



Universidad Central de Venezuela

Facultad de Ciencias

Escuela de Computación

Automatización de las guías de diseño
del ambiente de desarrollo del sistema CONEST

Trabajo Especial de Grado
presentando ante la ilustre
Universidad Central de Venezuela
Por la Bachiller:
Froiransi Verónica Camico Zambrano
C.I 18.751.667
Tutor: Prof. Sergio Rivas

Caracas, OCTUBRE 2014

Resumen

La gestión académica de la Facultad de Ciencias de la Universidad Central de Venezuela se lleva a cabo a través del sistema CONEST, el cual permite la automatización de la mayoría de los procesos gestionados por la División de Control de Estudios. El desarrollo de los módulos que lo componen se ha incrementado aceleradamente en los últimos años, sobrepasando el alcance del proyecto original de este sistema y generando inconsistencias en sus interfaces de usuario. Por esta razón, el grupo de desarrollo decidió crear una nueva versión de este sistema, que incorpora nuevas tecnologías y que cumple con estándares en sus interfaces de usuario; además, permitan que la evolución y el crecimiento de este sistema se realicen de manera organizada. Este trabajo de investigación se centra en el desarrollo de guías formales de diseño que permitan establecer los estándares de creación de interfaces de usuario del sistema CONEST, por medio de la creación de los patrones de interacción asociados a los componentes de interfaz de usuario del sistema. Esta guía de diseño se desarrolló a través de la aplicación de la metodología ágil de desarrollo de software AgilUs, la cual se basa en la evaluación de la usabilidad y la incorporación de los usuarios finales durante el proceso de desarrollo del software. El sistema CONEST Desarrollo, implementado en este trabajo, surge debido a las tendencias actuales que proponen la utilización de guías de estilo automatizadas con el fin de reducir los costos y minimizar el tiempo requerido para la creación de nuevas aplicaciones Web, aumentando la satisfacción del usuario y promoviendo la reutilización de componentes de interfaz de usuario.

Palabras clave: Gestión académica, AgilUs, Desarrollo de software, Patrones de Interacción, Sistema de gestión académica CONEST, Componentes, Interfaz de usuario, Guía de diseño.

Froiransi V. Camico Z. Correo electrónico: froiransi@gmail.com

Sergio J. Rivas A. Correo electrónico: sergiorivas@gmail.com

Índice general

Introducción	10
1. Metodología AgilUs y Tecnologías Web	14
1.1. Metodología de desarrollo AgilUs	14
1.2. Tecnologías Web	21
1.2.1. HTML	22
1.2.2. CSS	22
1.2.3. JavaScript	23
1.2.4. Ruby on Rails	24
1.2.5. MySQL	28
2. Análisis de las interfaces de usuario de CONEST	29
2.1. Organización de las IU del sistema CONEST	31
2.1.1. Componentes generales	31
2.1.2. Módulo estudiante	34
2.1.3. Módulo docente	38
2.1.4. Módulo administrativo	41
2.2. Creación de patrones de interacción	49

3. Implementación del sistema CONEST Desarrollo	63
3.1. Fase I: Requerimientos	64
3.1.1. Tormenta de ideas	64
3.1.2. Evaluación de sistemas existentes	65
3.1.3. Perfiles de usuario	73
3.1.4. Requerimientos funcionales y no funcionales	73
3.2. Fase II: Análisis	74
3.2.1. Modelo de casos de uso	74
3.2.2. Modelo de objetos del dominio	76
3.2.3. Prototipo en papel	77
3.2.4. Guía de estilo	80
3.2.5. Patrones de interacción	82
3.3. Fase III: Protipaje	91
3.3.1. Iteración 1: Funcionalidades de gestión	92
3.3.2. Iteración 2: Funcionalidades de consulta	96
3.4. Fase IV: Entrega	96
3.4.1. Pruebas de aceptación	96
Conclusiones	98
Referencias Bibliográficas	101

Índice de figuras

1.	Proyecto CONEST Desarrollo	12
1.1.	Metodología AgilUs	15
1.2.	Etapa de Requisitos	16
1.3.	Etapa de Análisis	17
1.4.	Etapa de Prototipaje	19
1.5.	Etapa de Entrega	20
1.6.	Arquitectura cliente/servidor	21
1.7.	Arquitectura MVC	26
1.8.	Componentes de Ruby on Rails	27
2.1.	Iteraciones del proceso de organización de IU	30
2.2.	Componente general: Menú Principal	32
2.3.	Componente general: Directorio actual	32
2.4.	Componente general: Perfil de usuario	32
2.5.	Componente general: Widget	33
2.6.	Componente general: Consultar período académico	33
2.7.	Módulo estudiante: Carga académica actual	34
2.8.	Módulo estudiante: Realizar preinscripción	34

ÍNDICE DE FIGURAS

2.9. Módulo estudiante: Agregar asignatura a preinscripción	35
2.10. Módulo estudiante: Eliminar asignatura de preinscripción	35
2.11. Módulo estudiante: Consultar requisitos de asignatura	36
2.12. Módulo estudiante: Horario parcial	36
2.13. Módulo estudiante: Seleccionar sección en asignatura	37
2.14. Módulo estudiante: Formalizar preinscripción	37
2.15. Módulo estudiante: Retirar asignatura	37
2.16. Módulo estudiante: Horario del período académico actual	38
2.17. Módulo docente: Realizar calificación	39
2.18. Módulo docente: Consultar solicitudes de modificación de inscripción .	39
2.19. Módulo docente: Solicitar jurado para Trabajo Especial de Grado . . .	39
2.20. Módulo docente: Consultar solicitudes de jurado para Trabajo Especial de Grado	40
2.21. Módulo docente: Consultar planilla de notas de Trabajo Especial de Grado	40
2.22. Módulo docente: Formalizar Calificación	41
2.23. Módulo administrativo: Crear evento en el calendario	41
2.24. Módulo administrativo: Agregar período académico	42
2.25. Módulo administrativo: Gestionar planificación del período académico .	42
2.26. Módulo administrativo: Búsqueda rápida	43
2.27. Módulo administrativo: Agregar asignatura a período académico	43
2.28. Módulo administrativo: Actualizar disponibilidad de asignatura	44
2.29. Módulo administrativo: Actualizar requisito de preinscripción de asignatura	44
2.30. Módulo administrativo: Generar mapa de aulas	45
2.31. Módulo administrativo: Modificar horario de asignatura	45

