío de ser necesario.
- Se logra unificar y estandarizar el proceso, así como los documentos derivados, los cuales pueden ser recuperados por los roles con los privilegios apropiados en cualquier punto del proceso.
- Se minimiza la posibilidad de cometer errores y la incongruencia de la información al integrar la información del jurado aprobado de una solicitud para TEG con la planilla de notas, permitiendo realizar los cambios pertinentes de ser necesario.
- La posibilidad de que el docente pueda solicitar las planillas de notas para seminario especificando el jurado correspondiente agiliza el trabajo de la DCE, así como la impresión y entrega de las planillas.

- La DCE puede tener información sobre los próximos graduandos y realizar el chequeo de requisitos con tiempo, así como organizar las actividades relacionadas con el proceso de grado, las cuales deben ser planificadas con antelación.

En la siguiente sección se relata la experiencia vivida durante el despliegue de la aplicación, las dificultades encontradas y los ajustes realizados en dicha fase.

4.4.3. Despliegue

Esta fase consiste en la puesta en producción de la aplicación, permitiendo su uso para los usuarios del sistema CONEST. El primer paso fue recopilar los datos del personal de las escuelas, a fin de registrar sus credenciales y habilitar su acceso.

Durante las primeras semanas del despliegue ocurrieron ciertas dificultades y se presentaron algunas situaciones que no habían sido anticipadas en fases anteriores, tales como excepciones para algunas validaciones que se implementaron de forma estricta, dificultades para agregar jurados externos, lentitud a la hora de mostrar la página de consulta de solicitudes y planillas debido a consultas complejas y a la cantidad de registros en la base de datos de producción, errores en la generación de las planillas de notas, consecuencias no previstas al realizar el cambio del período académico e iniciar uno nuevo, así como al retirar la asignatura involucrada en una solicitud de jurado, entre otros. Esto afectó principalmente al personal de grado de la DCE debido a la cantidad de funcionalidades que requerían utilizar, así como por la urgencia de generar las solicitudes y planillas necesarias antes del cierre del semestre 02-2013.

Para resolver estos inconvenientes se realizaron los ajustes, optimizaciones y correcciones en el menor tiempo posible, respondiendo a las contingencias a medida que surgían y realizando las pruebas necesarias, para luego corroborar el funcionamiento correcto de la aplicación en el sitio de trabajo, directamente con los usuarios. Adicionalmente, con el transcurso del tiempo y debido al proceso de migración de datos de la versión anterior de CONEST, se tuvo la necesidad de realizar tareas adicionales, agregar nuevas funcionalidades y extender otras, tales como:

- Organizar la información de los docentes externos obtenidos a partir de los registros de la base de datos de la versión anterior de CONEST para ser migrados en la nueva base de datos, logrando recuperar la mayor cantidad de datos y evitando el registro manual de cientos de docentes.
- Desarrollar una funcionalidad que permitiera el registro de información sobre los Trabajos Especiales de Grado de semestres anteriores que debían ser cargados en la nueva base de datos para la generación correcta de las constancias de tutoría. Para esto se reutilizó

Capítulo 4. Desarrollo del módulo solicitudes

código de las solicitudes de jurado, específicamente del formulario y de la consulta de solicitudes.

- Debido a que la Escuela de Física tiene un flujo de proceso ligeramente diferente, se restringió la creación de solicitudes sólo para el personal con los cargos autorizados para ello, tales como el jefe de departamento y el director de la escuela.
- Por la razón anterior fue necesario extender las vistas y la permisología a los cargos de mayor autoridad de las escuelas, ya que no habían sido considerados anteriormente.
- Se extendió la funcionalidad de planillas de notas para seminario, reutilizando este código para la generación de planillas de notas para pasantía.

De esta forma, además de corregir las fallas locales a la aplicación, se apoyó el proceso de migración de datos para la versión 3.0 de CONEST, la cual fue bien recibida por parte de la comunidad debido a las mejoras en el diseño de la interfaz de usuario y de las funcionalidades que estaban disponibles en la versión anterior, generando altas expectativas sobre la implementación de los nuevos requerimientos.

Después de varios meses realizando seguimiento al comportamiento del módulo de solicitudes de jurado, se puede decir que se logró la estabilización del mismo al optimizar las consultas y al corregir los casos excepcionales, pero comunes, que pueden darse en el agitado ritmo de la vida académica.

Como resultados de la puesta en producción de este módulo, a finales del semestre 02-2013, se realizaron **58** solicitudes de jurado entre todas las escuelas, se solicitaron **57** planillas de notas para TEG y **81** planillas de notas para seminario, representando estas últimas el mayor volumen de trabajo. La Tabla 4.5 presenta estos datos y su detalle por carrera, mientras que la figura 4.20 consolida esta información.

Tabla 4.5: Cantidad de solicitudes de jurado y planillas de notas procesadas por carrera en el semestre 02-2013.

Carrera	Solicitud de Jurado	Planilla Notas TEG	Planilla Notas Seminario
Biología	16	16	41
Computación	22	21	17
Física	8	8	5
Geoquímica	1	1	0
Matemática	4	4	9
Química	7	7	9
Total	58	57	81

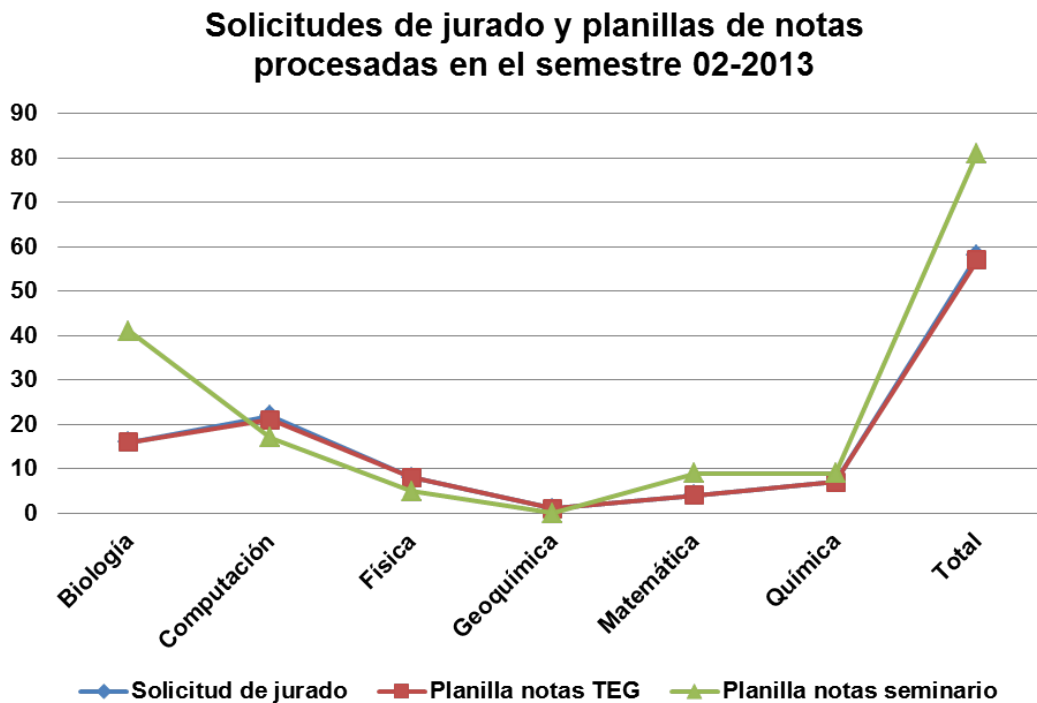


Figura 4.20: Solicitudes de jurado y planillas de notas procesadas en el semestre 02-2013.

Al analizar estos datos se tiene que, en total, la operación más realizada es la solicitud de planilla de notas para seminario, siendo la Escuela de Biología la que presenta mayor número de solicitudes, seguida por la Escuela de Computación. Estas escuelas presentaron el mayor volumen de solicitudes de jurado y planillas de notas para TEG en el semestre 02-2013. Las cantidades de estas últimas dos solicitudes están estrechamente relacionadas, como se puede observar en la figura 4.20, puesto que las tendencias de las solicitudes de jurado y planilla de notas para TEG siguen el mismo patrón. Esto ocurre ya que al solicitar el jurado para la defensa del trabajo de investigación, existen altas probabilidades de que este sea calificado posteriormente, por lo que se solicita la planilla de notas en el mismo semestre de la solicitud. Sin embargo, existen casos donde esto no ocurre, produciéndose el retiro de la asignatura para realizar la presentación del trabajo en el semestre siguiente.

Esta primera experiencia de despliegue sirvió para aprender sobre la importancia de un buen diseño y de mantener el código simple, así como para adquirir conocimientos sobre la puesta en producción de una aplicación web. En consecuencia, las cuatro (4) iteraciones descritas anteriormente sirvieron de base para el diseño y construcción de los requerimientos relacionados con las solicitudes de modificación de inscripción y solicitudes de reincorporación, con el objeto de no cometer los mismos errores, reutilizar el código que ya fue corregido y probado en producción, reducir el tiempo de codificación y entregar resultados eficaces de manera oportuna.

4.5. Iteración 5 - Solicitud de modificación de inscripción

En esta sección se describen las fases de Análisis y Diseño, con un enfoque general a todos los roles, y la fase de Construcción, enfocada hacia el proceso del estudiante, las cuales fueron llevadas a cabo durante esta primera iteración de las solicitudes de modificación de inscripción.

4.5.1. Análisis

La fase de Análisis se fundamenta principalmente en la discusión de los requerimientos más importantes a desarrollar y extender, teniendo como referencia las funcionalidades implementadas en la versión anterior de CONEST, y el proceso de inscripción ya implementado en la nueva versión; así como en la reformulación del proceso de solicitudes estudiantiles, para solventar las deficiencias asociadas al mismo, y que han sido descritas en capítulos anteriores. Para la obtención de la información necesaria, se realizaron tormentas de ideas con los usuarios más experimentados. Los resultados se describen a continuación.

4.5.1.1. Tormenta de ideas

Después de sucesivas reuniones y discusiones con los usuarios para determinar el nuevo flujo de proceso y el comportamiento general que debería tener el sistema, se plantearon alternativas y soluciones a través de tormentas de ideas, a partir de las cuales se formulan los requerimientos funcionales y aspectos más relevantes, listados a continuación:

- Permitir que el estudiante realice sus solicitudes de modificación de inscripción a través del sistema, en el período establecido para ello, proporcionándole retroalimentación sobre el cumplimiento de la normativa de su escuela al momento de seleccionar las asignaturas que son objeto de solicitud. En este punto es necesaria la detección correcta del tipo de solicitud, y se establecen límites estrictos respecto a las asignaturas y cantidad de solicitudes permitidas por tipo.
- Se debe permitir al estudiante realizar solicitudes, aunque no satisfaga los requerimientos académicos establecidos en la normativa, brindándole un espacio para colocar una justificación, ya que la toma de decisiones corresponde a la UAA y al Consejo de Escuela respectivos.
- El estudiante debe imprimir una planilla que describe las solicitudes que ha realizado, las sugerencias emitidas por el sistema en base a la validación de la normativa, el historial de solicitudes estudiantiles realizadas anteriormente, un espacio para el registro de las decisiones que serán recogidas a lo largo del proceso, incluyendo las copias pertinentes, además del *kardex* del estudiante, con el propósito de brindar suficiente información para

la toma de decisiones a los entes correspondientes, y reducir los costos de impresión para el personal administrativo.

- Se debe permitir el reinicio y anulación de solicitudes como procedimientos de contingencia ante las eventualidades. La referida planilla de solicitud debe estar disponible para su consulta o impresión, para todos los roles y en cualquier punto del proceso.
- El procesamiento de las solicitudes debe ser lo suficientemente simple para la División de Control de Estudios, debido al alto número de solicitudes que se generan. Esta funcionalidad debe ser robusta para garantizar la correcta inscripción de las asignaturas solicitadas por los estudiantes, e incluir la notificación a los estudiantes sobre el estado final de sus solicitudes.
- Se debe contar con una herramienta para la administración de reglas y configuración del proceso, modificación de parámetros, activación o desactivación de los mismos, con el fin de garantizar la adaptabilidad del sistema a lo largo del tiempo.
- Ofrecer la funcionalidad de exportar las solicitudes por semestre mediante un archivo de hoja de cálculo *Excel*, para el control interno de las escuelas y de la DCE.

La figura 4.21 sintetiza los aspectos listados anteriormente a través de un mapa mental.

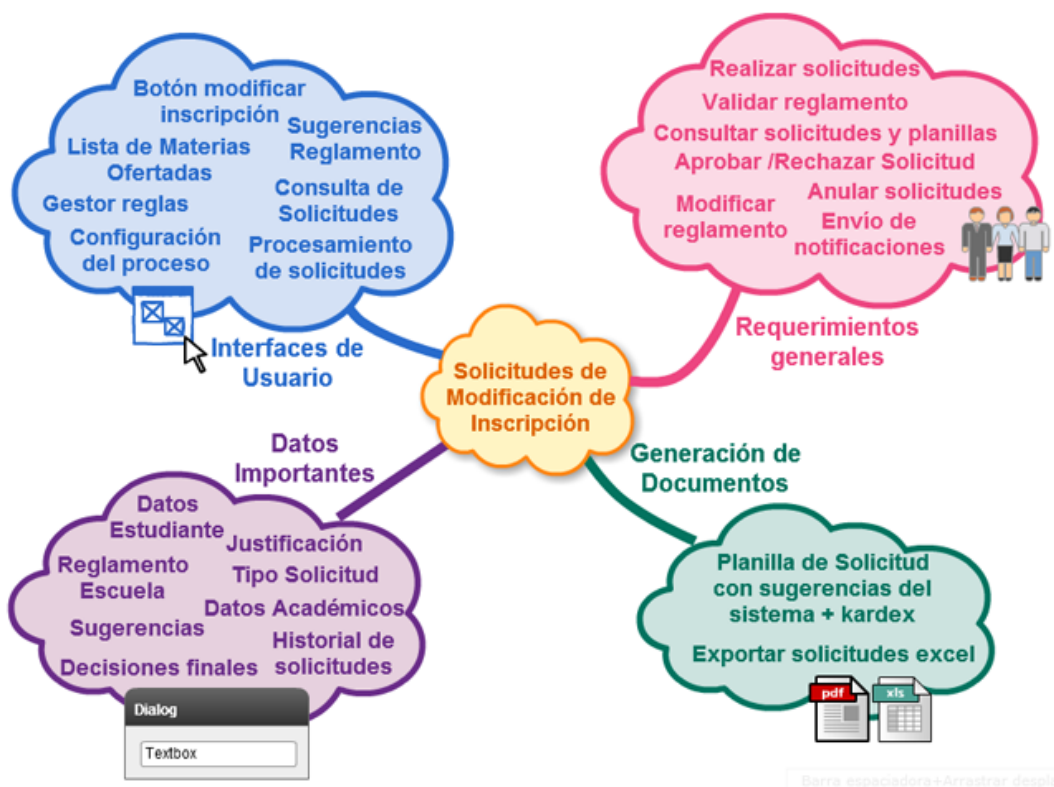


Figura 4.21: Tormenta de ideas para las solicitudes de modificación de inscripción.

4.5.2. Diseño

Esta fase consiste en el modelado del proceso simplificado para la modificación de inscripción, el diseño del gestor de reglas a nivel del modelo de datos para poder brindar la retroalimentación adecuada, tanto al estudiante, como a las unidades de asesoramiento académico; la elaboración del prototipo de interfaz de usuario para el estudiante, que incluye las observaciones emitidas por el sistema, así como el diseño del formato de la planilla de solicitud de modificación de inscripción, el cual es un documento complejo por la cantidad de información que incluye, y que a su vez debe ser recabada a lo largo del proceso.

4.5.2.1. Diseño del flujo de comunicación

Como ya se ha mencionado anteriormente, es necesario realizar cambios al proceso involucrado en la gestión de solicitudes de modificación de inscripción para solventar sus deficiencias, minimizar los costos y agilizar los tiempos de respuesta. Para el diseño del nuevo proceso se realizaron discusiones con las diversas escuelas, negociando los aspectos fundamentales y logrando un consenso general. El resultado fue la simplificación del proceso, llevando la mayoría de las tareas anteriormente automatizadas a una gestión tradicional, pero sencilla y rápida, a fin de automatizar sólo lo necesario, evitar complicar el proceso con demasiados pasos secuenciales, y permitir que el personal nuevo pueda integrarse con facilidad al proceso, sin necesidad de pasar por un entrenamiento exhaustivo en el uso del sistema.

Los cambios más relevantes están relacionados con el registro de acciones en el sistema. Esto ya no es necesario, puesto que en la planilla impresa por el estudiante al inicio del proceso, se recoge toda esta información de forma manual. Este documento circula a través de todos los entes pertinentes, siendo llenado durante el proceso, lo cual permite la abstracción del sistema sobre la organización compleja de algunas escuelas. Las decisiones finales emitidas por el Consejo de Escuela son plasmadas en el mismo documento, el cual se envía a la DCE para su ejecución final, y posterior almacenamiento en el expediente del estudiante. La copia correspondiente reside en la escuela como respaldo.