ÍNDICE DE FIGURAS

2.32. Módulo administrativo: Anular solicitud de jurado de Trabajo Especial de Grado	46
2.33. Módulo administrativo: Aprobar solicitud de jurado para Trabajo Especial de Grado	46
2.34. Módulo administrativo: Resumen de solicitud	47
2.35. Módulo administrativo: Gestionar solicitudes de modificación de inscripción	47
2.36. Módulo administrativo: Gestionar estudiante	48
2.37. Módulo administrativo: Gestionar proceso de preinscripción	49
2.38. Taxonomía de los patrones de interacción	50
2.39. Iteraciones de la construcción del catálogo de componentes de IU	51
2.40. Catálogo de patrones del sistema CONEST	52
2.41. Principal estudiante	53
2.42. Gestionar de planes de estudio	53
2.43. Principal docente	53
2.44. Gestionar procesos	54
2.45. Gestionar aulas	54
2.46. Gestionar planillas de notas	55
2.47. Principal administrativo	55
2.48. Gestionar modificación de inscripción	55
2.49. Gestionar solicitud jurado TEG	56
2.50. Gestionar planificación oferta académica	57
2.51. Gestionar entidad	58
3.1. Iteraciones de la implementación del sistema CONEST Desarrollo	63
3.2. Tormenta de ideas de CONEST Desarrollo	64

ÍNDICE DE FIGURAS

3.3. Modelo de casos de uso CONEST Desarrollo	75
3.4. Modelo de objetos del dominio CONEST Desarrollo	77
3.5. Prototipo en papel - Inicio sesión	78
3.6. Prototipo en papel - Guía de inicio	79
3.7. Prototipo en papel - Componentes	79
3.8. Prototipo en papel - Gestión de usuarios	80
3.9. Lenguaje de patrones de interacción CONEST Desarrollo	82
3.10. Ambiente de desarrollo	92
3.11. Prueba de aceptación CONEST Desarrollo	97

Índice de tablas

2.1. Patrón de interacción - Gestión académica	58
2.2. Patrón de interacción - Estudiante	59
2.3. Patrón de interacción - Rol de usuario	59
2.4. Patrón de interacción - CONEST	60
2.5. Patrón de interacción - Principal estudiante	61
2.6. Patrón de interacción - Consultar usuario	62
3.1. Evaluación de sistemas existentes: Android Design	66
3.2. Evaluación de sistemas existentes: iOS Human Interface Guideline	67
3.3. Evaluación de sistemas existentes: Google Design	68
3.4. Evaluación de sistemas existentes: Bootstrap	69
3.5. Evaluación de sistemas existentes: Foundation	70
3.6. Evaluación de sistemas existentes: Kube	71
3.7. Cuadro comparativo de sistemas existentes	72
3.8. Guía de estilo de CONEST Desarrollo	81
3.9. Patrón de interacción - Guías de estilo automatizadas	82
3.10. Patrón de interacción - Desarrollador	83
3.11. Patrón de interacción - Administrador	83

ÍNDICE DE TABLAS

3.12. Patrón de interacción - Recordar contraseña	83
3.13. Patrón de interacción - CONEST Desarrollo	84
3.14. Patrón de interacción - Consultar guía de estilo	85
3.15. Patrón de interacción - Realizar búsqueda	86
3.16. Patrón de interacción - Eliminar entidad	87
3.17. Patrón de interacción - Consultar guía de inicio	88
3.18. Patrón de interacción - Consultar catálogo de componentes	89
3.19. Patrón de interacción - Consultar componente	90
3.20. Patrón de interacción - Gestionar entidad	91
3.21. Metáforas de la aplicación CONEST Desarrollo	93
3.22. Resultados de la evaluación Heurística	95

Introducción

Las tecnologías de información y comunicación, a partir del auge del Internet, han automatizado una gran cantidad de actividades y procesos cotidianos. Las aplicaciones Web se han convertido en la opción más utilizada por las organizaciones, debido a que el uso del Internet hace posible que la comunicación sea rápida y sencilla. Las aplicaciones Web permiten disminuir los costos en tiempo, materiales y recursos humanos necesarios para gestionar los procesos existentes dentro de estas organizaciones. Alrededor del mundo son innumerables las organizaciones que utilizan las aplicaciones Web para gestionar sus operaciones, desde Instituciones Bancarias hasta Universidades.

La Universidad Central de Venezuela (UCV) es una de las más grandes e importantes universidades de Venezuela, fue fundada el 22 de diciembre del año 1721 y desde el año 1953 se encuentra ubicada en la Ciudad Universitaria de Caracas. En la actualidad, mantiene una matrícula estudiantil estimada de 54000 estudiantes de pregrado, 8600 docentes, 3000 empleados administrativos y 1000 obreros aproximadamente, distribuidos en 11 Facultades. En las facultades se llevan a cabo las actividades más importantes de los procesos de Gestión Académica y Administrativa. Este Trabajo Especial de Grado se enmarca en la Gestión Académica de pregrado de la Facultad de Ciencias, la cual esta conformada por las licenciaturas de: Biología, Computación, Física, Geoquímica, Matemática y Química.

La División de Control de Estudios (DCE) de la Facultad de Ciencias, es la unidad encargada de llevar a cabo la administración y gestión de la información relacionada con los procesos académicos relacionados al personal docente, estudiantes y su personal administrativo que conforma la comunidad, además de relacionarse con las dependencias internas y externas, llevando a cabo procesos y actividades tales como: preinscripción, inscripción, calificación, registro y control de grado, emisión de constancias, entre otros.

A través del sistema CONEST, la DCE ha logrado automatizar distintos procesos y actividades, brindando respuestas rápidas y la mejora de los servicios prestados. Este sistema se encuentra operativo desde el año 2007 y ha sido desarrollado por la comunidad de la Escuela de Computación, a través de la participación de diferentes estudiantes y profesores, por medio de la realización de numerosas actividades académicas tales como: pasantías, proyectos, seminarios y trabajos especiales de grado.

INTRODUCCIÓN

CONEST originalmente fue diseñado para la gestión de los tres principales procesos llevados a cabo por esta división: Inscripción, Calificación y Graduación. A través del tiempo se incorporaron nuevas funcionalidades y procesos, lo que conllevó a que este sistema creciera de manera desorganizada y sin un conjunto de estándares que permitieran definir los lineamientos al momento de crear nuevas funcionalidades, generando de esta manera, un número significativo de inconsistencias en sus interfaces de usuario.