Tareas como registrar la entrega del aval por parte de los docentes, y registrar las recomendaciones emanadas de cada comisión o unidad de asesoramiento académico ya no son automatizadas por el sistema, puesto que son tan simples que pueden ser realizadas de forma manual con facilidad. Esto permite reducir la cantidad de pasos secuenciales en el proceso, simplificando su implementación en el sistema y facilitando su uso.

La impresión de cientos de lotes de solicitudes por escuela es una acción descartada, ya que el material físico es provisto por el estudiante al inicio del proceso. Cargar previamente las decisiones finales por parte del personal administrativo de las escuelas es otra acción eliminada en el proceso actual. La DCE es la encargada de la ejecución de las decisiones, ya

que sólo puede procesarlas al recibir el documento físico donde se constata esa información. Anteriormente, en caso de discrepancia entre las solicitudes precargadas y los lotes de documentos físicos con las decisiones finales, el proceso podía sufrir un fuerte retraso y múltiples complicaciones. Es por ello que ahora se busca procesar las solicitudes a medida que llegan las decisiones en físico, evitando realizar procedimientos de validación y control entre grandes lotes de documentos y el sistema, lo cual agiliza el proceso y se traduce en respuestas más rápidas para los estudiantes.

En resumen, la automatización de este proceso está dada principalmente por la validación y retroalimentación oportuna sobre el reglamento de las escuelas, la impresión de la planilla de solicitud con toda la información necesaria para soportar el proceso, y el procesamiento final de las solicitudes, incluyendo mecanismos para la gestión del reglamento. Los pasos intermedios han sido simplificados a través de su gestión manual, o descartados en algunos casos, reduciendo la interacción del personal administrativo y de la UAA de las escuelas con el sistema. La figura 4.22 muestra la simplificación del proceso de solicitudes de modificación de inscripción, evidenciando los cambios anteriormente referidos.

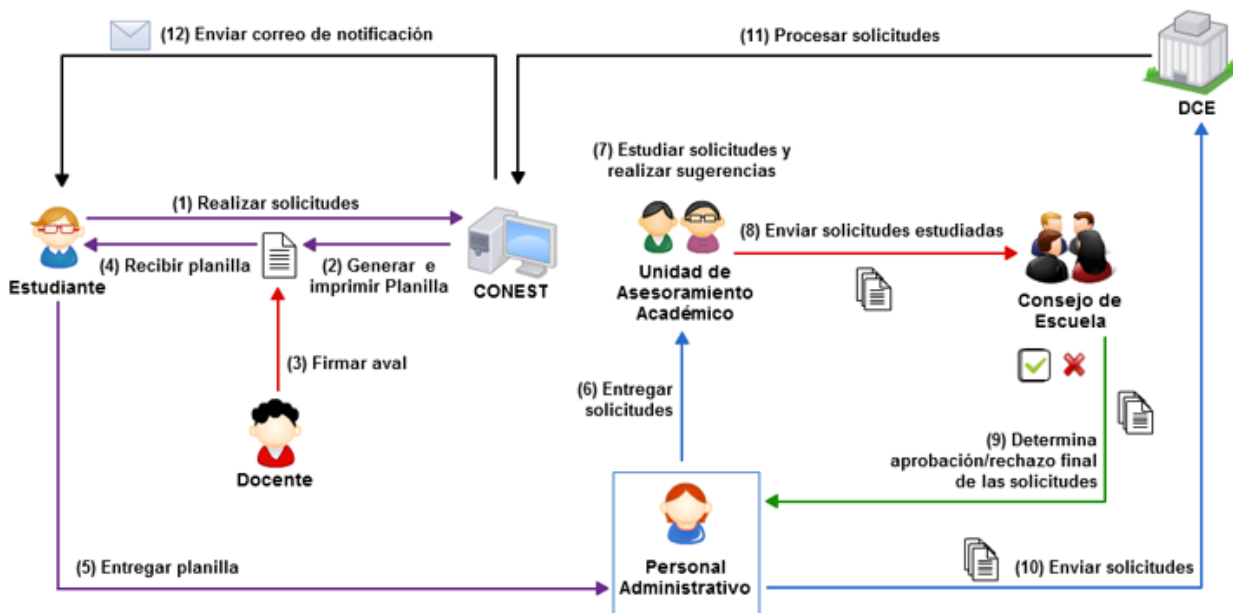


Figura 4.22: Proceso adaptado para las solicitudes de modificación de inscripción.

4.5.2.2. Esquema de base de datos

Para soportar las reglas que constituyen a cada una de las normativas para solicitudes estudiantiles de cada escuela, y permitir su posterior modificación, es necesario crear nuevas tablas en la base de datos. Antes de presentar el esquema de estas tablas, es oportuno explicar el funcionamiento general de las reglas y su integración con el sistema.

En general, una regla de modificación de inscripción está asociada a una *carrera*, aplica a un *tipo de solicitud* (paralelo, exceso de créditos, inscripción extemporánea), y puede derivarse de un *artículo* específico de la normativa, o ser una regla independiente. Las reglas están compuestas y son ejecutadas a través de un conjunto de *condiciones*, que en realidad son métodos implementados en las clases del modelo de la aplicación. Estos métodos son llamados y se ejecutan al seleccionar cada asignatura, para su validación y retroalimentación. Los métodos reciben parámetros, los cuales se almacenan en la base de datos como *valores*. Son éstos los que están sujetos a modificaciones con el paso del tiempo.

Cada regla posee un *estado* (activa, inactiva), un *orden* que especifica su secuencia de ejecución respecto a las demás reglas, y un *tipo de ejecución* específico: parámetro, obligatoria, u opcional. Las reglas de tipo parámetro no se ejecutan al momento, puesto que no son métodos, tan sólo contienen valores que son consumidos al iniciar la ejecución del proceso. En cambio, las reglas de ejecución obligatoria sí están implementadas como métodos y deben ser evaluadas en cada acción, retornando como retroalimentación la descripción de las reglas que han fallado. El último tipo, las reglas de ejecución opcional, se diseñaron para modificar el comportamiento anterior y dar cabida a otras alternativas.

La retroalimentación y sugerencias obtenidas después de la ejecución de un conjunto de reglas, es almacenada en una tabla especial para la posterior recuperación de la información en cualquier punto del proceso. El comportamiento anteriormente descrito es posible mediante la estructura documentada en la figura 4.23.

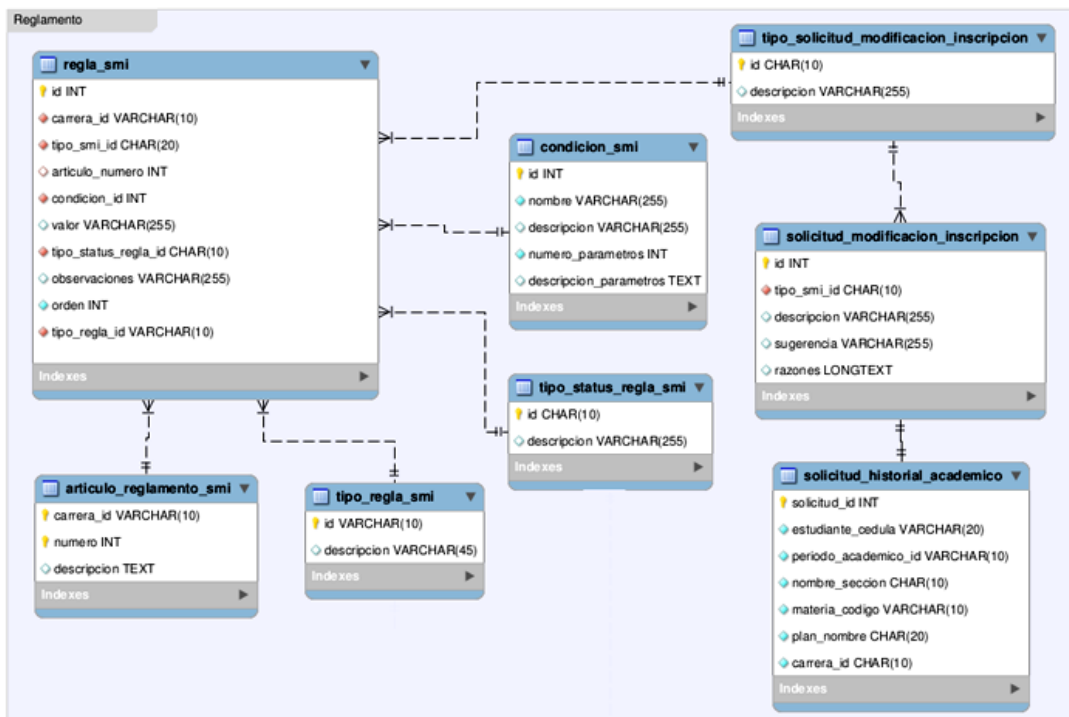


Figura 4.23: Esquema de base de datos para el gestor de reglas de las solicitudes de modificación de inscripción.

4.5.2.3. Prototipos de interfaz

La interfaz de usuario más importante a diseñar en esta iteración es la que permite al estudiante realizar sus solicitudes de modificación de inscripción. Al tener el proceso de inscripción ya implementado, con una interfaz bien definida y altamente interactiva, la idea es reutilizar este diseño y extenderlo para incluir la información asociada a las solicitudes, aprovechando la experiencia previa de los estudiantes, obtenida al momento de realizar el proceso de inscripción a través de la nueva versión de CONEST.

La prioridad de diseño es encontrar la forma de mostrar al estudiante las observaciones emitidas por el gestor de reglas, especificar si cumple los requisitos, o si por el contrario, se producen objeciones en algunas reglas de la normativa. Estas objeciones deben detallar el enunciado del artículo que falla, o la explicación general de la regla. Al contar con un espacio limitado por la cantidad de secciones en la interfaz, se decidió ordenar la información en una tabla ubicada en la parte superior de la misma, aprovechando todo el espacio horizontal disponible. El prototipo hecho en papel, mostrado en la figura 4.24 ilustra someramente la idea anteriormente planteada, comparando y reorganizando la interfaz del proceso de inscripción.

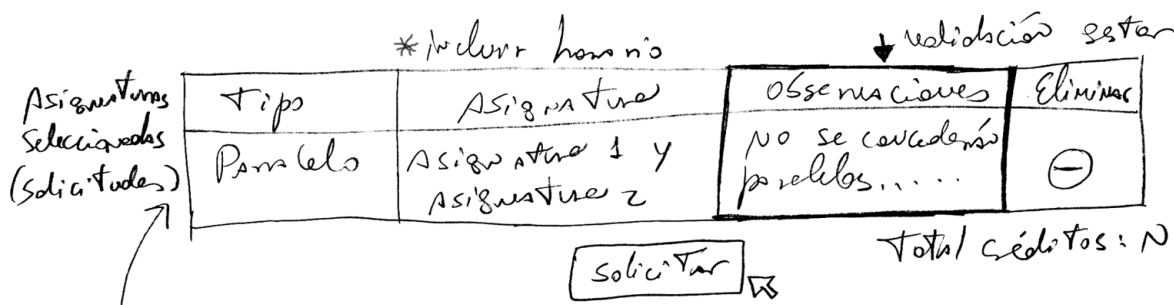




Figura 4.24: Prototipo de interfaz del estudiante para realizar solicitudes de modificación de inscripción.

Capítulo 4. Desarrollo del módulo solicitudes

Otro elemento fundamental del proceso es la planilla de solicitud, la cual es un documento completo que debe recoger información sobre las solicitudes y proporcionar sugerencias para facilitar la toma de decisiones de las UAA, en base a un análisis automatizado del reglamento de cada escuela. A su vez, esta planilla debe contener la justificación del estudiante, el historial de solicitudes que ha realizado, el espacio determinado para recoger las recomendaciones y decisiones finales, incluyendo el *kardex* del estudiante, entre otra información. Uno de los primeros prototipos funcionales elaborados para esta planilla se muestra con algunas correcciones por realizar, en la figura 4.25.



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
DIVISIÓN DE CONTROL DE ESTUDIOS



Fecha: 24/04/14
Periodo Lectivo: 02-2013

SOLICITUD DE MODIFICACIÓN DE INSCRIPCIÓN

ESTUDIANTE: CI 20957778, Br. MANCINI OCAMPO ADRIANO ✓		LICENCIATURA: COMPUTACION ✓	
TELÉFONOS: 0555-5555555 / 0555-5555555		CORREO: uncorreo@servidor.com ✗	

teomde

SOLICITUD			
PARALELO: (6002) ORG.Y ESTRUCT.DEL COMP.II y (6004) LENGUAJES DE PROGRAMACIÓN			
SUGERENCIA: SE SUGIERE SU RECHAZO. RAZONES: NO SE CONCEDERÁN PARALELOS DE MATERIAS QUE NO HAYAN SIDO CURSADAS COMPLETAS EL SEMESTRE INMEDIATO ANTERIOR. LAS MATERIAS RETIRADAS NO SERÁN CONSIDERADAS PARA SOLICITUDES DE PARALELOS. <i>+ S. m. de</i>			
ESPACIO RESERVADO PARA LA COMISIÓN DE ESCUELA			
APROBAR <input type="checkbox"/>		RECHAZAR <input type="checkbox"/>	
JUSTIFICACIÓN: <i>(observación) + S. m. de</i>			
ESPACIO RESERVADO PARA EL CONSEJO DE ESCUELA			
APROBAR <input type="checkbox"/>		RECHAZAR <input type="checkbox"/>	

Simple favor

SOLICITUD			
INSCRIPCIÓN EXTEMPORÁNEA Y EXTRACRÉDITO: (0231) - APRECIACION MUSICAL I			
SUGERENCIA: SE SUGIERE SU APROBACIÓN. RAZONES: CUMPLE CON LOS REQUISITOS.			
AVAL DEL DOCENTE			
CI:	NOMBRE:	FIRMA:	
ESPACIO RESERVADO PARA LA COMISIÓN DE ESCUELA			
APROBAR <input type="checkbox"/>		RECHAZAR <input type="checkbox"/>	
JUSTIFICACIÓN:			
ESPACIO RESERVADO PARA EL CONSEJO DE ESCUELA			
APROBAR <input type="checkbox"/>		RECHAZAR <input type="checkbox"/>	

JUSTIFICACIÓN DEL ESTUDIANTE

DEBIDO A PROBLEMAS DE SALUD, SOBRE LOS CUALES TENGO CONSTANCIA MÉDICA, ME VI OBLIGADO A RETIRAR LAS MATERIAS DEL SEMESTRE ANTERIOR, POR LO QUE DESEO PONERME AL DÍA Y ADELANTAR MATERIAS INSCRIBIENDO LAS SIGUIENTES: SISTEMAS OPERATIVOS Y APRECIACIÓN MUSICAL I.

DATOS ACADÉMICOS

PERÍODO INGRESO: 02-2011			TIPO INGRESO: PRUEBA INTERNA	
ULTIMO PERIODO ACADÉMICO CURSADO: 02-2013			REGLAMENTO VIGENTE: --	
INSCRIPCIÓN DEL PERIODO 01-2013			INSCRIPCIÓN DEL PERIODO ACTUAL 02-2013	
ASIGNATURA	NOTA	CRÉDITOS	ASIGNATURA	CRÉDITOS
(6002) ORG.Y ESTRUCT.DEL COMP.II	RET	5	(6002) ORG.Y ESTRUCT.DEL COMP.II	5
(6203) INGENIERIA DE SOFTWARE	00	5	(6203) INGENIERIA DE SOFTWARE	5
(8207) MATEMATICA II	RET	6	(8207) MATEMATICA II	6
TOTAL CRÉDITOS		16	TOTAL CRÉDITOS	16

2 solicitudes por página

Figura 4.25: Prototipo de planilla de solicitud de modificación de inscripción.

En la fase de Construcción, descrita a continuación, se muestra la implementación de las ideas planteadas durante esta fase.

4.5.3. Construcción - Perspectiva del estudiante

El reto más desafiante en todo el proceso es la implementación del gestor de reglas. En esta iteración, el enfoque se basa sólo en la incorporación y ejecución de las reglas de la normativa de cada escuela, así como en la presentación de la retroalimentación derivada, sea en la interfaz o en la planilla de solicitud. Es decir, el objetivo es la implementación interna de las reglas y su integración con el proceso de modificación de inscripción, reservando su manejo administrativo para la siguiente iteración.

Como se mencionó en la fase de Diseño, la idea es reutilizar el código y las interfaces de usuario del proceso de inscripción. Durante el período establecido para realizar solicitudes de modificación de inscripción, cuya duración está entre una y dos semanas, el estudiante visualiza un botón a través del cual se ejecutan un conjunto de validaciones previas para preparar el ambiente de ejecución. En este paso, se almacena en la sesión del usuario una amplia cantidad de parámetros que permiten el funcionamiento eficiente del proceso. Posteriormente, se presenta la oferta de asignaturas disponibles para realizar solicitudes, basado en un análisis automatizado del historial académico del estudiante. Todo este comportamiento se hereda del proceso de inscripción. La figura 4.26 presenta la interfaz inicial del estudiante en este punto del proceso.