En el año 2013, el grupo de desarrollo de CONEST decide crear una nueva versión del sistema, incorporando nuevas tecnologías y aplicando los conocimientos y la madurez adquirida del desarrollo de la versión inicial, promoviendo de esta manera una reorganización estructural de este proyecto y anticipando los cambios que pueden surgir en las funcionalidades existentes. Por esta razón, surge la necesidad de definir el conjunto de lineamientos que deben seguirse a la hora de desarrollar nuevas interfaces de usuario.

En general, los diseñadores con experiencia no resuelven la mayoría de los problemas nuevos desde cero, es común reusar soluciones que han aplicado con éxito anteriormente y que han formado parte de su experiencia. Este conocimiento de los expertos o especialistas puede traducirse en componentes que pueden ser utilizados por otras personas en la resolución de problemas similares. Una forma de registrar este conocimiento es a través de la implementación de patrones de interacción, permitiendo a los diseñadores reducir costos y tiempo de desarrollo de software, al tiempo que se crean soluciones a problemas en el diseño de interfaz de usuario recurrentes, como la redundancia e inconsistencia de información. En el caso del sistema de gestión académica CONEST, se ha observado que la naturaleza de los equipos de desarrollo suelen cambiar de manera frecuente, ya que en general las actividades y proyectos que permiten la implementación de nuevas funcionalidades en el sistema suelen ser cortos, lo que implica un diseño poco planificado y que carece de estándares para cada uno de los componentes de interfaz de usuario que ofrece el sistema, trayendo como consecuencia interfaces de usuario inconsistentes, repetición de código, nombres e interacciones diferentes para tareas similares, errores en la aplicación de reglas y respuestas inesperadas.

Adicionalmente, en la actualidad existen diferentes tendencias en las organizaciones que llena a cabo la producción de software, las cuales incorporan guías de diseño automatizadas que facilitan a los desarrolladores el desarrollo de módulos e interfaces de usuario dentro de una aplicación. Estas guías de diseño permiten reducir costos de tiempo y recursos en la consecución de los proyectos dentro de las organizaciones, permitiendo incrementar la satisfacción de los clientes y promoviendo el reuso de soluciones existentes.

En vista de la problemática existente y la tendencia a la utilización de las guías de diseño, surge esta investigación que tiene como objetivo general “Automatizar las guías de diseño del ambiente de desarrollo del sistema CONEST”.

INTRODUCCIÓN

Los objetivos específicos que se derivan son:

- Realizar el estudio de herramientas que proveen componentes de interfaz de usuario, a fin de reusar las mejores prácticas implementadas en estas.
- Realizar el estudio de las interfaces de usuario del sistema CONEST, con el fin de clasificar los componentes de interfaz de usuario.
- Definir un catálogo de componentes de interfaz de usuario, basados en patrones de interacción enriquecidos, que definan los estándares de los componentes de interfaz de usuario del sistema de gestión académica CONEST.
- Desarrollar una aplicación Web, para la gestión del catálogo de componentes y los estándares de interfaz de usuario del sistema de gestión académica CONEST aplicando el método de desarrollo ágil AgilUs. .

La aplicación implementada en esta investigación se denomina CONEST Desarrollo e incorpora diferentes herramientas que resultan útiles para el desarrollo de un sistema Web, en la Figura 1 pueden observarse las herramientas que se contemplan dentro de este proyecto. Dada la importancia del sistema CONEST en la comunidad de la Facultad de Ciencias y la cantidad de personas que lo utilizan (aproximadamente 4000), se estableció como prioridad implementar aquellas funcionalidades relacionadas directamente con las interfaces de usuario de un sistema, las cuales son guías de instalación, guías de estilo y componentes de interfaz de usuario.



Figura 1: *Proyecto CONEST Desarrollo*

INTRODUCCIÓN

Los módulos del sistema CONEST Desarrollo implementados en este trabajo incluyen: un catálogo de componentes de interfaz de usuario descritos mediante patrones de interacción enriquecidos, la guía de estilo del sistema CONEST, estándares para la apariencia de las interfaces de usuario y una guía de instalación del ambiente del sistema CONEST, el cual provee los pasos necesarios para la instalación del sistema operativo, manejadores de versiones, el *framework* de programación *Ruby on Rails*, entre otros.

Para llevar a cabo esta investigación, fue necesario realizar un análisis detallado de las interfaces de usuario del sistema CONEST, con el fin de obtener el catálogo de componentes de interfaz de usuario de este sistema descritos mediante patrones de interacción enriquecidos, ya que son el insumo principal de la implementación de la solución propuesta a la problemática existente. Adicionalmente es importante conocer todas aquellas tecnologías y metodología de desarrollo de software que permitan llevar a cabo de manera exitosa la implementación de la solución propuesta en esta investigación. Por esta razón el presente Trabajo Especial de Grado se encuentra estructurado de la siguiente manera:

El Capítulo 1 “Metodología de desarrollo AgilUs y Tecnologías Web”: describe la metodología AgilUs y las tecnologías relacionadas con el desarrollo de aplicaciones Web, tales como HTML, CSS, JavaScript, Ruby on Rails y MySQL, las cuales conforman la plataforma del sistema CONEST.

El Capítulo 2 “Análisis de las interfaces de usuario del sistema CONEST”: describe el estudio realizado a las interfaces de usuario del sistema CONEST y las actividades llevadas a cabo durante este análisis. Iniciando con la definición del esquema de clasificación de interfaces de usuario, seguidamente se describe en detalle esta clasificación y por último se enuncian los patrones interacción, los cuales permitirán llevar a cabo la construcción del catálogo de componentes de interfaz de usuario del sistema CONEST.

El Capítulo 3 “Implementación del sistema CONEST Desarrollo”: describe el proceso de desarrollo de este sistema a través de la metodología AgilUs, el cual incorpora diferentes fases y actividades de este proceso.

Seguidamente se presentan las conclusiones y recomendaciones derivadas de este trabajo, las cuales contemplan los objetivos y experiencias alcanzadas a través de esta investigación.

Finalmente, se presentan las referencias bibliográficas, las cuales sustentan las bases conceptuales expuestas a lo largo de esta investigación.

Capítulo 1 Metodología de desarrollo AgilUs y tecnologías Web

En este capítulo se describe el proceso de desarrollo del sistema CONEST Desarrollo aplicando de la metodología AgilUs y se describen las tecnologías de la plataforma del sistema CONEST, y aquellas que pueden ser utilizadas para la implementación del sistema CONEST Desarrollo. En la primera sección se describen las fases del proceso de desarrollo y los artefactos propuestos por la metodología AgilUs. En la segunda sección, se describen las tecnologías del lado del cliente (HTML, CSS, JavaScript y AJAX) y las tecnologías del lado del servidor *Ruby on Rails* y *MySQL*.