Modificación de Inscripción (01-2014)

Modificar Inscripción

Modificación de Inscripción 01-2014

Solicitudes a realizar

Paso 2: Verifique su carga académica y posibles observaciones

Solicitud	Tipo	Asignatura	UC	Observaciones	Eliminar
Total de Créditos: 20					

Justificación (opcional)

Se recomienda agregar una justificación breve y concisa en caso de no satisfacer los requisitos para realizar las solicitudes.

Paso 3: Realice sus solicitudes

Solicitar

Asignaturas disponibles

Paso 1: Agregue las materias que desee inscribir

Tipo	Asignatura	UC	Agregar
O	(6204) LENGUAJES DE PROGRAMACION	5	<input type="button" value="➕"/>
C	(0206) ACTIVIDAD CORAL	3	<input type="button" value="➕"/>
C	(0007) DEPORTES	2	<input type="button" value="➕"/>

Asignaturas inscritas actualmente

Tipo	Asignatura	UC
O	(6104) PROB. Y ESTADISTICA	5
O	(6108) MATEMATICAS DISCRETAS III	4
P	(6403) PASANTIAS	5
C	(8601) INTROD. A LA COMPUTACION	6

Figura 4.26: Interfaz inicial del estudiante para solicitudes de modificación de inscripción.

Cuando el estudiante selecciona una asignatura de la oferta disponible, con la intención de solicitar su inscripción, el sistema ejecuta la detección del tipo de solicitud (paralelo, exceso de créditos o inscripción extemporánea) y su validación a través del gestor de reglas, el cual no es más que un método desarrollado en una clase del modelo que contiene los mecanismos para llamar al resto de los métodos que implementan la normativa de cada escuela. Este método recupera de la base de datos los valores que deben cumplirse, y los envía a los métodos para su ejecución. El anexo J, en la página 121, presenta un extracto de código en *Ruby* para la detección del tipo de solicitud, mientras que el anexo K, ubicado en la página 122, muestra un extracto del código implementado en el gestor de reglas. La figura 4.27 muestra una lista de los métodos principales que implementan a las diversas normativas.

```
# Metodos que implementan las reglas de c/u Escuela

def numero_materias_con_estatus_en_n_periodos_anteriores?(
  operador, nro_materias, tipo_status_materia_id, nro_periodos_anteriores, incluir_verano)
end

def contar_materias_con_estatus_en_n_periodos_anteriores?(
  operador, nro_materias, tipo_status_materia_id, nro_periodos_anteriores, incluir_verano)
end

def materia_con_estatus_en_periodo_anterior?(tipo_status_materia_id)
end

def cumple_art_156?
end

def paralelo_no_concedido_anteriormente?
end

def materia_con_estatus_en_paralelo_anterior?(tipo_status_materia_id)
end

def paralelo_tipo_materia_requiere_aval?(tipo_materia_requisito_id, tipo_materia_paralelo_id)
end

def paralelo_verificar_cantidad_tipo_materia?(arreglo)
end

def eficiencia_estudiante(operador, eficiencia)
end
```

Figura 4.27: Métodos del gestor de reglas de modificación de inscripción.

Cabe destacar que la traducción de las normativas, las cuales están escritas en lenguaje natural, hacia un lenguaje de programación para su automatización, es compleja, pero la ventaja es que son bastante similares entre sí, difiriendo sólo en algunos parámetros, por lo que al lograr su generalización e implementación es posible modelar las reglas más relevantes de todas las escuelas, usando aproximadamente diez (10) métodos para validar las cinco (5) normativas. Al permitir la modificación de los parámetros que éstos reciben, se obtiene un comportamiento mucho más flexible para cubrir una amplia variedad de casos, y para adaptarse a los cambios que puedan surgir. Además, siempre es posible añadir más métodos, con sus respectivos parámetros, para modelar nuevas reglas a futuro. El fragmento de código presentado

Capítulo 4. Desarrollo del módulo solicitudes

en la figura 4.28 muestra la implementación de una regla muy común en todas las normativas, relacionada con la verificación del estado de una asignatura en el semestre anterior.

```
def materia_con_estatus_en_periodo_anterior?(tipo_status_materia_id)
  periodo_academico_anterior = obtener_periodo_academico_anterior
  materias_requisito = obtener_materias_requisito(@materia_codigo)
  materias_requisito.each do |materia_codigo_requisito|

    historial_academico = HistorialAcademico.where(
      estudiante_cedula: @estudiante_cedula,
      carrera_id: @carrera_id,
      periodo_academico_id: periodo_academico_anterior.id,
      materia_codigo: materia_codigo_requisito).take

    return false unless historial_academico
    return historial_academico.tipo_status_materia_id == tipo_status_materia_id
  end
end
```


Figura 4.28: Método que permite la validación de una regla general de la normativa de solicitudes estudiantiles.

La cadena de acciones referida anteriormente se produce en cada intento de agregar asignaturas. En resumen, lo que se realiza es una ejecución indirecta de una cadena de métodos, y a medida que estos fallan se almacenan las objeciones derivadas de los artículos o descripción de las reglas, para ser mostradas al estudiante. La figura 4.29 muestra la descripción de un conjunto de solicitudes realizadas por un estudiante, junto con las observaciones generadas por el sistema en base a la ejecución de la normativa. También se observa el espacio para que el estudiante agregue una exposición de motivos al momento de realizar sus solicitudes. Aquí se evidencian los cambios introducidos en base al prototipo mostrado en la fase de Diseño.

Modificación de Inscripción 01-2014

Solicitudes a realizar

Paso 2: Verifique su carga académica y posibles observaciones

Solicitud	Tipo	Asignatura	UC	Observaciones	Eliminar
EXTRACRÉDITO, PARALELO	O	(6204) LENGUAJES DE PROGRAMACION y (6108) MATEMATICAS DISCRETAS III	5	EL ESTUDIANTE QUE SOLICITE EXCESO DE CRÉDITOS DEBE HABER APROBADO AL MENOS 6 (SEIS) ASIGNATURAS EN LOS DOS SEMESTRES ANTERIORES AL DE LA SOLICITUD. SE CONSIDERARÁ UNA EFICIENCIA MAYOR O IGUAL A 0.8 PARA EL ORTOGAMIENTO DE PARALELOS.	

Total de Creditos: 25

Justificación (opcional)

Esta es la exposición de motivos proporcionada por el estudiante.


Paso 3: Realice sus solicitudes

[Solicitar](#)

Figura 4.29: Observaciones proporcionadas por el gestor de reglas de modificación de inscripción.

Capítulo 4. Desarrollo del módulo solicitudes

Después de confirmar su voluntad de realizar las solicitudes, el estudiante puede consultar un resumen, la organización de su posible nuevo horario, y descargar la planilla que deberá ser entregada en la escuela respectiva para su procesamiento. La figura 4.30 muestra esta interfaz.



The screenshot shows a web interface titled "Solicitudes Estudiantiles". At the top, there is a yellow notification box with an information icon and the text: "Sus solicitudes de modificación de inscripción han sido procesadas exitosamente. Próximamente será notificado vía correo electrónico sobre la aprobación o rechazo de las mismas. Recuerde imprimir la Planilla de Modificación de Inscripción y el Kardex para entregarlos en la Escuela correspondiente." Below this, a section titled "Solicitudes" contains a table with the following data:

Solicitud	Asignatura	UC	Observaciones
EXTRACRÉDITO, PARALELO	(6204) LENGUAJES DE PROGRAMACION y (6108) MATEMATICAS DISCRETAS III	5	EL ESTUDIANTE QUE SOLICITE EXCESO DE CRÉDITOS DEBE HABER APROBADO AL MENOS 6 (SEIS) ASIGNATURAS EN LOS DOS SEMESTRES ANTERIORES AL DE LA SOLICITUD. SE CONSIDERARÁ UNA EFICIENCIA MAYOR O IGUAL A 0.8 PARA EL ORTOGAMIENTO DE PARALELOS.

Below the table, it says "Total de Créditos: 25". At the bottom, there is a button with a printer icon and the text "Imprimir Planilla de Modificación Inscripción y Kardex".

Figura 4.30: Interfaz del estudiante para descargar la planilla de modificación de inscripción.

El formato final de la planilla de solicitud fue bastante simplificado debido a que mucha de la información que contenía en un principio está incluida en el *kardex* del estudiante, el cual se genera de forma conjunta con la planilla. Estos y los cambios realizados en base a la retroalimentación de los usuarios pueden observarse en el anexo L, ubicado en la página 123.

4.6. Iteración 6 - Solicitud de modificación de inscripción

En esta iteración se implementan las funcionalidades relacionadas con la consulta y procesamiento de solicitudes para el personal administrativo de las escuelas y para el personal de la DCE. Posteriormente, se realiza la implementación del gestor de reglas de modificación de inscripción, así como la herramienta de configuración inicial para este proceso.

4.6.1. Diseño

Esta fase está enfocada principalmente en el diseño de las interfaces para el procesamiento de las solicitudes de modificación de inscripción por parte del personal administrativo de la DCE, así como en el diseño de la interfaz del gestor de reglas.

Para la interfaz de consulta y procesamiento de solicitudes, se usa un estilo similar al de las solicitudes de jurado, específicamente para la agrupación de funcionalidades y uso de ventanas modales para la consulta de información, a fin de aprovechar el conocimiento previo. La DCE

Capítulo 4. Desarrollo del módulo solicitudes

es quien posee los mayores privilegios, de los cuales un subconjunto es heredado para el personal administrativo de las escuelas. El conjunto de funcionalidades correspondientes a ambos roles es bastante reducido, por lo cual el diseño resulta muy simple. El prototipo de la interfaz correspondiente a la DCE se muestra en la figura 4.31.

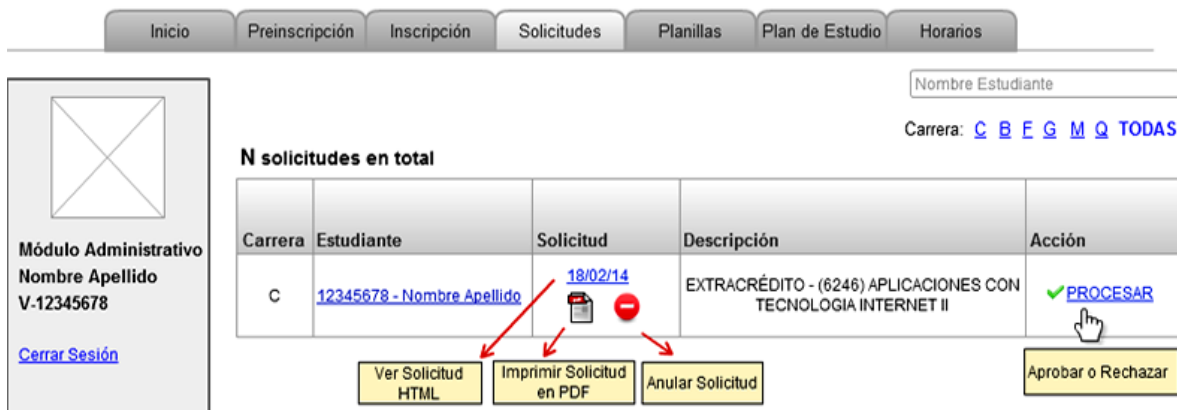


Figura 4.31: Prototipo de interfaz de consulta de solicitudes de modificación de inscripción para la DCE.

En cuanto al diseño del gestor de reglas, el objetivo es poder brindar una visión integral y sencilla del estado de las reglas así como de sus valores actuales, agrupándolas por escuela, para facilitar su consulta y edición por parte del personal de la DCE. La figura 4.32 muestra un prototipo hecho a lápiz y papel para plasmar rápidamente la información que debería proporcionar la interfaz del gestor.

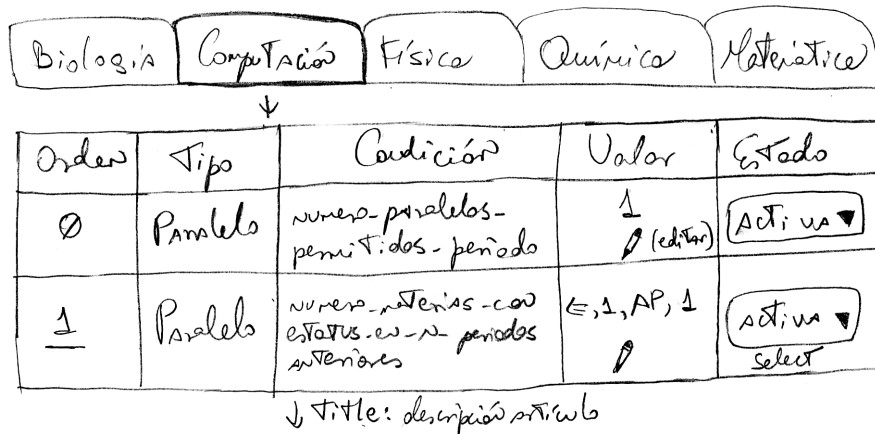


Figura 4.32: Prototipo de interfaz en papel para el gestor de reglas.

Los valores asociados a las reglas pueden ser difíciles de comprender a simple vista, ya que algunas reglas son bastante complejas. Esto puede causar complicaciones a la hora de modificar estos parámetros si no se tienen claros los efectos que se producirán. Por esta razón, se ha diseñado un mecanismo para validar los parámetros de acuerdo a sus características, y brindar mayor información sobre el significado de los mismos al momento de su edición. Esto

es posible mediante la descripción de cada valor asociado a una regla, a través de un objeto *JSON* que se almacena en la tabla correspondiente a las condiciones que implementan las normativas de las escuelas. La estructura general de este objeto se muestra, en la figura 4.33, mientras que la figura 4.34 presenta un ejemplo real aplicado al sistema.

```
1 ▾ [ #Todos los objetos JSON son tratados como arreglos
2 ▾ { #Un hash para cada valor
3     "tipo": <tipo de dato del valor>,
4     "etiqueta": <texto del campo>,
5     "min": <valor minimo para validaciones>,
6     "max": <valor maximo para validaciones>
7 }
8 ]
```

Figura 4.33: Estructura del objeto *JSON* para la descripción de valores de las reglas de solicitudes estudiantiles.

```
1 ▾ [
2 ▾ {
3     "tipo": "operador",
4     "etiqueta": "Operador de Comparación"
5 },
6 ▾ {
7     "tipo": "entero",
8     "etiqueta": "Número de materias por período",
9     "min": 0,
10    "max": 5
11 },
12 ▾ {
13     "tipo": "tipo_status_materia_id",
14     "etiqueta": "Estado de la asignatura"
15 },
16 ▾ {
17     "tipo": "entero",
18     "etiqueta": "Número de períodos anteriores",
19     "min": 1,
20     "max": 5
21 },
22 ▾ {
23     "tipo": "logico",
24     "etiqueta": "Incluir Verano"
25 }
26 ]
```

Figura 4.34: Ejemplo de objeto *JSON* para la descripción de valores de una regla.

4.6.2. Construcción - Perspectiva de la DCE

En esta fase de Construcción se priorizan las funcionalidades relativas al personal de la DCE para permitir el procesamiento oportuno de las solicitudes de modificación de inscripción. La implementación de estas funcionalidades se agiliza mediante la reutilización de código utilizado y probado en las solicitudes de jurado, tanto para las interfaces gráficas, como para la gestión de las solicitudes. La figura 4.35 muestra la interfaz de consulta de solicitudes para el personal de la DCE, donde resalta el ícono para exportar las solicitudes por semestre de todas las escuelas, en formato de hoja de cálculo *Excel*, así como el conjunto de acciones disponibles para cada estudiante que ha realizado solicitudes estudiantiles.

Nro	Carrera	Estudiante	Descripción	Solicitud	Estado
1	G	23640330 - CHACÓN GENESIS	EXTRACRÉDITO, INSCRIPCIÓN EXTEMPORÁNEA - (5441) COMUNICACION TECNICA	05/07/2014 	23/07/2014
2	G	19228000 - SALAZAR JAMIER	INSCRIPCIÓN EXTEMPORÁNEA - (5047) GEO DE EXPLOR Y PROD HIDRO INSCRIPCIÓN EXTEMPORÁNEA - (5042) GEOFISICA	30/06/2014 	23/07/2014
3	C	19044222 - LOPEZ CLELIA	EXTRACRÉDITO, PARALELO - (6204) LENGUAJES DE PROGRAMACION y (6108) MATEMATICAS DISCRETAS III	30/06/2014 	PROCESAR
4	C	19354766 - MACHADO LUIS	INSCRIPCIÓN EXTEMPORÁNEA - (6402) TRABAJO ESPECIAL DE GRADO	30/06/2014 	07/07/2014
5	C	19737013 - GUEVARA JORGE	INSCRIPCIÓN EXTEMPORÁNEA - (0206) ACTIVIDAD CORAL	30/06/2014 	PROCESAR

Figura 4.35: Interfaz de consulta de solicitudes de modificación de inscripción para la DCE.

Para el personal administrativo de las escuelas, la interfaz es similar, pero mucho más sencilla, ya que las funciones de anulación, reinicio y procesamiento de solicitudes están reservadas para el personal de la DCE. En su lugar, el personal de las escuelas, incluyendo a las UAA, pueden consultar los detalles de las solicitudes, y exportar las mismas en formato *Excel* para su control interno.

El procesamiento de las solicitudes, llevado a cabo por la DCE, se realiza mediante una ventana modal, presentada en la figura 4.36, donde se despliega una tabla que contiene todas las solicitudes de un estudiante, junto con las opciones de aprobación o rechazo que deben ser seleccionadas de acuerdo a las decisiones emitidas por el Consejo de Escuela respectivo.

Capítulo 4. Desarrollo del módulo solicitudes

El resultado del procedimiento anterior es informado detalladamente al usuario mediante un resumen de las acciones realizadas por el sistema.

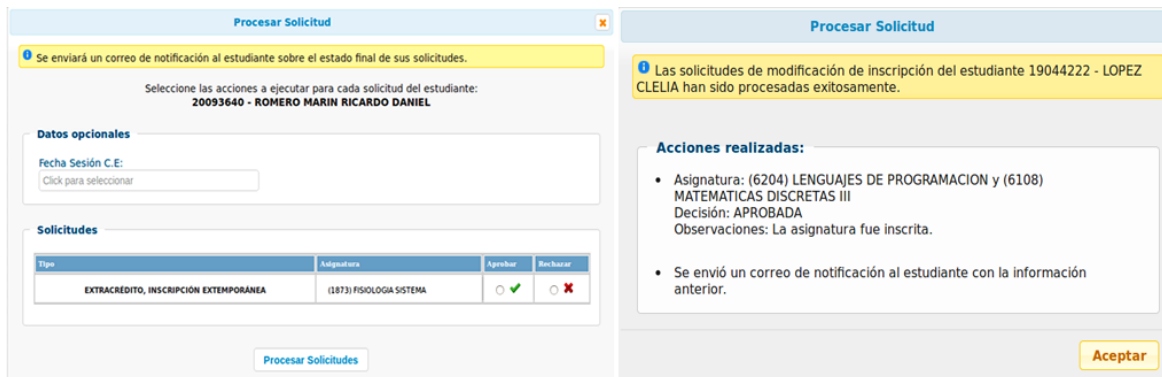


Figura 4.36: Ventana modal para la aprobación o rechazo de solicitudes de modificación de inscripción.

De esta forma, el estado de las solicitudes es actualizado, se realizan las inscripciones de las asignaturas pertinentes en caso de que hayan sido aprobadas, y además, se le notifica al estudiante vía correo electrónico de las acciones realizadas, así como del estado final de sus solicitudes.

Los últimos requerimientos implementados en esta iteración están relacionados con el gestor de reglas de solicitudes de modificación de inscripción. La interfaz de esta herramienta, que abarca a todas las escuelas, se muestra en la figura 4.37.

Reglas Solicitudes Estudiantiles						
BIOLOGIA COMPUTACION FISICA GEOQUIMICA MATEMATICA QUIMICA						
Reglas encontradas: 18						
Orden	Tipo Solicitud	Artículo	Condición	Valor	Tipo Regla	Estado
0	INSCRIPCIÓN EXTEMPORÁNEA	No tiene un artículo asociado.	OFERTAR ASIGNATURAS INGLES I Y II	NO	PARAMETRO	ACTIVA
0	INSCRIPCIÓN EXTEMPORÁNEA	No tiene un artículo asociado.	OFERTAR ASIGNATURAS PCI	NO	PARAMETRO	ACTIVA
0	INSCRIPCIÓN EXTEMPORÁNEA	No tiene un artículo asociado.	OFERTAR ASIGNATURAS COMPLEMENTARIAS	SI	PARAMETRO	ACTIVA
1	INSCRIPCIÓN EXTEMPORÁNEA	2	PARALELO TIPO MATERIA REQUIERE AVAL?	S,T	OBLIGATORIA	ACTIVA
0	EXTRACRÉDITO	9	NUMERO EXTRACREDITOS PERMITIDOS POR PERIODO	1	PARAMETRO	ACTIVA
1	EXTRACRÉDITO	5	NUMERO MATERIAS CON ESTATUS EN N PERIODOS ANTERIORES?	<=,1,AP,1,false	OBLIGATORIA	ACTIVA

Figura 4.37: Interfaz del gestor de reglas de modificación de inscripción.

Como se mencionó en la fase de Diseño, se ha prestado especial atención al momento de editar los valores de las reglas existentes, usando una estructura de datos en formato *JSON* para validar y describir estos parámetros. Se implementa un mecanismo que, dependiendo del tipo de dato de cada valor, genera el código *HTML* necesario para crear los campos de cada parámetro, a fin de facilitar el ingreso de los datos al usuario. De esta forma, es posible generar campos numéricos o de texto, listas desplegables con un conjunto de valores predeterminados, entre otros. La figura 4.38 presenta el resultado del funcionamiento de este mecanismo.

Editar Valores

Editar Valores

Todos los valores son requeridos. *

Operador de Comparación
Mayor o igual que

Número de materias por período
2

Estado de la asignatura
APROBADA

Número de períodos anteriores
2

Incluir Verano
SI

Guardar

Figura 4.38: Editar valores asociados a una regla de modificación de inscripción.

Por último, se ha desarrollado un pequeño gestor de configuración inicial para este proceso, con el objeto de proporcionar una visión general de la oferta académica disponible para asignaturas de tipo Complementaria, PCI y de Idiomas, a fin de realizar los ajustes de manera oportuna, antes de abrir el período de solicitudes estudiantiles.

Esta herramienta también brinda información básica sobre los valores actuales de las reglas más generales, y está enlazada con el gestor de reglas para facilitar los ajustes. De igual forma, el gestor permite abrir o cerrar el proceso de solicitudes estudiantiles en el sistema, entre otras opciones. La figura 4.39 muestra algunas capturas de la herramienta.

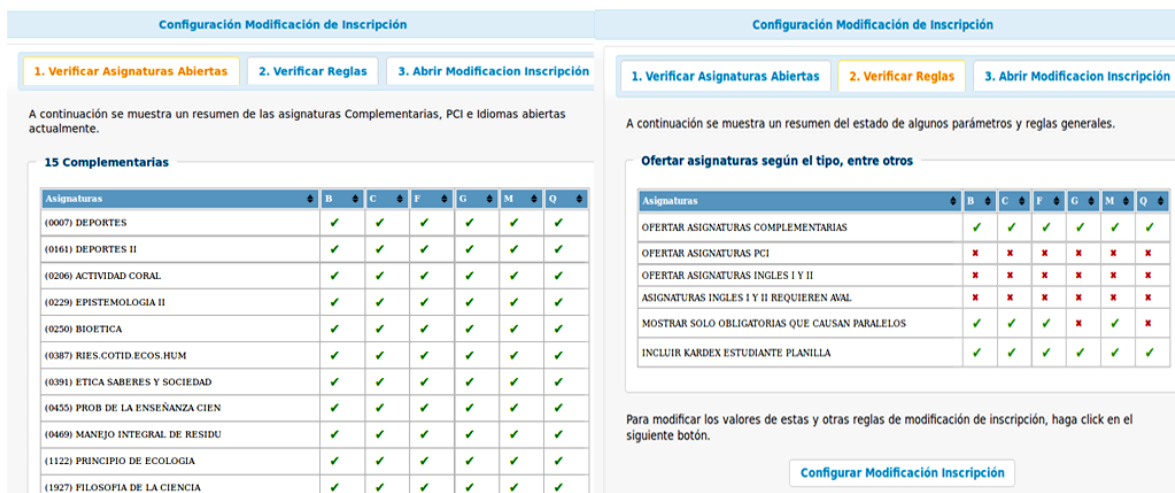


Figura 4.39: Gestor de configuración para el proceso de modificación de inscripción.

A continuación, se describen las experiencias obtenidas sobre este proceso, en las fases de Pruebas y Despliegue.

4.6.3. Pruebas

Para las solicitudes de modificación de inscripción, se realizaron pruebas de aceptación a través de una reunión con los usuarios, principalmente miembros de las UAA de las seis carreras y de la DCE. No se realizaron pruebas automatizadas para el rol del estudiante, ya que se requiere una alta interacción humana para probar los múltiples casos y validaciones relacionadas con las diferentes solicitudes, razón por la cual las pruebas se ejecutaron de forma manual, sin mencionar las restricciones de tiempo antes del despliegue de la aplicación. La sencillez del resto de funcionalidades hizo que no fuera necesario la ejecución de pruebas automatizadas para los requerimientos del rol administrativo.

Como resultados de las demostraciones y reuniones realizadas durante las pruebas de aceptación, se mencionan los siguientes:

- El personal administrativo de las escuelas y de la DCE estuvo, en su mayoría, de acuerdo con el rediseño del proceso para las solicitudes de modificación de inscripción, y coincidieron en que el resultado más notable es la simplificación y el ahorro de papel, ya que el estudiante proporciona los soportes necesarios para recoger la información a lo largo del proceso.
- Hubo opiniones encontradas en cuanto a la gestión manual de las solicitudes. Algunos usuarios consideraron, a su juicio personal que "se sufre un retroceso al realizar el proceso de forma manual cuando antes estaba automatizado, y además se genera un gasto mayor de materiales (papel) al estudiante". Aunque se puede estar de acuerdo con esta

opinión, se debe recordar que el estudiante es la persona que inicia este proceso y resulta beneficiada con el mismo. Tomando en cuenta el volumen de estudiantes que realizan numerosas solicitudes cada semestre, y de los costos ampliamente discutidos en capítulos anteriores, lo que se busca es que el estudiante contribuya con la gestión de sus solicitudes proporcionando la mínima cantidad posible de requisitos.

- Respecto a la gestión manual de las recomendaciones y decisiones, que antes eran cargadas en el sistema, es relevante destacar que la acción de emitir una sugerencia de aprobación o rechazo, y justificarla a través del sistema para cada estudiante, puede incluso tomar más tiempo que simplemente marcar la recomendación en la planilla, y escribir una justificación corta, la cual la mayoría de las veces no se coloca, puesto que coincide con la sugerencia y el razonamiento proporcionado por el sistema, el cual se basa a su vez en el reglamento. Si se toma en cuenta a los usuarios nuevos o inexpertos en el sistema, la gestión manual también puede resultar ventajosa, al no requerir un entrenamiento previo.
- Aunque los puntos anteriores pueden ser discutibles, la mayoría de las escuelas estuvo conforme con la gestión manual de las solicitudes, y manifestaron que el proceso, aunque sea tradicional, es más simple y rápido de esta manera. Con este acuerdo, se contribuye a estandarizar el flujo de proceso entre las escuelas, en vez de forzar al sistema a adaptarse a las particularidades de cada una.
- En la reunión con las UAA, se discutió y aclaró la normativa de cada escuela para asegurar que el modelado de las mismas fuese correcto en el sistema. Todas las escuelas estuvieron de acuerdo con las restricciones implementadas en cuanto al número de solicitudes de paralelos, extracréditos y tipos de asignaturas permitidas.
- Los usuarios están conscientes de que existen casos que el sistema no puede abarcar, por lo que las escuelas tienen mecanismos para la recepción y gestión manual de solicitudes a través de cartas al Consejo de Escuela. Es también por la existencia de este procedimiento que el rediseño del proceso no resultó en un gran impacto para la mayoría de las escuelas.
- El formato final de la planilla de solicitud fue corregido a lo largo de las iteraciones en base a la retroalimentación de los usuarios. Este documento fue presentado y aprobado por los representantes de las UAA de las diferentes escuelas que asistieron a la reunión.
- El funcionamiento del sistema, tanto para el estudiante, como para el personal administrativo, fue aprobado por los usuarios. La opinión generalizada es la facilidad de uso y simplicidad de la aplicación.
- El hecho de proporcionar información al estudiante sobre el reglamento mientras realiza sus solicitudes fue un aspecto tratado de forma muy positiva, ya que esto puede contribuir

a reducir el número de solicitudes y permite al estudiante estar informado de antemano sobre la viabilidad de las mismas.

En la siguiente sección se describe el comportamiento del sistema, así como los resultados obtenidos durante su puesta en producción.

4.6.4. Despliegue

El despliegue de la aplicación fue realizado a tiempo para soportar el proceso de solicitudes de modificación de inscripción del período 01-2014. Inicialmente, hubo algunas fallas relacionadas con el manejo de la oferta académica presentada a los estudiantes, la cual está constituida por las asignaturas disponibles para los mismos. Debido a errores en la interpretación de la definición de los paralelos, así como en el manejo de asignaturas electivas, algunas asignaturas no se mostraron correctamente en las primeras horas del despliegue. Estas fallas fueron corregidas rápidamente, de forma tal que las asignaturas disponibles correspondieran a aquellas que los estudiantes verdaderamente pudieran solicitar para su inscripción.

Los estudiantes presentaron algunas dudas comunes al usar el sistema, las cuales se basaron principalmente en el por qué el sistema no mostraba algunas asignaturas, o no les permitía realizar ciertas acciones. Estas dudas fueron solventadas oportunamente por el personal de la DCE, ya que no estaban relacionadas con problemas derivados del sistema.

Durante la evaluación y el estudio de las solicitudes, se detectó que las sugerencias del sistema necesitaban incluir el rendimiento del estudiante para emitir recomendaciones más acertadas y realistas. Este comportamiento se incorporó como una regla que toma en cuenta la eficiencia del estudiante, la cual puede ser personalizada para cada escuela.

En general, la aplicación tuvo un comportamiento muy estable, y los estudiantes pudieron realizar sus solicitudes sin mayores inconvenientes. En total, se realizaron **216** solicitudes de modificación de inscripción en el período 01-2014, hechas a su vez por 200 estudiantes de la Facultad de Ciencias, lo cual da como resultado un promedio de una solicitud por estudiante. Esto contrasta ampliamente con el promedio de solicitudes usual para todas las escuelas, el cual es de 400. Cabe resaltar que en este período no se ofertaron asignaturas de Idiomas (Inglés I e Inglés II), y el sistema no implementó la oferta de asignaturas PCI, lo cual pudo generar un impacto en el descenso de solicitudes. Las solicitudes de asignaturas PCI se gestionaron manualmente en las escuelas, pero posteriormente se implementaron en el sistema para poder soportarlas en el próximo período.

La Escuela de Química fue la que realizó más solicitudes de modificación de inscripción a través de CONEST, teniendo 85 solicitudes en total entre las carreras de Química y Geoquímica, seguida de la Escuela de Computación, con 64 solicitudes en total. La Tabla 4.6 presenta los resultados detallados por tipo de solicitud y por carrera.

Capítulo 4. Desarrollo del módulo solicitudes

Tabla 4.6: Cantidad de solicitudes de modificación de inscripción realizadas mediante CONEST 3.0 en el semestre 01-2014.

Carrera	Paralelo	Inscripción Extemporánea	Extracédito, Inscripción extemporánea	Extracédito, Paralelo	Extracédito, Paralelo, Inscripción extemporánea	Total
Biología	12	13	1	0	0	26
Computación	20	32	9	3	0	64
Física	17	11	1	1	0	30
Geoquímica	12	17	3	0	1	33
Matemática	1	8	1	1	0	11
Química	19	25	8	0	0	52
Total	81	106	23	5	1	216

Al analizar los resultados, se observa que las solicitudes de inscripción extemporánea son las más numerosas, seguidas de las solicitudes de paralelo, mientras que las solicitudes de exceso de créditos, y el resto de combinaciones, son menos frecuentes. Estos resultados se consolidan en la figura 4.40.

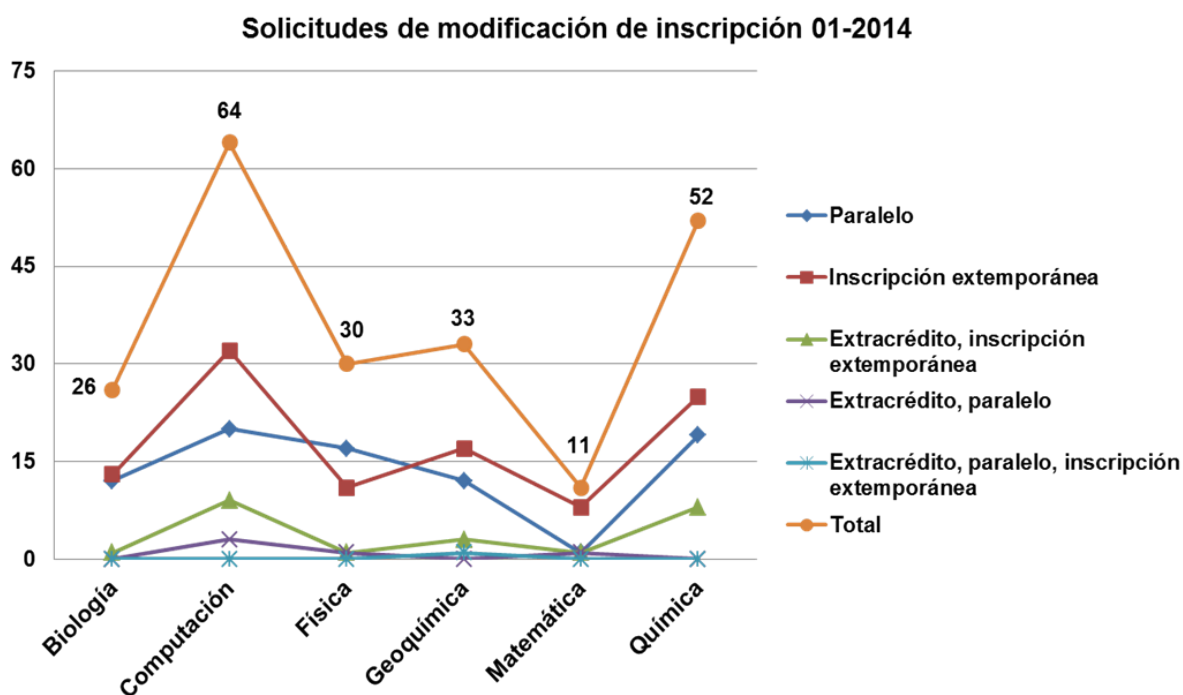


Figura 4.40: Tendencia de las solicitudes modificación de inscripción realizadas a través de CONEST en el semestre 01-2014.