1.1. Metodología de desarrollo AgilUs

La Ingeniería de Software es una disciplina que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema (Sommerville, 2005). Esta disciplina se apoya en la gestión de la calidad del software y se fundamenta en el proceso de desarrollo de software, el cual consiste en la aplicación de diferentes métodos que incluyen actividades descriptivas y de modelado que permiten realizar las tareas de análisis, diseño y construcción del software.

El proceso de desarrollo de software tiene como propósito la producción de un producto de software eficaz y eficiente que reúna los requisitos del cliente, y comúnmente son denominados metodologías para el desarrollo de software.

En el caso de esta investigación se utilizó la metodología ágil de desarrollo AgilUs, la cual es el resultado de una de las líneas de investigación desarrolladas en el Centro de Ingeniería de Software y Sistemas (ISYS) de la Escuela de Computación, Universidad Central de Venezuela (Acosta, 2013). AgilUs se fundamenta en el análisis centrado en el usuario y en la participación de especialistas, con el objetivo de evolucionar el software, a fin que éste alcance el mayor grado de usabilidad una vez culminado su desarrollo.

La Metodología AgilUs busca proporcionar un conjunto de actividades organizadas para construir la usabilidad en el diseño de interfaces de usuario durante el desarrollo de un

producto de software. El proceso de desarrollo de software engloba las actividades de requerimientos, análisis, prototipaje y entrega; así como las evaluaciones de usabilidad correspondientes a cada etapa del proceso. Se realizan en ciclos iterativos hasta alcanzar el producto final. En cada etapa del proceso de desarrollo de software, se incluyen actividades propias para la construcción de la usabilidad.

AgilUs integra las disciplinas de la Interacción Humano-Computador y la Ingeniería de Software, promoviendo el diseño centrado en el usuario, e impactando positivamente la calidad del software.

El ciclo de vida de AgilUs hace énfasis en la importancia del usuario y sus evaluaciones. Está basado en el desarrollo iterativo e incremental de prototipos de alta fidelidad hasta que se convierten en el producto final para entrega. Este producto final puede ser posteriormente modificado a través de un mantenimiento correctivo y/o evolutivo, que no está contemplado como parte de la metodología. En cada fase del desarrollo se incluyen actividades para la construcción de la usabilidad. Se busca proporcionar una manera de proceder organizadamente para construir la usabilidad durante el desarrollo de un producto. El ciclo de vida engloba la definición de requisitos, análisis, prototipaje y entrega. La Figura 1.1 muestra la relación entre cada etapa del ciclo de vida de AgilUsy los artefactos propuestos para cada una de ellas.



Figura 1.1: Metodología AgilUs

A continuación se describe las etapas de esta metodología y los artefactos elaborados en cada una de éstas:

1. **Requisitos:** Se realiza el análisis global del problema a solucionar, se estudian productos similares existentes, se genera el perfil de usuario, y se define la lista de requerimientos a desarrollar. Esta etapa es importante en el desarrollo del software, ya que un mal análisis de requisitos traería como consecuencia un software que no cumple con las necesidades e incluso con las expectativas del usuario. Los artefactos propuestos por la metodología en la etapa de requisitos son: tormenta de ideas, encuestas, entrevistas, cuestionarios, evaluación de sistemas existentes, perfiles de usuario, y requerimientos funcionales y no funcionales (Ver Figura1.2), los cuales se describen a continuación:

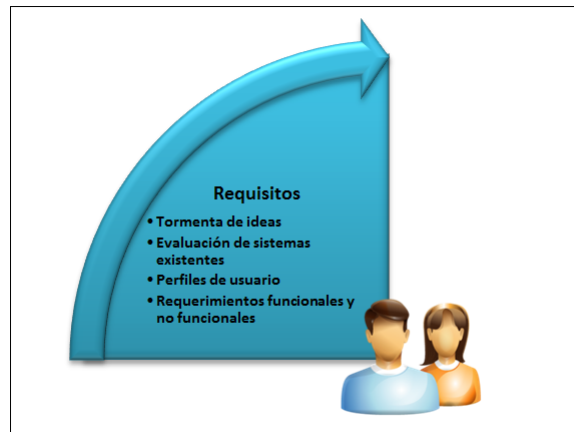


Figura 1.2: Etapa de Requisitos

- **Tormenta de ideas:** la técnica denominada tormenta de ideas consiste en la generación de ideas por parte de un grupo multidisciplinario, liberando la mente para proponer cualquier idea, permitiendo, así, la libertad para la creatividad. En el caso del proceso de desarrollo de software, corresponde al equipo de desarrollo realizar esta técnica, incorporando al usuario final. Es recomendable aplicar en las etapas tempranas del proceso de desarrollo, cuando se conoce poco acerca del diseño real y hay necesidad de nuevas ideas, a fin de determinar los requerimientos de usabilidad del sistema a desarrollar.
- **Evaluación de sistemas existentes:** consiste en la revisión de versiones anteriores del mismo sistema, así como sistemas de la competencia o afines, con el objetivo de identificar problemas de usabilidad. Esto permite, prevenir estos problemas en el sistema que se construye y obtener una medida de base para la usabilidad del mismo. Entre los principales beneficios de esta técnica se tiene que permite identificar problemas que podrán prevenirse en el nuevo

sistema y provee medidas de efectividad, eficiencia y satisfacción que pueden ser usadas como base para el mismo.

- Perfiles de usuario: los perfiles de usuario describen a los usuarios previstos del sistema y se detallan las características relevantes en un estudio de usuarios en cada perfil. De esta forma, se centra la deducción de requisitos en aquello que más útil puede resultar para el diseño de un sistema con un nivel de usabilidad adecuado.
 - Requerimientos funcionales y no funcionales: los requerimientos de un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema. Los requerimientos funcionales son aquellos servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. Los requerimientos no funcionales son aquellos que no se refieren a las funciones específicas del sistema, sino a las propiedades de éste como la fiabilidad, tiempo de respuesta y capacidad de almacenamiento.
2. Análisis: Se lleva a cabo el análisis de la solución a desarrollar, se emplean diagramas de casos de uso y modelo de objetos del dominio, siguiendo la notación UML, para definir las funcionalidades que tendrá el producto a desarrollar. Los artefactos propuestos por la metodología para la etapa de análisis son: modelo de casos de uso, modelo de objetos del dominio, prototipos en papel, guías de estilos, objetos del dominio y patrones de interacción (Ver Figura 1.3), los cuales se describen a continuación:



Figura 1.3: Etapa de Análisis

- Modelo de Casos de uso: un caso de uso expresa todas las formas de usar un sistema para alcanzar una meta particular para un usuario (Jacobson,

2005). En conjunto, los casos de uso le proporcionan todos los caminos útiles de usar el sistema e ilustran el valor que éste provee. Los casos de uso dejan claro lo que hará un sistema y, por omisión intencional, lo que no hará. Estos posibilitan una visión efectiva, el manejo del alcance y el desarrollo incremental de sistemas de cualquier tipo y cualquier tamaño.