El número real de solicitudes recibidas por las escuelas posiblemente sea menor al mencionado anteriormente, puesto que algunos estudiantes no formalizaron sus solicitudes entregando la planilla correspondiente en la escuela. Esto puede deberse a no obtener el aval de

Capítulo 4. Desarrollo del módulo solicitudes

los docentes, poca viabilidad de que sus solicitudes sean aprobadas al saber que no cumplen determinados requisitos, cambio de opinión, entre otras razones. Respecto al ámbito administrativo, no hubo mayores contratiempos en el procesamiento de las solicitudes. La DCE recibió las decisiones finales sobre las solicitudes y las procesó acorde a ellas, haciendo uso del sistema.

Como resultado de la implementación del proceso rediseñado y de su automatización, se destacan los siguientes resultados:

- Disminución considerable del número de solicitudes realizadas por los estudiantes a través del sistema, por lo que la carga de trabajo fue menor para las distintas escuelas. Sin embargo, fue necesario el procesamiento manual de solicitudes relacionadas con asignaturas PCI, así como del resto de solicitudes dirigidas al Consejo de Escuela.
- Las escuelas cuentan con un respaldo físico de las decisiones emitidas en caso de alguna contingencia. Además, tienen la facilidad de exportar las solicitudes a través del sistema mediante un archivo *Excel* para su control interno e integración con las solicitudes que son gestionadas de forma manual.
- El impacto en el ahorro de papel fue considerable. Al inicio del proceso, se manejaron cantidades significativas de planillas al ser recibidas en las escuelas, pero posteriormente las mismas fueron distribuidas a los entes correspondientes, sin acarrear mayores costos.
- La incorporación de la normativa de las escuelas impulsó la actualización del reglamento de la Escuela de Computación. Este puede ser un precedente para las demás, ya que el objetivo del sistema es precisamente soportar estos cambios.
- El uso del gestor de reglas reduce la cantidad de código necesario para modelar las reglas de todas las escuelas. Esto fue posible mediante la generalización de los métodos, lo cual a su vez facilita la modificación de las mismas.

Con la experiencia obtenida gracias a la puesta en producción de la aplicación, se espera que el próximo proceso de modificación de inscripción sea más flexible y abarque a las solicitudes de inscripción de asignaturas PCI, a fin de que estas sean incluidas en el sistema y se agilice su procesamiento.

4.7. Iteración 7 - Solicitud de reincorporación

El proceso de reincorporación, que es complejo y requiere la intervención de diversas dependencias, no es objeto de un rediseño profundo en este trabajo, sino que sólo se busca realizar la automatización de las actividades más importantes con el fin de incorporarlo al sistema CONEST 3.0. De esta forma, también se desea proporcionar información relevante a los involucrados sobre el cumplimiento del reglamento asociado a estas solicitudes, apoyando así la toma de decisiones a la hora de aprobarlas o rechazarlas. Estos requerimientos se describen en detalle en la fase de Análisis, presentada a continuación.

4.7.1. Análisis

Los requerimientos funcionales más relevantes para este proceso, derivados de varias reuniones con los usuarios, son los siguientes:

- Permitir a los estudiantes no inscritos realizar solicitudes de reincorporación mediante el sistema, redactando una carta dirigida tanto al Consejo de Escuela como al Servicio de Orientación, para exponer los motivos de la petición. Como requisito previo, los estudiantes deben proporcionar datos y una foto actualizada en el sistema, ya que pueden haber estado desincorporados por un largo período de tiempo.
- El estudiante debe descargar una planilla de reincorporación, la cual contiene un análisis del reglamento que rige a estas solicitudes, proporcionando los datos relevantes para ser estudiados por los entes correspondientes, así como para recoger las decisiones pertinentes. Asimismo, debe contener la cantidad de copias correspondientes para cada actor que interviene en el proceso, incluyendo la carta de exposición de motivos y el *kardex* del estudiante.
- El reglamento debe ser modelado y gestionado de forma similar al de las solicitudes de modificación de inscripción, permitiendo su adaptación a futuro.
- Proporcionar funcionalidades para la consulta y procesamiento de solicitudes, incluyendo notificaciones al estudiante.
- Permitir exportar las solicitudes en formato *Excel* para el control interno de los distintos entes involucrados, diferenciando las solicitudes de aquellos estudiantes que han cumplido con la sanción establecida en el reglamento, de aquellos estudiantes que se han desincorporado de manera voluntaria.

Teniendo estos requerimientos en cuenta, se da inicio a la fase de Diseño, descrita a continuación.

4.7.2. Diseño

En vista de que los requerimientos mencionados anteriormente son similares a los relacionados con las solicitudes de modificación de inscripción, no es necesario realizar prototipos de interfaz para la organización de las funcionalidades, ya que es posible reutilizar suficiente código de este proceso y adaptarlo para el manejo de las reincorporaciones. El objetivo es mantener las interfaces uniformes y aprovechar el conocimiento adquirido en los demás procesos descritos en el presente TEG. Por lo tanto, esta fase consistió principalmente en el diseño de la estructura de datos que debe soportar las reglas relacionadas con las reincorporaciones.

4.7.2.1. Esquema de base de datos

El diseño de esta estructura se basa en el del gestor de reglas para las solicitudes de modificación de inscripción, pero se requiere su generalización debido a ciertas diferencias entre ambos. Por ende, se ha diseñado una estructura capaz de modelar y soportar las reglas de reincorporación, así como otro tipo de reglas generales. La figura 4.41 presenta la estructura de las tablas diseñadas para este fin.

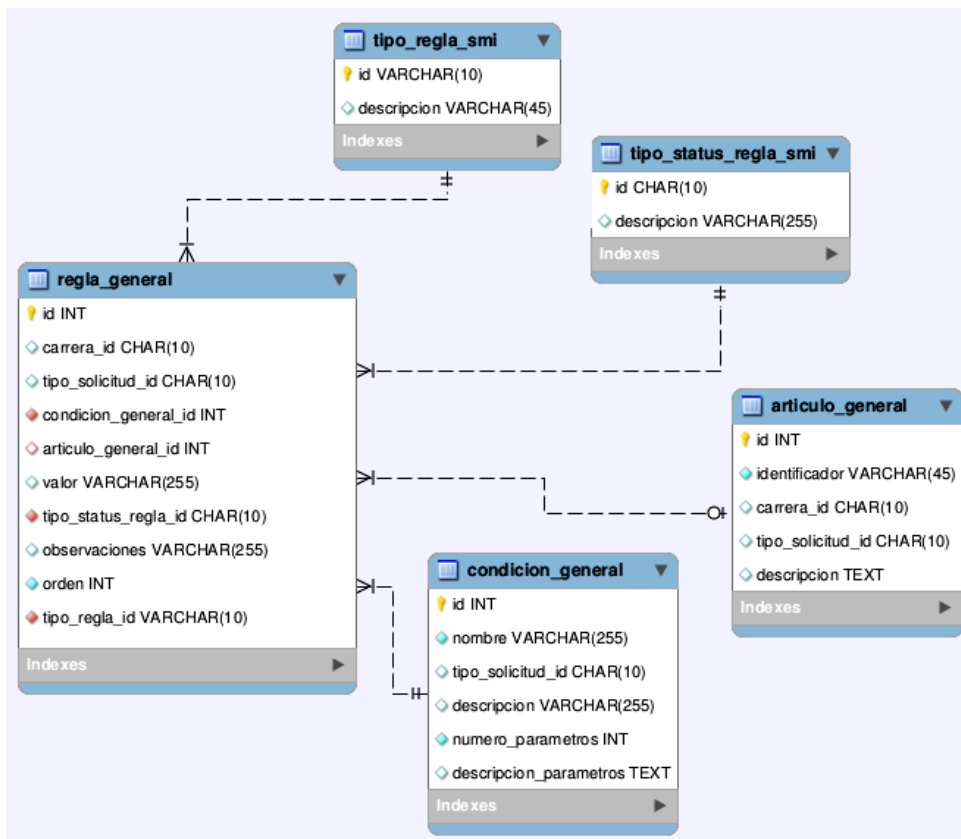


Figura 4.41: Esquema de base de datos para el gestor de reglas de solicitudes de reincorporación.

El diseño de la planilla de solicitud de reincorporación ha sido heredado y adaptado de las solicitudes de modificación de inscripción, siendo ajustado a través de la retroalimentación de los usuarios. La fase de Construcción, presentada a continuación, describe los resultados de la codificación de los requerimientos funcionales relativos a este proceso.

4.7.3. Construcción

Los primeros requerimientos implementados están relacionados con el estudiante y las acciones que permiten generar la solicitud de reincorporación. Para ello, el estudiante que no se ha inscrito en el período actual, y habiendo actualizado sus datos personales y su foto previamente, dispone de un botón a través del cual se despliega una ventana modal que le permite redactar una carta de exposición de motivos dirigida al Consejo de Escuela y al Servicio de Orientación. La figura 4.42 muestra una captura de esta interfaz.

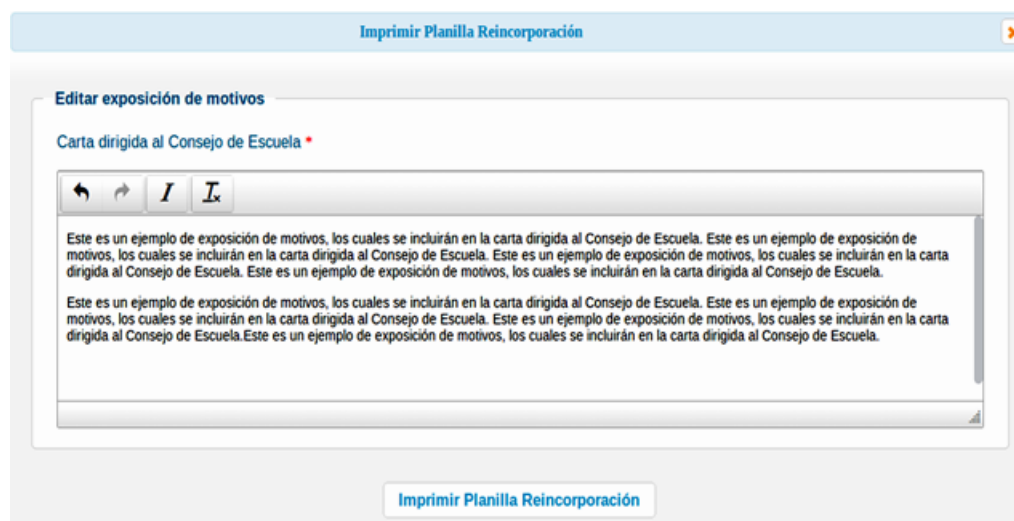


Figura 4.42: Interfaz gráfica del estudiante para realizar solicitudes de reincorporación.

Posteriormente, el estudiante podrá descargar la planilla de reincorporación mediante un botón dispuesto para ello, para así entregarla en la taquilla de la DCE. Esta planilla contiene el reglamento que rige a las reincorporaciones, y proporciona los datos del estudiante que están relacionados con el cumplimiento de las reglas, tales como la eficiencia, la cantidad de reincorporaciones anteriores y el número de períodos cursados por el estudiante, entre otros. El formato de esta planilla puede consultarse en el Anexo M, ubicado en la página 124.

Respecto al gestor de reglas de reincorporación, como se mencionó anteriormente, está basado en el gestor de modificación de inscripción, por lo que su interfaz y funcionamiento interno es bastante similar. La figura 4.43 muestra los métodos que implementan las diversas reglas de reincorporación, las cuales son más reducidas y más sencillas que las de la modificación de inscripción.

Capítulo 4. Desarrollo del módulo solicitudes

```

# Metodos que implementan las reglas de reincorporacion

def obtener_datos_academicos
end

def estudiante_articulo_6_puede_reincorporacion_inmediata?
end

def eficiencia_estudiante(operador, eficiencia)
end

def unidades_de_credito_aprobadas_licenciatura(operador, porcentaje)
end

def numero_periodos_cursados(operador, nro_periodos)
  estudiante = Estudiante.where(cedula: @estudiante_cedula).take
  nro_periodos_cursados = estudiante.periodos_cursados(@carrera_id).size
  razones = "PERIODOS CURSADOS: #{nro_periodos_cursados}"
  return eval("#{nro_periodos_cursados} #{operador} #{nro_periodos}"), razones
end

def numero_de_veces_reincorporado(operador, cantidad)
end

def anos_desincorporado(operador, anos)
end

# Metodo para reglas que no se ejecutan, sino que sólo se desea mostrar el artículo
def metodo_indefinido
  return true
end

```

Figura 4.43: Métodos que implementan las reglas de reincorporación.

La interfaz que permite la administración de estas reglas se muestra en la figura 4.44. Este proceso también cuenta con un pequeño gestor de configuración que permite establecer la fecha límite de recepción de solicitudes de reincorporación, así como abrir el proceso para los estudiantes.

Reglas Generales						
GENERALES BIOLOGIA COMPUTACION FISICA GEOQUIMICA MATEMATICA QUIMICA						
Reglas encontradas: 8						
Orden	Tipo Solicitud	Artículo	Condición	Valor	Tipo Regla	Estado
1	SOLICITUD DE REINCORPORACION	1.1	ESTUDIANTE ARTICULO 6 PUEDE REINCORPORACION INMEDIATA?		OBLIGATORIA	ACTIVA
2	SOLICITUD DE REINCORPORACION	1.2.a)	UNIDADES DE CREDITO APROBADAS LICENCIATURA	>=,20	OBLIGATORIA	ACTIVA
3	SOLICITUD DE REINCORPORACION	1.2.b)	EFICIENCIA ESTUDIANTE	>,0.40	OBLIGATORIA	ACTIVA
4	SOLICITUD DE REINCORPORACION	1.2.c)	NUMERO DE VECES REINCORPORADO	<=,1	OBLIGATORIA	ACTIVA
5	SOLICITUD DE REINCORPORACION	1.2.d)	ANOS DESINCORPORADO	<,5	OBLIGATORIA	ACTIVA
6	SOLICITUD DE REINCORPORACION	1.2.e)	NUMERO PERIODOS CURSADOS	<=,15	OBLIGATORIA	ACTIVA
7	SOLICITUD DE REINCORPORACION	1.3.	METODO INDEFINIDO		OBLIGATORIA	ACTIVA
8	SOLICITUD DE REINCORPORACION	2.	METODO INDEFINIDO		OBLIGATORIA	ACTIVA

Figura 4.44: Interfaz gráfica del gestor de reglas de reincorporación.

Capítulo 4. Desarrollo del módulo solicitudes

En el ámbito administrativo, se cuenta con una interfaz de consulta de solicitudes sencilla, similar para todos los roles involucrados, la cual también permite exportar estas solicitudes en un archivo *Excel*, realizando una diferenciación entre aquellos estudiantes que cumplieron con la sanción especificada en el reglamento, y que desean reincorporarse en el siguiente semestre, y aquellos que se desincorporaron voluntariamente. Esto permite priorizar los casos que han cumplido con la sanción para ofrecer respuestas oportunas.

La figura 4.45 muestra una captura de la interfaz administrativa de la DCE, que es el organismo que posee los permisos para realizar el procesamiento final de las solicitudes.

Nro	Carrera	Estudiante	Solicitud	Estado
1	C	5161956 - ARIAS DOMINGO	20/07/2014 	✓ PROCESAR
2	C	18896239 - BERMUDEZ JUAN	20/07/2014 	✓ PROCESAR
3	C	17962941 - BLONDELL RICHARD	20/07/2014 	✓ PROCESAR
4	C	15872514 - BUITRAGO CARLOS	20/07/2014 	✓ PROCESAR
5	C	19195103 - CASTILLO ANGEL	20/07/2014 	✓ PROCESAR
6	C	4883119 - DELGADO FELIX	20/07/2014 	✓ PROCESAR

Figura 4.45: Interfaz de consulta de solicitudes de reincorporación para la DCE.

De esta forma, se implementaron los requisitos básicos establecidos en la fase de Análisis. La siguiente fase, correspondiente a la de Pruebas, se describe a continuación.

4.7.4. Pruebas

A parte de las pruebas manuales realizadas durante la fase de Construcción, estas consistieron principalmente en pruebas de aceptación realizadas con los usuarios de la DCE, incluyendo algunos usuarios de las escuelas. Estas pruebas consistieron en la demostración, evaluación y aprobación del proceso automatizado, así como de las planillas y documentos generados durante el mismo. La retroalimentación obtenida fue tomada en cuenta y aplicada mediante varios ajustes en la aplicación y en los diversos formatos generados.

Asimismo, se realizó la migración de un conjunto de solicitudes de reincorporación que estaban pendientes y debían ser procesadas haciendo uso del sistema. Esto sirvió como una primera prueba del funcionamiento del gestor, de la cual se derivaron algunos ajustes en la implementación de los métodos, así como la actualización de la cantidad créditos establecidos para cada carrera, los cuales están almacenados en la base de datos, a fin de que la información proporcionada por el gestor fuese correcta.

La opinión de los distintos usuarios participantes en las pruebas coincidió en que esta automatización puede contribuir a agilizar el proceso, que se gestionaba de forma manual, y en general, estuvieron de acuerdo con las funcionalidades y mecanismos implementados en el sistema.

4.7.5. Despliegue

Durante esta fase, la aplicación fue integrada al servidor de producción y habilitada para el personal administrativo, a fin de hacer posible el procesamiento de las solicitudes de reincorporación migradas en la fase anterior. Sin embargo, la aplicación no fue abierta para los estudiantes, ya que para el momento de la elaboración de este trabajo no correspondía el período de recepción de solicitudes de reincorporación.

Una vez que se habilite el sistema a los estudiantes en el período determinado para ello, se espera poder contar con una aplicación estable, que agilice el proceso de reincorporación, y apoye la toma de decisiones, con el fin de prestar un mejor servicio.

De esta forma, se da por terminado este capítulo, donde se han descrito las diversas iteraciones realizadas para el desarrollo del módulo de las solicitudes estudiantiles.

A continuación, se presentan las conclusiones y recomendaciones producto del desarrollo de este trabajo.

Conclusiones y recomendaciones

Los Sistemas de Gestión Académica son implementados en instituciones educativas con el fin de resolver la problemática relacionada con la gestión de procesos complejos para un elevado número de usuarios, brindando a su vez un servicio de calidad. CONEST 3.0 es un sistema evolucionado en base a una amplia experiencia y dedicación, que ha adoptado esta premisa como piedra angular para solventar las deficiencias actuales y mejorar los servicios que ofrece la Facultad de Ciencias a su amplia comunidad. Mediante el desarrollo del presente trabajo, se ha obtenido una serie de resultados que contribuyen a alcanzar este propósito, y que están estrechamente relacionados con los objetivos planteados. Dichos resultados se describen a continuación:

- Se automatizó el proceso de solicitud de jurado y solicitud de planilla de notas para seminario y TEG, proporcionando un mecanismo de integración de la información aprobada en la solicitud de jurado y la planilla de notas, a fin de garantizar su consistencia. Como resultado, se produjo una reducción en los gastos materiales, así como en el tiempo y esfuerzo invertido para la gestión de estas solicitudes. Esta automatización permitió agilizar y unificar el proceso general de solicitudes de jurado y planillas de notas para las escuelas, estandarizar el flujo de documentos originados durante el mismo, y apoyar las actividades relacionadas con el proceso de grado de la facultad.
- Se diseñó el flujo de comunicación del proceso de solicitudes de modificación de inscripción, ya que se descartó una cantidad significativa de acciones propensas a provocar retrasos, simplificando este proceso en gran medida. Los beneficios más notables después de poner a prueba los cambios se traducen en la reducción de los costos materiales, la estandarización de documentos, y una gestión más sencilla de las solicitudes para el personal administrativo, independientemente de la complejidad de los procesos internos de las diferentes escuelas.
- Se automatizó el procedimiento de solicitudes de reincorporación, incluyendo mecanismos para apoyar la toma de decisiones de las instancias evaluadoras que participan a lo largo del proceso.
- Se aplicó al sistema el reglamento relacionado con las solicitudes de modificación de inscripción, así como el que rige a las solicitudes de reincorporación. Esto hace posible brindar la retroalimentación necesaria, tanto para el estudiante, como para las UAA de las diversas escuelas, lo que facilitó el análisis de las solicitudes y la toma de decisiones.

Conclusiones y recomendaciones

- Se desarrolló de forma exitosa un gestor para la administración del reglamento de solicitudes de modificación de inscripción y del reglamento de solicitudes de reincorporación, cuyo funcionamiento básico puede ser extendido y aplicado al modelado de otros reglamentos y normativas de la facultad que sean susceptibles a sufrir cambios a lo largo del tiempo. Este mecanismo permite flexibilizar el sistema y adaptarlo a diversas particularidades, favoreciendo a su vez el cumplimiento de las normativas.

De esta manera, es posible afirmar que se logró el cumplimiento satisfactorio del objetivo general planteado al inicio de este trabajo, implementando una solución a la problemática relacionada con la gestión de solicitudes estudiantiles y docentes, e integrándola con éxito al sistema CONEST.

Para la consecución de los objetivos planteados, se utilizó un método ágil a lo largo del desarrollo que facilitó la obtención de los requisitos mediante reuniones frecuentes con los usuarios y el intercambio de información a través de tormentas de ideas, priorizando el diseño y construcción ágil de soluciones, produciendo además la documentación necesaria. Este método de desarrollo iterativo e incremental toma aspectos importantes de métodos como AgilUs y XP para reforzar esas fases específicas, ya que por sí mismos, estos métodos tienen algunas carencias y limitaciones que dificultan su aplicación directa. AgilUs hace especial hincapié en la obtención y análisis de requisitos, así como en el prototipaje de soluciones enfocadas en la usabilidad para sistemas altamente interactivos, mientras que XP descuida un poco este aspecto, otorgándole mayor relevancia a la construcción ágil de la aplicación a través de una serie de buenas prácticas, aspecto donde precisamente AgilUs se queda corto, especialmente cuando se trata de aplicaciones de alta complejidad.

En resumen, la intención del método propuesto y aplicado en este trabajo es poner en práctica algunos de los aspectos más relevantes encontrados en ambos métodos, aprovechando sus ventajas y compensando las deficiencias para obtener resultados positivos en el desarrollo del sistema, evitando forzar la adaptación excesiva de un método concreto al cambiar muchas de sus ideas fundamentales hasta hacerlo prácticamente irreconocible.

Un factor importante que contribuyó considerablemente en el desarrollo ágil del sistema es el empleo de un *framework* como *Ruby on Rails*, ya que su estructura, simplicidad y las facilidades que ofrece en general, permitieron reducir la complejidad de la codificación y facilitar la reutilización de código, lo cual se tradujo en la entrega más rápida de resultados concretos. El uso del conjunto de tecnologías seleccionadas hizo posible, y facilitó en gran medida, la implementación de las soluciones planteadas en este trabajo.

Por otra parte, el uso de una herramienta para realizar pruebas automatizadas resultó de utilidad en la codificación y validación de la aplicación en un inicio; sin embargo, por las restricciones de tiempo, el uso de esta herramienta se vio afectado en las últimas iteraciones, ya que el proceso de escribir pruebas para una gran variedad de escenarios requiere una cantidad

Conclusiones y recomendaciones

significativa de tiempo. Además, existen casos donde es muy difícil escribir pruebas automatizadas, por lo que en algunos escenarios complejos es necesario el juicio del usuario y la ejecución de pruebas manuales.

No obstante, la razón del desarrollo exitoso de este sistema, está dada por la disposición de los usuarios y su participación a lo largo del proceso, durante las fases de Análisis, Diseño, Pruebas y Despliegue, proporcionando una valiosa retroalimentación y numerosos aportes que contribuyeron a mejorar considerablemente el funcionamiento de la aplicación para ofrecer un mejor servicio. Es de vital importancia comprender la forma de pensar de los usuarios, para así comprender sus necesidades y plantear las mejores soluciones que puedan resolver la problemática que les aqueja.

Participar en un proyecto tan amplio e importante como CONEST, destinado a la puesta en producción en un ambiente real para ofrecer servicios altamente demandados por la comunidad de la Facultad de Ciencias, con un equipo de desarrollo integrado y usuarios altamente comprometidos, fue sin duda una experiencia gratificante y aleccionadora, que requirió una alta carga de responsabilidad, donde se evidenció la importancia del trabajo mancomunado, y donde fue posible poner en práctica los conocimientos adquiridos a lo largo de la carrera, así como adquirir nuevas destrezas relacionadas con el despliegue de aplicaciones web, con el fin de obtener resultados reales y positivos para los usuarios.

Finalmente, las recomendaciones que se proponen para mejorar y extender el funcionamiento del sistema desarrollado en este trabajo, así como de CONEST en general, son:

- Automatizar procesos similares al de reincorporación, tales como las solicitudes de cambio de escuela y egresados.
- La validación del *pensum* de las diferentes carreras es un requerimiento que está relacionado con el proceso de grado y que es altamente demandado por el personal de la DCE, ya que es un proceso engorroso y puede ser automatizado. Como el *pensum* está conformado por un conjunto de requisitos que los estudiantes deben cumplir, este pudiera ser manejado a través de un gestor de reglas que permita su validación y adaptación a futuro.
- Existen reglas y parámetros dispersos para el control de diversos procesos, los cuales deberían ser estandarizados para mejorar la organización interna del sistema y facilitar su modificación a lo largo del tiempo. Este estándar puede basarse en el uso de estructuras de datos uniformes para la generalización de las reglas, así como en interfaces de usuario que faciliten su administración.

Referencias

- Acosta, A. (2011). AgilUs: un método ágil de desarrollo de software que incorpora la usabilidad. *Universidad Central de Venezuela*, 20–36. Recuperado de: <http://www.ciens.ucv.ve/escueladecomputacion/documentos/archivo/121>. Lecturas en Ciencias de la Computación.
- AdaptativePath (2005). A new approach to Web applications. Recuperado de: <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>.
- Albaladejo, X. (n.d.). Desarrollo iterativo e incremental. Recuperado de: <http://www.proyectosagiles.org/desarrollo-iterativo-incremental>.
- Beck, K. (2000). *Extreme Programming explained. Embrace change*. Pearson Education.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., F., & M., Thomas, D. (2001). Manifiesto for Agile Software Development. Recuperado de: <http://agilemanifesto.org/>.
- Belfort, P. & Ojeda, L. (2009). Desarrollo de la aplicación de gestión académica CONEST 2.0: módulo administrativo. *Universidad Central de Venezuela*. Trabajo Especial de Grado.
- Brown, G. (2008). Fast, nimble PDF writer for Ruby. Recuperado de: <https://github.com/prawnpdf/prawn>.
- Carneiro, Y. & Pérez, G. (2013). Desarrollo del módulo de gestión de solicitudes estudiantiles para el sistema CONEST 3.0 de la Facultad de Ciencias. *Universidad Central de Venezuela*. Seminario.
- CoffeeScript (n.d.). CoffeeScript. Recuperado de: <http://coffeescript.org/>.
- Da Silva, A. (n.d.). Programación en el lado del cliente. Recuperado de: <http://tecsanmartinisc.galeon.com/CLIENTE.pdf>.
- EcuRed (2012). Lenguaje de programación Ruby. Recuperado de: www.ecured.cu/index.php/Lenguaje_de_Programación_Ruby.
- Eguiluz, J. (2013a). Introducción a AJAX. Recuperado de: <http://www.librosweb.es/ajax>.
- Eguiluz, J. (2013b). Introducción a CSS. Recuperado de: <http://www.librosweb.es/css>.
- Eguiluz, J. (2013c). Introducción a JavaScript. Recuperado de: <http://www.librosweb.es/javascript>.

Referencias

- Git (n.d.). Git - About version control. Recuperado de: <http://git-scm.com/book/en/Getting-Started-About-Version-Control>.
- Grove, R. (2010). Web based application development. *USA: Jones & Barlett Publishers, LLC*, 17–21.
- Haml-Documentation (2013). Reference Haml Documentation. Recuperado de: <http://haml.info/docs/yardoc/file.REFERENCE.html>.
- Jaldín, R. (2010). Implementación - Fase de Construcción - RUP. Recuperado de: <http://rolandojaldin.blogspot.com/2010/10/implementacion-fase-de-construccion-rup.html>.
- Janssen, C. (n.d.). What is a Web-based application? Recuperado de: <http://www.techopedia.com/definition/26002/web-based-application>.
- Joskowicz, J. (2008). Reglas y prácticas en eXtreme Programming. Recuperado de: <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>.
- jQuery (2013). Página oficial. Recuperado de: <http://jquery.com/>.
- jQueryUI (2013). Página oficial. Recuperado de: <http://jqueryui.com/>.
- JSON.org (n.d.). Introducción a JSON. Recuperado de: <http://json.org/json-es.html>.
- Larman, C. (2003). *UML y Patrones*. Prentice-Hall.
- Lemus, J. (2007). CSS 3: Más social que nunca. Recuperado de: <http://www.maestrosdelweb.com/editorial/css-3-mas-social-que-nunca/>.
- Matsuda, A. (2011). A scope & engine based, clean, powerful, customizable and sophisticated paginator for Rails 3 and 4. Recuperado de: <https://github.com/amatsuda/kaminari>.
- Mendez, N. & Boyer, Y. (2008). Automatización de procesos relacionados con las solicitudes estudiantiles y actividades administrativas y de docencia de la Facultad de Ciencias. *Universidad Central de Venezuela*. Trabajo Especial de Grado.
- MySQL (n.d.a). MySQL 5.1 Reference manual. Recuperado de: <http://dev.mysql.com/doc/refman/5.1/en/what-is-mysql.html>.
- MySQL (n.d.b). MySQL Customers. Recuperado de: <http://www.mysql.com/customers>.
- Ortiz, M. (2012). Infografía - Modelo iterativo e incremental. Recuperado de: <http://isw-udistrital.blogspot.com/2012/09/infografia-modelo-iterativo-e.html>.
- Pressman, R. (2002). *Ingeniería del Software. Un enfoque práctico* (5ta ed.). Madrid: McGraw-Hill, 5ta edition.

Referencias

- RailsGuides (2013). Ruby on Rails 4.0 Release notes. Recuperado de: http://edgeguides.rubyonrails.org/4_0_release_notes.html#major-features.
- Rails.mx (n.d.). La comunidad mexicana de Ruby on Rails. Recuperado de: <http://rails.mx/>.
- Rivas, S., Correa, E., & Metzner, C. (2010). Ruby: Algunas características del lenguaje dinámico. *Lecturas en Ciencias de la Computacion*, ISSN 1316-6239.
- Sass (n.d.). Sass: Syntactically Awesome Style Sheets. Recuperado de: <http://sass-lang.com>.
- Stocker, M. (2013). Rails 4 Released: faster pages with turbolinks. Recuperado de: <http://www.infoq.com/news/2013/06/rails4>.
- Suárez, D. (2013). Haciendo BDD con Cucumber. Recuperado de: <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=bdd-con-cucumber>.
- TrazosWeb (2010). HTML5: Qué es y cómo usarlo. Recuperado de: <http://www.trazos-web.com/2010/02/01/html5-que-es-y-como-usarlo/>.
- Universidad de Alicante (2012). Modelo Vista Controlador (MVC). Recuperado de: <http://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>.
- W3C (n.d.). Guia breve de CSS. Recuperado de: <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>.
- W3schools (n.d.a). AJAX introduction. Recuperado de: http://www.w3schools.com/ajax/ajax_intro.asp.
- W3schools (n.d.b). JSON tutorial. Recuperado de: <http://www.w3schools.com/json/>.
- Walker, G. (2011). Fundamentos HTML5, parte 1. Recuperado de: <http://www.ibm.com/developerworks/ssa/web/library/wa-html5fundamentals/>.

Anexos

Anexo A - Extracto de la normativa de solicitudes estudiantiles de la Escuela de Biología



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Biología
Dirección

NORMATIVA PARA LA DIVISION DE CONTROL DE ESTUDIOS SOBRE LAS SOLICITUDES ESTUDIANTILES EN FORMATO ELECTRONICO, DE LA ESCUELA DE BIOLOGÍA

Normativas aprobadas en el Consejo de Escuela del xx/xx/2011

Artículo 1. Los estudiantes harán vía web (a través de CONEST) sus solicitudes de modificación de inscripción (paralelos, inscripciones extemporáneas, retiros, exceso de créditos, estudios dirigidos) durante las dos primeras semanas de iniciado el semestre.

Artículo 2. Sólo se aceptarán hasta dos (2) solicitudes de paralelo de materias (una asignatura de Teoría y una de Laboratorio), por semestre, para cada estudiante.

Artículo 3. No se concederán paralelos de materias que no hayan sido cursadas completas el semestre inmediato anterior. Las materias retiradas no serán consideradas para solicitudes de paralelos.