- **Modelo de Objetos del dominio:** un modelo de objetos del dominio permite a los desarrolladores describir formalmente las características de la solución a desarrollar a través de la elaboración de un diagrama que describe los objetos del dominio del problema y sus relaciones estáticas.
- **Prototipo en papel:** esta técnica se basa en una simulación de la interfaz de usuario hecha sobre papel con el objetivo de explorar los requerimientos de usuario y de la interfaz. Se puede realizar utilizando papel y lápiz o utilizando herramientas de dibujo. El resultado obtenido es un prototipo de baja fidelidad que será refinado luego de la evaluación con el usuario. Se requieren materiales y recursos mínimos, es sencillo de realizar y puede desarrollarse en cualquier momento.
- **Guía de estilo:** una guía de estilo es un documento que recopila directrices relacionadas con el aspecto de la interfaz de usuario que ofrece una aplicación, tales como: logotipo, la gama de colores a utilizar en la interfaz de usuario, fuentes, distribución de los elementos en los diálogos y aspectos de los estilos de interacción (menú, botones, enlaces,...). La adopción de guías de estilo específicas se puede establecer como parte de los requerimientos no funcionales o requerimientos de usabilidad. Las guías de estilo incorporan buenas prácticas en el diseño de interfaces de usuario y mejoran su calidad.
- **Patrones de interacción:** Los patrones de interacción describen aspectos concernientes a la interfaz de usuario; están orientados a presentar soluciones apropiadas a problemas recurrentes en un contexto dado que se les presentan a los usuarios cuando utilizan las aplicaciones interactivas (Acosta, 2013). Son conocidos como patrones de interfaces de usuario o patrones de Interacción Humano-Computador (IHC). Para construir un patrón de interacción se indica su nombre, el problema de interacción que resuelve, la solución, y el contexto en el cual esa solución tiene éxito, además los aspectos de usabilidad que se implementan en esa solución, las fuerzas de la solución que están constituidas por sus fortalezas y debilidades, las consecuencias de usar el patrón que consiste en la respuesta por parte del sistema, y por último se indica si el patrón que se construye está relacionado con algún otro patrón de interacción que describa alguna interfaz que surja a partir de la solución que se describe. Los patrones de interacción asociados a una aplicación deben describir todas las interfaces de usuario provistas por el sistema que se desarrolla, y conforman un catálogo de patrones para esa aplicación. Existen diferentes tipos de patrones que dependen de lo que se desea describir: usuario, dominio, sistema, tarea, elemento compuesto y

elemento simple.

3. Prototipaje: Se implementa un prototipo rápido de la interfaz de usuario a partir de los patrones de interacción, el cual va evolucionando hasta convertirse en el producto final, se determina la guía de estilo, y se realizan evaluaciones de usabilidad apropiadas a esta etapa: las evaluaciones heurísticas y la lista de comprobación. Los artefactos propuestos por la metodología para la etapa de prototipaje son: prototipo rápido, evaluación heurística, listas de comprobación, pensamiento en voz alta y prototipo ejecutable (Ver Figura1.4), los cuales son descritos a continuación:

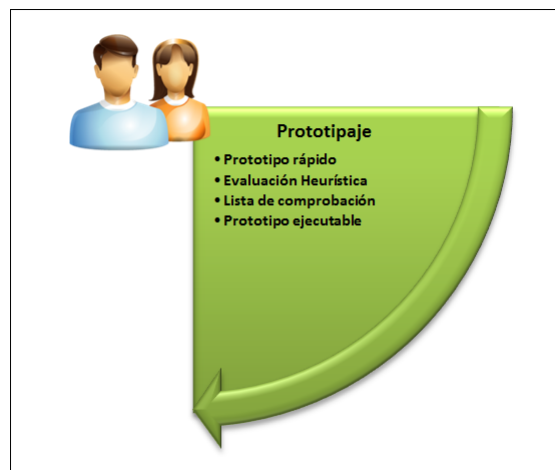


Figura 1.4: *Etapa de Prototipaje*

- Evaluación Heurística: es una técnica de inspección de usabilidad en la que los especialistas buscan problemas de usabilidad en la interfaz de usuario por medio del estudio de la interfaz de usuario y la evaluación de cada uno de sus elementos ante una lista de principios o heurísticas, comúnmente aceptadas. Nielsen propone una lista de diez (10) heurísticas, resultando suficiente y aceptable para cualquier evaluación de diseño. En la mayoría de las ocasiones se genera un informe con el resumen de los problemas de usabilidad encontrados, el cual indica las heurísticas no respetadas, proporcionando una orientación para su solución y una valoración del producto de software.
- Lista de comprobación: Son recomendaciones que el equipo de desarrollo ha acordado considerar en el diseño de la interfaz de usuario. Es posible utilizar las heurísticas de Nielsen o en algunos casos adaptar a los aspectos a los que se enfrenta el usuario del producto en desarrollo. Las listas de comprobación se pueden utilizar en cualquier etapa del ciclo de vida del software, siempre que se cuente con un prototipo del sistema en desarrollo.
- Prototipo ejecutable: un prototipo ejecutable consiste en un prototipo de alta fidelidad, que contiene la integración y la implementación de la navegación

entre las pantallas y el resto de los detalles necesarios para que este artefacto sea ejecutable.

4. Entrega: Se aplican las pruebas al sistema para certificar que la aplicación desarrollada sea un software usable y sin errores, finalmente se pone en producción la aplicación. Los artefactos propuestos por la metodología para la etapa de entrega son: protocolo de preguntas, pruebas de aceptación y aplicación a liberar (Ver Figura1.5), los cuales son descritos a continuación.



Figura 1.5: Etapa de Entrega

- Protocolo de preguntas: esta técnica permite conocer las manifestaciones del usuario con respecto al producto mediante la formulación de preguntas directas acerca del mismo. La capacidad del usuario para contestar estas preguntas sirven de ayuda para determinar las debilidades de la interfaz de usuario.
- Pruebas de aceptación: Las pruebas de aceptación conducidas por el cliente verifican que el sistema satisface los requerimientos funcionales y no funcionales. En general son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario; sin embargo, se pueden utilizar para verificar la satisfacción de los clientes en relación a los requerimientos no funcionales, aquellos que competen a la interfaz.
- Aplicación a liberar: esta aplicación consiste en el producto de software que será puesto en producción y será utilizado por el usuario final.