Artículo 4. No se concederán paralelos de materias a estudiantes que hayan aplazado más de una materia en el semestre inmediato anterior (artículo 156 de la Ley de Universidades).

Artículo 5. El paralelo de Inglés I con la asignatura Seminario I se aprobará automáticamente y de acuerdo a lo establecido en el Artículo 2 de esta Normativa.

Artículo 6. Los paralelos para cada materia se concederán una sola vez. No se concederán paralelos a una materia que no haya sido aprobada durante un paralelo anterior.

Artículo 7. Los estudiantes que soliciten la inscripción de una asignatura por el Sistema de Estudios Dirigidos (SED), deberán cumplir con el Reglamento de Estudios Dirigidos, a saber:

a) Los estudiantes que hubieran sido reprobados en la materia durante el curso ordinario, pero que hayan presentado, por lo menos, el setenta y cinco por ciento (75%) de las pruebas de evaluación realizadas durante dicho curso.

b) En caso de que se realizaran sólo dos (2) evaluaciones, deberán haber presentado la totalidad.

c) Los estudiantes que, sin haber cursado previamente esa materia, hayan demostrado buen rendimiento estudiantil en el semestre anterior.

Se entiende por buen rendimiento el haber aprobado todos los créditos en los cuales se hubieran inscrito en ese semestre.

Artículo 8. El Profesor coordinador del curso inmediato anterior de la asignatura que durante el semestre de la solicitud estudiantil será dictado por SED, será responsable de remitirle a la División de Control de Estudios el listado de los estudiantes que cumplen con el punto 7a, y podrán inscribirse para el curso SED.

Anexo B - Normativa de solicitudes de modificación de inscripción de la Escuela de Computación

Artículo 156 de la ley de Universidades: "los alumnos que resulten aplazados por primera vez en exámenes de reparación en no más de una asignatura pueden inscribirse condicionalmente en todas las asignaturas del curso inmediato superior y podrán presentar exámenes finales de una y otra en el período ordinario de exámenes. Si la asignatura pendiente tiene prelación sobre alguna o algunas del curso superior no podrán rendirse los exámenes de ésta sin haber aprobado previamente aquella. El Consejo de Facultad determinará el orden de prelación de asignaturas".

Extracto de la Normativa Aprobada por el Consejo de Escuela de Computación para Solicitudes de Modificación de Inscripción

Artículo 1: Durante las primeras dos semanas de cada semestre, estará disponible la solicitud de modificación de inscripción en línea a través del sistema diseñado para tal fin.

Artículo 2: El estudiante debe consignar la planilla de solicitud de modificación de inscripción con el kardex actualizado y cualquier otro soporte en la escuela.

Artículo 3: Sólo se aceptará una solicitud de paralelo entre asignaturas por semestre.

Artículo 4: No se concederán paralelos de asignaturas que no hayan sido aplazadas en el semestre inmediato anterior.

Artículo 5: No se concederán paralelos ni excesos de créditos de asignaturas a estudiantes que hayan aplazado más de una asignatura en el semestre inmediato anterior.

Artículo 6: Los paralelos para cada asignatura se concederán por una única vez.

Artículo 7: Excepcionalmente se considerará la inscripción extemporánea del Trabajo Especial de Grado con Seminario, siempre y cuando éste último este calificado como aprobado en las primeras tres semanas de iniciado el semestre.

Artículo 8: El estudiante que solicite exceso de créditos debe haber aprobado al menos 6 (seis) asignaturas en los dos semestres anteriores al de la solicitud.

Artículo 9: El exceso de crédito puede ser aprobado para una asignatura que sobrepase la cantidad máxima de créditos inscritos en el semestre (Máximo 24 U.C en semestre regulares e intensivos 1 asignatura).

Artículo 10: Si el exceso de crédito genera una situación de paralelo de asignaturas, deben cumplirse previamente todas las condiciones para el otorgamiento de los paralelos.

Artículo 11: No se concederán inscripciones extemporáneas de asignaturas obligatorias, sólo de asignaturas electivas y obligatorias optativas, previo aval del profesor que dicta la asignatura.

Artículo 12: Para efectos de la inscripción extemporánea, las asignaturas electivas y obligatorias optativas deben cumplir los requisitos de prelación. En caso de generarse paralelos o exceso de créditos, deben cumplir todas las condiciones previstas en la normativa.

Artículo 13: Todo lo no contemplado en esta normativa será discutido en el Consejo de Escuela.

Anexo C - Normativa de solicitudes estudiantiles de la Escuela de Química



Universidad Central de Venezuela

Facultad de Ciencias. Escuela de Química. ~ Departamento de Química ~
Los Chaguaramos. Caracas.

AP. 47102 ~ Venezuela. Teléfono 58-212-6051019 ~ 58-212-6051015 ~ Fax 58-212-6051220

NORMATIVAS DE LAS SOLICITUDES ESTUDIANTILES DE LA ESCUELA DE QUIMICA

Normativas aprobadas en la reunión Departamental efectuada el 24/03/2010

Artículo 1. Los estudiantes harán via web (a través de CONEST) sus solicitudes de modificación de inscripción (paralelos, inscripciones extemporáneas, exceso de créditos) durante las dos primeras semanas de iniciado el semestre.

Artículo 2. En el caso de que la solicitud requiera de algún tipo de soporte (aval de un profesor o de una Unidad Docente), éste debe ser entregado a la Unidad de Asesoramiento Académico.

Artículo 3. Sólo se aceptará una solicitud de paralelo de materia por semestre para cada estudiante.

Artículo 4. No se concederán paralelos de materias que no hayan sido cursadas completas el semestre inmediato anterior. Las materias retiradas no serán consideradas para solicitudes de paralelos (artículo 156 de la Ley de Universidades).

Artículo 5. No se concederán paralelos de materias a estudiantes que hayan aplazado más de una materia en el semestre inmediato anterior (artículo 156 de la Ley de Universidades).

Artículo 6. Los paralelos para cada materia se concederán por una única vez. No se concederán paralelos a una materia que no haya sido aprobada durante un paralelo anterior.

Artículo 7. El exceso de créditos generado por las materias Inglés I, Inglés II, Seminario de Investigación y las complementarias se aprobarán automáticamente.

Artículo 8. El estudiante que solicite exceso de créditos debe haber aprobado al menos tres materias en cada uno de los dos semestres anteriores al de la solicitud. Excepcionalmente se considerará si el estudiante está cursando sus últimas materias en la carrera.

Artículo 9. El exceso de créditos puede ser aprobado en más de una oportunidad si el desempeño del estudiante lo justifica de acuerdo al artículo anterior.

Artículo 10. Si una solicitud de paralelo genera un exceso de créditos, deben cumplirse previamente todas las condiciones para el otorgamiento de los paralelos para poder aprobar el exceso de créditos.

Artículo 11. Para efectos de la inscripción extemporánea, las materias electivas y obligatorias deben cumplir los requisitos de prelación. En caso de generarse paralelos, deben cumplir todas las condiciones previstas en la normativa. En todos los casos, la solicitud debe estar acompañada del aval por escrito del profesor de la materia.

Artículo 12. Para modificación de inscripciones, la UAA solamente considerará aquellos casos que a su juicio representen errores materiales durante el proceso de inscripción.

Artículo 13. El estudiante puede hacer el retiro total del semestre directamente ante la DCE. En este caso, para su reincorporación el estudiante debe cumplir los requisitos usuales de reincorporación (no podrá reincorporarse al semestre siguiente). Si el estudiante desea hacer un retiro con reincorporación inmediata debe tramitar su solicitud a través del Consejo de Escuela. Dicha solicitud debe ser justificada y venir acompañada de los soportes correspondientes (Enfermedad del estudiante o un familiar cercano; muerte de un familiar cercano; problemas económicos).

Artículo 14. Los retiros extemporáneos de materias se solicitarán directamente ante el Consejo de Escuela.

Anexo D - Normativa interna para reincorporaciones



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
COORDINACIÓN ACADÉMICA

NORMATIVA INTERNA PARA LAS REINCORPORACIONES A LA FACULTAD DE CIENCIAS (Aprobada por el Consejo de la Facultad de Ciencias en sesión ordinaria de fecha 17 de abril de 2006)

El ingreso de los alumnos por reincorporación a la Facultad de Ciencias se rige por el Reglamento de ingreso de alumnos a la Universidad Central de Venezuela, aprobado por el Consejo universitario en su sesión del día 8 de marzo de 2000 (Anexo), y por la presente Normativa Interna.

1. Se podrán reincorporar a la Facultad de Ciencias sin que se requiera un estudio del caso por la Unidad de Asesoramiento Académico :
 - 1.1. Los bachilleres que, Incursos en el Artículo N°6 de las Normas sobre Rendimiento Mínimo y Condiciones de Permanencia de los Alumnos en la UCV, soliciten reincorporarse en el semestre inmediatamente siguiente de haberse cumplido la sanción correspondiente.
 - 1.2. Los bachilleres que cumplan las siguientes cinco (5) condiciones:
 - a) Tener aprobado al menos, el 20% de las unidades de la Licenciatura correspondiente.
 - b) Tener un índice de Eficiencia mayor que 0,40
 - c) No haberse retirado y reincorporado anteriormente en más de una oportunidad.
 - d) Estar desincorporado menos de 5 años
 - e) No haber cursado más de 15 semestres en la Escuela a la cual solicita reincorporación.
 - 1.3. Los bachilleres a quienes el Consejo de Escuela respectivo les haya aprobado retirarse por un tiempo definido, para luego reincorporarse en un semestre determinado.
2. Los casos que no estén contemplados en la disposición anterior, serán estudiados por la Unidad de Asesoramiento Académico, la cual hará la recomendación razonada que considere pertinente en cada caso, ante el Consejo de Escuela respectivo.
3. Las solicitudes de reincorporación deben hacerse en el lapso establecido por la División de Control de Estudios de la Facultad de Ciencias.
4. Todos los casos de reincorporaciones deben ser aprobados por el Consejo de la Facultad de Ciencias

Procedimiento para el trámite de reincorporación

- Las solicitudes de reincorporación se recibirán por la Taquilla de la División de Control de Estudios dos veces al año:
 - Primer Período: Las dos primeras semanas del mes de mayo.
 - Segundo Período: La última semana del mes de noviembre y la primera semana del mes de diciembre.
- La solicitud debe incluir:
 - Planilla de solicitud de reincorporación.
 - Expediente Curricular actualizado (Kardex).
 - Carta de solicitud del estudiante dirigida al Director(a) de la Escuela respectiva
 - Constancia de entrevista en el Servicio de Orientación de la Facultad de Ciencias.
 - Foto actualizada tipo carnet.

RESOLUCIÓN: Aprobado, sesión del 17-04-2006.

Anexo E - Solicitud de jurado



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computacion



Fecha: 13/05/2014
Período Lectivo: 02-2013

Solicitud de Nombramiento de Jurado para Trabajo Especial de Grado

Estudiante Br. CARNEIRO BELISARIO YSABELLA ROSALINA, CI 19739906

Trabajo (6420) TRAB.ESP.GRADO(APLI.INTE)
Automatización de procesos de solicitudes docentes y estudiantiles para
el Sistema de Gestión Académica CONEST.

Opción APLICACIONES EN INTERNET

Proposición de Jurado

Jurado Principal Profa. Zambrano Jossie (Tutor Firmante)

Prof. Rivas Sergio (Tutor)

Profa. Villapol Maria

Prof. Scalise Eugenio

Jurado Suplente Profa. Acosta Alecia

Profa. Ojeda Lucia

Tutor
Profa. Zambrano Jossie

Coordinador/Jefe Dpto.

Nota: Se anexa Kardex del estudiante.

Anexo F - Carta de aprobación de solicitud de jurado



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computacion



EC-03.13.147
Caracas, 08 de agosto de 2014
Período Lectivo: 02-2013

MEMORÁNDUM

PARA: Prof. Damaris Barrantes, Jefa de División de Control de Estudios
DE: Profa. Zenaida Castillo, Directora de la Escuela de Computacion
ASUNTO: Nombramiento de Jurado para Trabajo Especial de Grado

Por medio de la presente me dirijo a usted con la finalidad de informarle que el Consejo de Escuela de Computacion, en sesión del día **21/05/2014**, aprobó la solicitud de jurado que evaluará el Trabajo Especial de Grado que consta de los siguientes datos.

Estudiante Br. **CARNEIRO BELISARIO YSABELLA ROSALINA**, CI 19739906
Asignatura (6420) TRAB.ESP.GRADO(APLI.INTE), OPCIÓN APLICACIONES EN INTERNET
Título Automatización de procesos de solicitudes docentes y estudiantiles para el Sistema de Gestión Académica CONEST.

Jurado Principal

Profa. Zambrano Jossie (Tutor Firmante)
Prof. Rivas Sergio (Tutor)
Profa. Villapol Maria
Prof. Scalise Eugenio

Jurado Suplente

Profa. Acosta Alecia
Profa. Ojeda Lucia

Sin otro particular al que referirme, quedo de usted.

Atentamente,

Profa. Zenaida Castillo
Directora de la Escuela de Computacion

Anexo H - Constancia de tutoría para TEG



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
COORDINACIÓN ACADÉMICA
DIVISIÓN DE CONTROL DE ESTUDIOS



CONSTANCIA DE TUTORÍA DE T.E.G

QUIEN SUSCRIBE, JEFE DE LA DIVISIÓN DE CONTROL DE ESTUDIOS DE LA FACULTAD DE CIENCIAS DE LA UNIVERSIDAD CENTRAL DE VENEZUELA, POR MEDIO DE LA PRESENTE HACE CONSTAR QUE LA PROFA. **ZAMBRANO JOSSIE**, PORTADORA DE LA CÉDULA DE IDENTIDAD N° **13872341**, ADSCRITA A LA ESCUELA DE COMPUTACION, HA PARTICIPADO COMO TUTOR DE LOS SIGUIENTES TRABAJOS ESPECIALES DE GRADO.

PERÍODO	AUTOR	TRABAJO ESPECIAL DE GRADO	NOTA
01-2007	14035663 - GESIA FRANCO 13531701 - DAVILA YETZINA	(6430) DESARROLLO DE UNA APLICACIÓN WEB CUENTE-SERVIDOR PARA LA AUTOMATIZACIÓN DE LOS PROCESOS RELACIONADOS CON LOS ACTOS ACADÉMICOS DE PREGRADO DE LA DIVISION DE CONTROL DE ESTUDIOS DE LA FACULTAD DE CIENCIAS.. TUTORÍA CONJUNTA CON LA PROFA. DI VASTA CONCETTINA. PREMIO MENCION HONORIFICA.	20
01-2007	15474128 - DE CASTRO GORETTI	(6430) DESARROLLO DE UNA APLICACIÓN WEB PARA LA DIVISIÓN DE CONTROL DE ESTUDIOS QUE AUTOMATICE LA ASIGNACIÓN DE HORARIOS Y AULAS EN EL PROCESO DE INSCRIPCIÓN DE ALUMNOS DE PREGRADO DE LA FACULTAD DE CIENCIAS.. TUTORÍA CONJUNTA CON LA PROFA. DI VASTA CONCETTINA. PREMIO MENCION HONORIFICA.	20
01-2007	16273188 - RIVERO KUNDRY	(6430) DESARROLLO DE UNA APLICACIÓN WEB PARA LA DIVISIÓN DE CONTROL DE ESTUDIOS QUE AUTOMATICE LA ASIGNACIÓN DE HORARIOS Y AULAS EN EL PROCESO DE INSCRIPCIÓN DE ALUMNOS DE PREGRADO DE LA FACULTAD DE CIENCIAS.. TUTORÍA CONJUNTA CON LA PROFA. DI VASTA CONCETTINA. PREMIO MENCION HONORIFICA.	20
01-2008	17711836 - MENDEZ NESTOR 17970342 - BOYER YURBELIS	(6430) AUTOMATIZACIÓN DE PROCESOS RELACIONADOS CON LAS SOLICITUDES ESTUDIANTILES Y ACTIVIDADES ADMINISTRATIVAS Y DE DOCENCIA DE LA FACULTAD DE CIENCIAS.. TUTORÍA CONJUNTA CON LA PROFA. DI VASTA CONCETTINA.	20
02-2008	15665008 - KLOPP ANJA 14200556 - FLORES DIEGO	(6420) DESARROLLO DE UNA APLICACIÓN WEB QUE PERMITE DETECTAR SIMILITUDES ENTRE DOCUMENTOS DIGITALES.. TUTORÍA CONJUNTA CON EL PROF. RIVAS SERGIO.	20
02-2009	17389408 - OJEDA LUCIA 16224834 - BELFORT PABLO	(6420) DESARROLLO DE LA APLICACIÓN DE GESTIÓN ACADÉMICA CONEST 2.0: MÓDULO ADMINISTRATIVO. TUTORÍA CONJUNTA CON EL PROF. RIVAS SERGIO. PREMIO MENCION HONORIFICA.	20
02-2009	17389408 - OJEDA LUCIA 16224834 - BELFORT PABLO	(6420) DESARROLLO DE LA APLICACIÓN DE GESTIÓN ACADÉMICA CONEST 2.0: MÓDULO ADMINISTRATIVO. TUTORÍA CONJUNTA CON EL PROF. RIVAS SERGIO. PREMIO MENCION HONORIFICA.	20
02-2010	17962467 - RUTIGLIANO JONATHAN 17529333 - PEREIRA JESUS	(6420) DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN DE HISTORIAS CLINICAS DIGITALES. TUTORÍA CONJUNTA CON EL PROF. SCALISE EUGENIO.	18
02-2010	17124125 - SIMON MARIANA 16033952 - MATA JESUS	(6420) DESARROLLO DE UNA APLICACIÓN WEB PARA LA AUTOMATIZACIÓN DE LA UNIDAD DE SERVICIO COMUNITARIO DE LA FACULTAD DE CIENCIAS. TUTORÍA CONJUNTA CON EL PROF. RIVAS SERGIO.	20
01-2011	14907279 - MORANTES GUSTAVO	(6430) DESARROLLO DE UNA APLICACIÓN WEB PARA MANEJAR LOS PROCESOS ADMINISTRATIVOS DE RESERVACIÓN DE LAS SALAS DE ENSAYO Y ESTUDIOS DE GRABACIÓN.CASO DE ESTUDIO:ROCK & FOLK. TUTORÍA CONJUNTA CON EL PROF. SANOJA ANDRÉS.	20
02-2011	15573230 - MOROS DANIEL	(6420) DESARROLLO DE UN MODULO DE ADMINISTRACIÓN DEL PROCESO DE GESTIÓN ACADÉMICA DE LOS CURSOS IMPARTIDOS POR LA ESCUELA DE IDIOMAS MODERNOS DE LA UNIVERSIDAD CENTRAL DE VENEZUELA.. TUTORÍA CONJUNTA CON EL PROF. RIVAS SERGIO.	19
02-2011	18249230 - DA MATA ANY	(6420) DESARROLLO DEL MÓDULO DE ESTADÍSTICAS DE GRADO PARA EL SISTEMA DE GESTIÓN ACADÉMICA ADMINISTRATIVA CONEST.. TUTORÍA CONJUNTA CON EL PROF. RIVAS SERGIO.	20
01-2012	19395140 - ARREAZA ALEXANDRA	(6420) DESARROLLO DE UN MÓDULO DE GESTIÓN Y PLANIFICACIÓN PARA LOS CURSOS DE EXTENSIÓN DE LA ESCUELA DE IDIOMAS MODERNOS DE LA FACULTAD DE HUMANIDADES Y EDUCACIÓN DE LA UCV.. TUTORÍA CONJUNTA CON EL PROF. RIVAS SERGIO. PREMIO MENCION HONORIFICA.	20
01-2012	19220025 - MANYOMA WILSON	(6420) DESARROLLO DE UN MÓDULO DE GESTIÓN Y PLANIFICACIÓN PARA LOS CURSOS DE EXTENSIÓN DE LA ESCUELA DE IDIOMAS MODERNOS DE LA FACULTAD DE HUMANIDADES Y EDUCACIÓN DE LA UCV.. TUTORÍA CONJUNTA CON EL PROF. RIVAS SERGIO. PREMIO MENCION HONORIFICA.	20

Anexo I - Constancia de tutoría para seminario



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
COORDINACIÓN ACADÉMICA
DIVISIÓN DE CONTROL DE ESTUDIOS



CONSTANCIA DE TUTORÍA DE SEMINARIOS

QUIEN SUSCRIBE, JEFE DE LA DIVISIÓN DE CONTROL DE ESTUDIOS DE LA FACULTAD DE CIENCIAS DE LA UNIVERSIDAD CENTRAL DE VENEZUELA, POR MEDIO DE LA PRESENTE HACE CONSTAR QUE LA PROFA. **ZAMBRANO JOSSIE**, PORTADORA DE LA CÉDULA DE IDENTIDAD N° **13872341**, ADSCRITA A LA ESCUELA DE COMPUTACION, HA PARTICIPADO COMO TUTOR DE LOS SIGUIENTES SEMINARIOS.

PERIODO	AUTOR	SEMINARIO	NOTA
01-2008	18031704 - HERGUETA RICARDO	(6419) SEMINARIO (APLICACIONES INTERNET)	20
01-2008	15665008 - KLOPP ANJA	(6419) SEMINARIO (APLICACIONES INTERNET)	20
02-2008	17160506 - KHAWAM GEORGE	(6419) SEMINARIO (APLICACIONES INTERNET)	20
01-2009	16224834 - BELFORT PABLO	(6419) SEMINARIO (APLICACIONES INTERNET)	20
01-2009	17389408 - OJEDA LUCIA	(6419) SEMINARIO (APLICACIONES INTERNET)	20
02-2009	17124125 - SIMON MARIANA	(6419) SEMINARIO (APLICACIONES INTERNET)	20
02-2009	16033952 - MATA JESUS	(6419) SEMINARIO (APLICACIONES INTERNET)	20
01-2010	17962467 - RUTIGLIANO JONATHAN	(6419) SEMINARIO (APLICACIONES INTERNET)	20
01-2010	17529333 - PEREIRA JESUS	(6419) SEMINARIO (APLICACIONES INTERNET)	20
01-2011	16179441 - HERNANDEZ JOSE	(6419) SEMINARIO (APLICACIONES INTERNET)	19
02-2012	19023671 - COLMENARES DANIELA	(6419) SEMINARIO (APLICACIONES INTERNET)	20
01-2013	18761047 - DIAZ SARAI	(6419) SEMINARIO (APLICACIONES INTERNET)	20
01-2013	18738592 - ROSALES ROGER	(6419) SEMINARIO (APLICACIONES INTERNET)	20
01-2013	19739906 - CARNEIRO YSABELLA	(6419) SEMINARIO (APLICACIONES INTERNET)	20
01-2013	15805976 - PEREZ GABRIEL	(6419) SEMINARIO (APLICACIONES INTERNET)	20
01-2013	19018395 - MONTILLA JOSEFINA	(6419) SEMINARIO (APLICACIONES INTERNET)	20

CONSTANCIA QUE SE EXPIDE A SOLICITUD DE LA PARTE INTERESADA EN CARACAS, A LOS 12 DÍAS DEL MES DE AGOSTO DE 2014.

PROF. DAMARIS BARRANTES
JEFA DE DIVISIÓN DE CONTROL DE ESTUDIOS

Anexo J - Extracto de código de detección del tipo de solicitud

```

def validar_tipo_smi_y_ejecutar_reglas(mp) # mp = materia_en_plan
  arreglo_respuestas = []
  # Detectar extracreditos
  creditos = recalculacion_creditos
  if creditos + mp.nro_creditos > max_creditos_por_periodo
    if materias_exceso_creditos.size + 1 > numero_extracreditos_permitidos
      # Retornar error: numero extracreditos alcanzado
    else
      # Ejecutar reglas Extracredito
      respuesta = gestor.ejecutar_reglas(TipoSolicitud::EXTRACREDITO)
      arreglo_respuestas << respuesta

      # Detectar si la asignatura también es Paralelo
      genera_paralelo, nro_paralelos_generados = materia_genera_paralelo?(materia_codigo)
      if genera_paralelo
        if numero_paralelos_solicitados + 1 > numero_paralelos_permitidos
          # Retornar error: numero paralelos alcanzado
        else
          # Ejecutar reglas
          respuesta = gestor.ejecutar_reglas(TipoSolicitud::PARALELO)
          # Almacenar respuesta
        end
      end
      # Verificar si es Extemporanea
      if !extemporanea_materias_no_permitidas_tipo.include?(mp.tipo_materia_id)
        # Ejecutar reglas si el tipo de materia Extemporanea esta permitida
        respuesta = gestor.ejecutar_reglas(TipoSolicitud::EXTEMPORANEA)
        arreglo_respuestas << respuesta
      end
    end
  else
    # No es Extracredito, detectar si es Paralelo
    if genera_paralelo
      # Ejecutar codigo de paralelos
    else
      # Si no es extracredito ni paralelo, es Extemporanea
      if !genera_paralelo &&
        extemporanea_materias_no_permitidas_tipo.include?(mp.tipo_materia_id)
          # Retornar error: tipo de asignatura extemporanea no permitida
        else
          # Ejecutar reglas
          respuesta = gestor.ejecutar_reglas(TipoSolicitud::EXTEMPORANEA)
          arreglo_respuestas << respuesta
        end
      end
    end
  end
  return arreglo_respuestas # Contiene toda la info de las solicitudes
end

```

Anexo K - Extracto de código del gestor de reglas

```

def ejecutar_reglas(tipo_smi_id) # Tipo de solicitud
  gestor = self
  errores = []

  # Obtener reglas de ejecucion obligatoria por escuela y por tipo de solicitud
  reglas_escuela = ReglaSmi.where(
    carrera_id: @organizacion_id, tipo_smi_id: tipo_smi_id,
    tipo_status_regla_id: TipoStatusReglaSmi::ACTIVA,
    tipo_regla_id: TipoReglaSmi::OBLIGATORIA).order(orden: :asc)

  reglas_escuela.each do |regla|
    metodo = regla.condicion_smi.nombre
    if regla.valor ==
      # Determinar tipo de parametro
    else
      parametros = [] # el parametro es nil
    end

    # Validar que el metodo de la regla exista y reciba los N parametros obligatorios
    if gestor.methods.include?(metodo.to_sym) &&
      gestor.method(metodo.to_sym).arity == parametros.size

      # Ejecutar regla y obtener resultado (bool)
      se_cumple_regla = gestor.send(metodo, *parametros)
      if eval(se_cumple_regla.to_s)
        # Se cumple la regla
      else
        # Obtener articulo y almacenarlo en arreglo de errores
      end
    else
      # Error: No se encontro el metodo
    end
  end#reglas

  # Emitir sugerencias dependiendo del numero de errores
  case errores.size
  when 0 == # Se sugiere la aprobacion
  when 1 == # Para los demas casos se sugiere el rechazo
  else ==
  end
end

```

Anexo L - Planilla de solicitud de modificación de inscripción



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
DIVISIÓN DE CONTROL DE ESTUDIOS



SOLICITUD DE MODIFICACIÓN DE INSCRIPCIÓN

*** ORIGINAL EXPENDIENTE ***

FECHA: 25/08/2014
PERÍODO LECTIVO: **01-2014**
LICENCIATURA: **COMPUTACION**

ESTUDIANTE:

INSCRIPCIÓN EXTEMPORÁNEA: (6213) MINERIA DE DATOS		
SE SUGIERE SU APROBACIÓN. RAZONES: CUMPLE CON LOS REQUISITOS		
AVAL DEL DOCENTE		
CI:	NOMBRE:	FIRMA:
ESPACIO RESERVADO PARA LA UNIDAD DE ASESORAMIENTO ACADÉMICO		
APROBAR <input type="checkbox"/>		RECHAZAR <input type="checkbox"/>
ESPACIO RESERVADO PARA EL CONSEJO DE ESCUELA		
APROBAR <input type="checkbox"/>		RECHAZAR <input type="checkbox"/>

EXTRACRÉDITO, INSCRIPCIÓN EXTEMPORÁNEA: (6344) PLAN. ESTRATEG. DE SISTEMAS		
SE SUGIERE SU RECHAZO. RAZONES: EL ESTUDIANTE QUE SOLICITE EXCESO DE CRÉDITOS DEBE HABER APROBADO AL MENOS TRES MATERIAS EN CADA UNO DE LOS DOS SEMESTRES ANTERIORES AL DE LA SOLICITUD. EN CASO DE QUE EL SEMESTRE PREVIO SEA UN INTENSIVO SE CONTABILIZARÁ LA MATERIA APROBADA AL SEMESTRE PREVIO REGULAR.		
AVAL DEL DOCENTE		
CI:	NOMBRE:	FIRMA:
ESPACIO RESERVADO PARA LA UNIDAD DE ASESORAMIENTO ACADÉMICO		
APROBAR <input type="checkbox"/>		RECHAZAR <input type="checkbox"/>
ESPACIO RESERVADO PARA EL CONSEJO DE ESCUELA		
APROBAR <input type="checkbox"/>		RECHAZAR <input type="checkbox"/>

JUSTIFICACIÓN DEL ESTUDIANTE	
FUI RECHAZADO EN LA PRESCRIPCIÓN, SIN EMBARGO DESPUÉS DE LA INSCRIPCIÓN ,QUEDARON CUPOS LIBRES. COMO QUIERO IR POR LA RAMA DE BASES DE DATOS Y SISTEMAS DE INFORMACIÓN ,ME PARECIÓ BUENA IDEA INSCRIBIR ESTA MATERIA. ADICIONALMENTE, EL SEMESTRE PASADO, SE ME RECHAZARON TODAS LAS PRE INSCRIPCIONES DE LA RAMA DE BASES DE DATOS Y SISTEMAS DE INFORMACIÓN QUE ME INTERESABAN, LO QUE ME "CONDUJO" A HACER LAS PCI Y LA PASANTIA, NO QUISIERA QUE ME PASARA ESO DE NUEVO.	

HISTORIAL DE SOLICITUDES			
PERÍODO	TIPO	ASIGNATURA	ESTADO
02-2011	PARALELO	(6108) MATEMATICAS DISCRETAS III y (8207) MATEMATICA II	APROBADA

Anexo M - Planilla de solicitud de reincorporación



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
DIVISIÓN DE CONTROL DE ESTUDIOS



SOLICITUD DE REINCORPORACIÓN

*** ORIGINAL EXPENDIENTE ***

FECHA: 25/08/2014
PERÍODO LECTIVO: **01-2014**
LICENCIATURA: **COMPUTACION**

ESTUDIANTE:

ÚLTIMO PERÍODO CURSADO: **01-2013**

CONDICIÓN EVALUADA SEGÚN NORMATIVA INTERNA DE LA FACULTAD DE CIENCIAS	OBSERVACIONES
<p>1. SE PODRÁN REINCORPORAR A LA FACULTAD DE CIENCIAS SIN QUE SE REQUIERA UN ESTUDIO DEL CASO POR LA UNIDAD DE ASESORAMIENTO ACADÉMICO:</p> <p>1.1. LOS BACHILLERES QUE, INCURSOS EN EL ARTÍCULO N° 6 DE LAS NORMAS SOBRE RENDIMIENTO MÍNIMO Y CONDICIONES DE PERMANENCIA DE LOS ALUMNOS EN LA UCV, SOLICITEN REINCORPORARSE EN EL PERÍODO INMEDIATAMENTE SIGUIENTE DE HABERSE CUMPLIDO LA SANCIÓN CORRESPONDIENTE.</p> <p>1.2. LOS BACHILLERES QUE CUMPLAN LAS SIGUIENTES CINCO (5) CONDICIONES:</p> <p>1.2. a) TENER APROBADO AL MENOS EL 20% DE LAS UNIDADES DE LA LICENCIATURA CORRESPONDIENTE.</p> <p>1.2. b) TENER UN ÍNDICE DE EFICIENCIA MAYOR A 0,40.</p> <p>1.2. c) NO HABERSE RETIRADO Y REINCORPORADO ANTERIORMENTE EN MÁS DE UNA OPORTUNIDAD.</p> <p>1.2. d) ESTAR DESINCORPORADO MENOS DE 5 AÑOS.</p> <p>1.2. e) NO HABER CURSADO MÁS DE 15 PERÍODOS EN LA ESCUELA A LA CUAL SOLICITA REINCORPORACIÓN.</p> <p>1.3. LOS BACHILLERES A QUIENES EL CONSEJO DE ESCUELA RESPECTIVO LES HAYA APROBADO RETIRARSE POR UN TIEMPO DEFINIDO, PARA LUEGO REINCORPORARSE EN UN SEMESTRE DETERMINADO.</p> <p>2. LOS CASOS QUE NO ESTÉN CONTEMPLADOS EN LA DISPOSICIÓN ANTERIOR, SERÁN ESTUDIADOS POR LA UNIDAD DE ASESORAMIENTO ACADÉMICO, LA CUAL HARÁ LA RECOMENDACIÓN RAZONADA QUE CONSIDERE PERTINENTE EN CADA CASO, ANTE EL CONSEJO DE ESCUELA RESPECTIVO.</p>	<p>EL ESTUDIANTE NO ESTÁ INCURSO EN ARTICULO 6. ÚLTIMO PERIODO CURSADO: 01-2013</p> <p>4/(38 UC - 20%) U.C APROBADAS</p> <p>EFICIENCIA: 0.25</p> <p>REINCORPORACIONES: 0</p> <p>AÑOS DESINCORPORADO: 0</p> <p>PERIODOS CURSADOS: 2</p>

RESOLUCIÓN	
APROBAR <input type="checkbox"/>	RECHAZAR <input type="checkbox"/>

BR.

FECHA: ___/___/___

FUNCIONARIO RECEPTOR

DIVISIÓN DE CONTROL DE ESTUDIOS

FECHA: ___/___/___

FIRMA DEL DECANO O DIRECTOR

FECHA: ___/___/___