La metodología AgilUs descrita en esta sección será utilizada para llevar a cabo el proceso de desarrollo del sistema CONEST Desarrollo, durante el cual se realizan

varias iteraciones y se utilizan algunos artefactos propuestos por ésta metodología. Este proceso se describe en el capítulo 3 de este trabajo.

Una vez descritas las fases y artefactos que propone la metodología AgilUs, a continuación se describen las tecnologías que se relacionan con la plataforma del sistema CONEST y que pueden ser utilizadas como parte de la plataforma del sistema CONEST Desarrollo.

1.2. Tecnologías Web

Debido a las múltiples ventajas que ofrecen las Tecnologías de Información y Comunicación, el sistema de gestión académica CONEST esta basado en una aplicación Web. Se denomina *aplicación Web* a aquella que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una Intranet mediante un navegador Web (Shklar y Rosen, 2009). Son populares debido a lo práctico del navegador Web como cliente ligero, así como por la facilidad de actualización y mantenimiento de las mismas.

Las aplicaciones Web funcionan bajo la arquitectura *cliente/servidor*. Esta arquitectura representa un modelo de aplicación distribuida, donde las tareas se dividen entre los proveedores de recursos (servidores) y los entes que realizan peticiones (clientes), los cuales se guían por un conjunto de reglas en su comunicación (protocolo HTTP). Es decir, un cliente realiza peticiones a un servidor, el cual procesa la petición y envía una respuesta, a través de la utilización del protocolo HTTP. Esta separación entre el cliente y el servidor es de tipo lógico, ya que no necesariamente se ejecutan en diferentes computadoras. En la Figura 1.6 puede observarse gráficamente el funcionamiento de una arquitectura *cliente/servidor*.

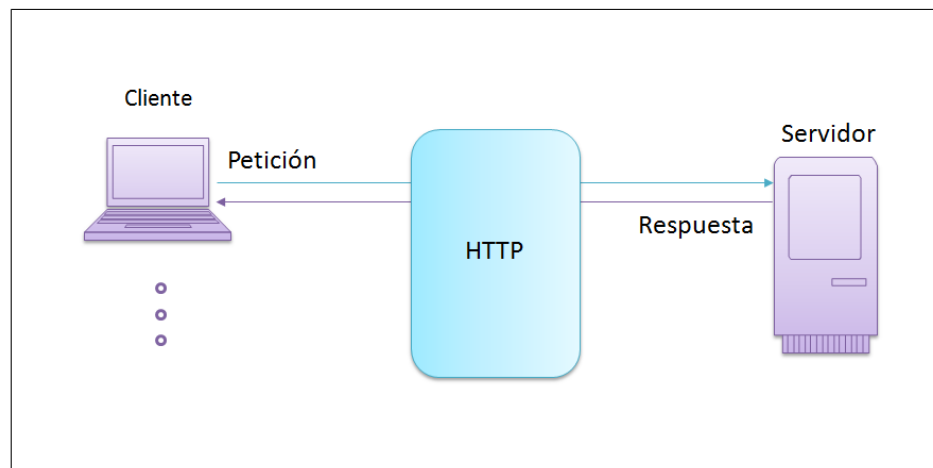


Figura 1.6: *Arquitectura cliente/servidor*

Algunas ventajas de las aplicaciones Web son:

- El navegador del cliente, permite que una misma versión de la aplicación pueda ejecutarse en múltiples plataformas como Windows, Linux, entre otros.
- Las aplicaciones basadas en tecnologías Web no necesitan ser descargadas, instaladas y configuradas. Además pueden ser accedidas desde cualquier dispositivo conectado a Internet a través de un navegador.

La principal desventaja de las aplicaciones Web es que la disponibilidad de una aplicación Web depende del proveedor de servicios del usuario, ya que requieren de una conexión a Internet.

En el contexto del desarrollo de una aplicación Web existen tecnologías que se ejecutan del lado del cliente y otras que se ejecutan del lado del servidor. 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. Las tecnologías del lado del cliente interesantes en el contexto de esta investigación son: HTML, CSS y *JavaScript*. Estas tecnologías se encuentran descritas a continuación.

1.2.1. HTML

HTML es el lenguaje utilizado para describir la estructura de los documentos Web, según (W3C-b, 2013), y es el código interpretado por el cliente de las aplicaciones Web. HTML utiliza una sintaxis especial que contiene etiquetas asociadas a segmentos de código, las cuales indican al navegador cómo debe interpretarse el documento. Las aplicaciones Web son ampliamente utilizadas debido a lo práctico del navegador Web como cliente, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software. HTML es utilizado en las aplicaciones Web para designar la función de cada elemento dentro de la aplicación, por ejemplo: una tabla, un párrafo, etc. Una vez creados los contenidos, se utilizan diferentes tecnologías para mejorar la apariencia final de la aplicación, como por ejemplo CSS.

1.2.2. CSS

CSS (*Cascade Style Sheet*) u *Hojas de estilo en cascada* es un mecanismo utilizado para definir el aspecto y la presentación de documentos escritos en HTML y XML (*Extended Markup Language*), separando el contenido de la presentación. Determina cómo deben

desplegarse los elementos estructurados en una pantalla, en un documento, entre otros, según (W3C-a, 2011). La apariencia final de un elemento puede ser especificado en diferentes lugares, lo que hace a CSS una herramienta muy poderosa pero puede agregar complejidad a la hora de depurar el código. Los estilos relacionados con el documento pueden estar especificados en tres posibles lugares: en un archivo externo, en una definición al comienzo del documento o en un elemento específico dentro de una etiqueta del documento HTML.

Algunas ventajas de la utilización de CSS son:

- Separación del contenido de la presentación, lo que facilita al creador, diseñador, usuario o dispositivo electrónico que muestre la página, la modificación de la visualización del documento sin alterar el contenido del mismo, sólo modificando algunos parámetros del CSS.
- Permite que el contenido pueda ser visualizado en diferentes formatos y dispositivos seleccionando el CSS apropiado.

Cuando el navegador despliega un documento, lo realiza en dos fases, primero el navegador convierte el código del lenguaje de marcado en el DOM (*Document Object Model*), el cual representa el documento en memoria y lo combina con el estilo del mismo para posteriormente mostrar por pantalla su contenido.

El *DOM* o *Modelo de Objetos del Documentos*, es un *API* (*Application Programming Interface*) o *Interfaz de Programación de Aplicaciones* utilizada para manipular documentos HTML, que proporciona una representación estructural de los documentos y permite modificar su contenido y presentación visual. El DOM es una representación completamente orientada a objetos de un documento HTML, que puede ser modificado con un lenguaje de *scripting* como *JavaScript*.

1.2.3. JavaScript

Un *script* es un código de programa que no necesita procesamiento antes de su ejecución (W3C-c, 2013). En general, en el contexto Web, *scripting* se refiere a código de programa escrito en *JavaScript* ejecutado por el navegador en respuesta a un evento activado por el usuario, estos *scripts* permiten que las aplicaciones Web sean más dinámicas, incorporando una interactividad significativa para el usuario.

JavaScript es un lenguaje de programación interpretado, conocido como el lenguaje de programación para páginas Web (MozillaDeveloperNetwork-d, 2014). Está basado en prototipos, es dinámico y compatible con los paradigmas de programación orientado a

objetos, imperativo y funcional. Actualmente, la mayoría de los navegadores interpretan el código *JavaScript* integrado en las páginas Web. Para interactuar con una página Web se provee al lenguaje *JavaScript* de una implementación del DOM. Dado que el código *JavaScript* puede ejecutarse localmente en el navegador del usuario, este puede responder a las acciones del usuario con rapidez, haciendo una aplicación más eficiente, además permite detectar acciones de los usuarios que HTML por sí solo no puede.

Las características más importantes de este lenguaje son:

- Los tipos de datos están relacionados con valores y no con variables. Soporta varios métodos para comprobar el tipo de dato de un objeto.
- Es posible ejecutar sentencia de código leídas en tiempo real.
- Las funciones son de primera clase, por lo que son objetos en sí mismos y como tal tienen propiedades y métodos y pueden ser pasadas como parámetro.
- Soporta expresiones regulares.
- Al igual que el código de CSS el código de JavaScript pueden estar embebidos en una sección del código HTML del documento o en un archivo externo.

Las tecnologías estudiadas en esta sección se encuentran relacionadas a la implementación de la aplicación CONEST Desarrollo, ya que a través de la utilización de tecnologías como HTML, CSS, *Javascript* y AJAX, es posible la creación de lineamientos para la creación de componentes de interfaz de manera estandarizada. Aunque son numerosos los beneficios ofrecidos por estas tecnologías estudiadas, es necesario estudiar las tecnologías del lado del servidor, las cuales permiten incorporar funcionalidades adicionales necesarias para la implementación de la aplicación CONEST Desarrollo.

Las tecnologías del lado del servidor permiten procesar las peticiones del usuario mediante la interpretación de *scripts* en el servidor Web, para generar páginas HTML dinámicamente como respuesta al cliente. A continuación se describe las tecnologías del lado del servidor *Ruby on Rails* y *MySQL*.

1.2.4. Ruby on Rails

*Ruby on Rails*¹ (RoR) es un *framework* de desarrollo de aplicaciones Web de código abierto escrito en el lenguaje de programación *Ruby*, el cual es un lenguaje de programación interpretado, dinámico, reflexivo y orientado a objetos, creado por el programador japonés Yukihiro "Matz" Matsumoto en 1995. Su implementación oficial es distribuida bajo una licencia de software libre. Combina una sintaxis influenciada por *Perl*, *Eiffel*, *Python* y *Smalltalk*.

¹Página oficial de Ruby on Rails: <http://rubyonrails.org/>

Las características generales de *Ruby* son:

- Soporta múltiples paradigmas de programación, incluyendo el paradigma funcional, orientado a objetos, e imperativo.
- Soporta alteración de objetos en tiempo de ejecución.
- Permite el manejo de excepciones.
- Soporta el manejo de hilos independientemente del sistema operativo.
- Permite la carga dinámica de DLL/bibliotecas compartidas en la mayoría de las plataformas.

El *framework Ruby on Rails* combina la simplicidad con la posibilidad de desarrollar aplicaciones escribiendo menos código que con otros *frameworks* y con un mínimo de configuración, promoviendo el desarrollo ágil. Este *framework* se enfoca en la utilización de los principios y buenas prácticas conocidas como *Convention over configuration* y *Don't repeat yourself*, así como el patrón arquitectónico *Modelo-Vista-Controlador* (MVC).

Convention over Configuration, también conocido como CoC, es un paradigma de programación que busca disminuir el número de decisiones que un desarrollador necesita hacer. La fase principal implica que un programador solo necesita especificar los aspectos no convencionales de la aplicación y cuando la convención definida no es suficiente para lograr el comportamiento deseado, el desarrollador puede alterar el comportamiento por defecto y adaptarlo a sus necesidades. Mediante este paradigma se hace innecesario realizar aquellas tareas para las que la convención ya ha definido un comportamiento.

Don't repeat yourself, es un principio de definición de procesos, en el cual toda porción de información (desde datos en la base de datos de la aplicación hasta una fracción de código) nunca debe ser duplicada, ya que la duplicación incrementa la dificultad en los cambios y puede provocar inconsistencias. Cuando este principio se aplica de forma eficiente, los cambios en cualquier parte de la aplicación requiere cambios en un único lugar.

Ruby on Rails utiliza el patrón arquitectónico MVC con el fin de mejorar la capacidad de mantenimiento de la aplicación. El *modelo* centraliza la lógica de negocio, la *vista* gestiona la lógica de visualización, mientras que el *controlador* gestiona el flujo de la aplicación. MVC permite una separación limpia de la lógica de negocio y las vistas HTML. En la Figura 1.7 se observa el funcionamiento de la arquitectura MVC.

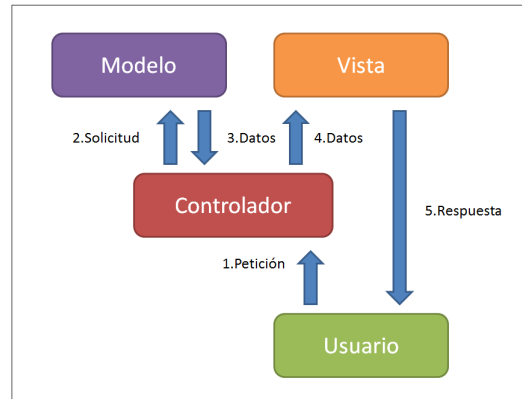


Figura 1.7: *Arquitectura MVC*

Los componentes de la arquitectura MVC desde la perspectiva de *Ruby on Rails* son las siguientes:

- **Modelo:** el modelo lleva la lógica de negocio de la aplicación y las reglas para manipular los datos. En *Ruby on Rails*, se utilizan los modelos para gestionar la interacción con los elementos correspondientes de la base de datos. Los modelos representan la información en la base de datos y hacen las validaciones apropiadas.
- **Vista:** es el *front-end* de la aplicación, representando la interfaz de usuario. En *Ruby on Rails*, las vistas son archivos HTML con código *Ruby* embebido, el cual es bastante simple (ciclos y condicionales). Sólo se utiliza para mostrar los datos al usuario en forma de vistas. Las vistas se utilizan para proporcionar los datos para los navegadores que solicitaron las páginas Web. Las vistas proveen el contenido del servidor en varios formatos, como HTML, PDF, XML, entre otros.
- **Controlador:** el controlador interactúa con los modelos y las vistas. Las solicitudes entrantes de los navegadores son procesados por los controladores, que procesan los datos de los modelos y la transmiten a las vistas para la presentación.

Muchas de las tareas comunes del desarrollo Web son incorporadas al *framework* para trabajar de manera modular, por ejemplo: la gestión del correo electrónico, la correspondencia de los objetos de base de datos, estructuras de archivos, generación de código, entre otras. *Ruby on Rails* cuenta con las siguientes características:

- Implementa la técnica REST (*Representational State Transfer*) para servicios Web.
- Soporta las principales manejadores de bases de datos: *MySQL*, *Oracle*, *MS SQL Server*, *PostgreSQL*, entre otros.

- Posee generadores de scripts para automatizar tareas.

En la Figura 1.8 se observan los módulos que componen *Ruby on Rails* y se describen a continuación.

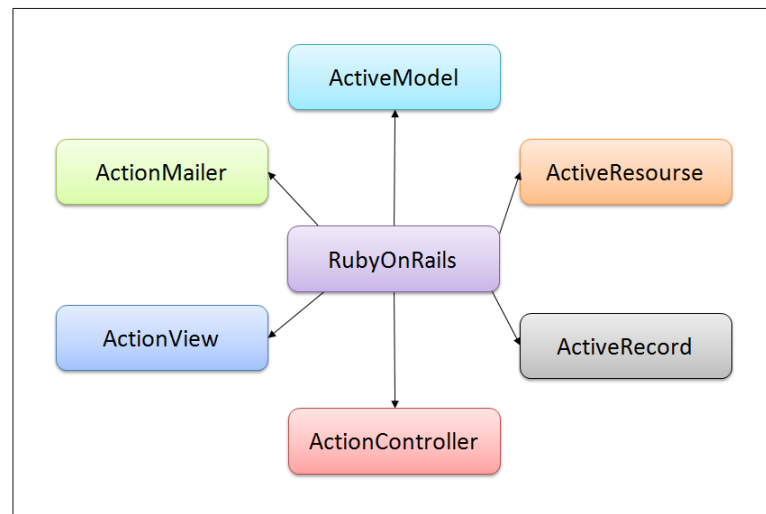


Figura 1.8: *Componentes de Ruby on Rails*

- *ActionMailer*: es responsable de proveer servicios de correo electrónico. Es utilizado para procesar correos electrónicos: bandeja de entrada, envío de texto simple o correos complejos basados en plantillas.
- *ActionController*: es el componente que gestiona los controladores en una aplicación Rails, proporcionando un controlador base del cual pueden heredar otros controladores. Procesa las solicitudes entrantes, extrae los parámetros y despacha a la acción correspondiente. Los servicios provistos por ActionController incluyen la gestión de sesiones, módulos auxiliares (*helpers*), despliegue de plantillas y gestión de redirecciones.
- *ActionView*: proporciona los diseños generales de la aplicación, plantillas de búsqueda y los ayudantes de la vista (*helpers*) que facilitan la creación de código HTML y otros formatos de presentación, es decir, gestiona todas las vistas de la aplicación Rails. Implementa las funcionalidades de AJAX.
- *ActiveModel*: proporciona una interfaz entre los módulos de ActiveSupport y ActiveRecord. Estas interfaces también pueden ser utilizadas fuera del framework, para proveer funcionalidades de ORM (*Object-Relational Mapping*).

- *ActiveRecord*: es un patrón arquitectónico que provee la funcionalidad de manejar datos en una base de datos relacional como objetos. Este módulo provee la capa modelo que conecta las tablas de la base de datos con su representación en las clases de Ruby. Provee independencia de la base de datos, funcionalidades básicas CRUD (*Create, Read, Update and Delete*), capacidades avanzadas de búsqueda y la habilidad de relacionar modelos entre sí. Este módulo contiene la lógica de negocio, maneja las validaciones y relaciones, proporciona los métodos para consultar y modificar registros en la base de datos y soporta diferentes tipos de bases de datos.
- *ActiveResource*: provee un marco de desarrollo para gestionar la conexión entre los objetos de negocio y los servicios Web. Implementa una forma de realizar la correspondencia de los recursos Web a objetos locales con semántica CRUD.

1.2.5. MySQL

En la actualidad los datos son uno de los activos más importantes de cualquier organización. Por esta razón es necesario contar con un software robusto, seguro y de alta disponibilidad, que pueda almacenar información y procesarla rápidamente. Una *Base de Datos* (BD) es un repositorio de datos estructurado, diseñado para apoyar el almacenamiento de datos eficiente (Chong, 2010). La principal ventaja de utilizar BD es que múltiples usuarios pueden acceder a ellas al mismo tiempo. La administración de las Bases de Datos se realiza a través de un *Sistema Manejador de Base de Datos*.

Se define un *Sistema Manejador de Base de Datos* (SMBD) como una colección de datos relacionados entre sí, estructurados y organizados que posee un conjunto de herramientas de software que controlan el acceso, organización, almacenamiento, gestión, recuperación y mantenimiento de los datos (Chong, 2010).

MySQL es un sistema manejador de bases de datos relacional, multihilo y multiusuario. Una base de datos relacional archiva datos en tablas separadas, esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas en una consulta. Provee un esquema de licenciamiento dual, que permite descargar, modificar y usar gratuitamente *MySQL* bajo una licencia GNU (*General Public License*). *MySQL* es utilizado por sitios Web populares en la actualidad, tales como *Youtube, Facebook, y Twitter*, entre otros.

Para el desarrollo de una aplicación Web que satisfaga los requerimientos del usuario es necesario seleccionar un conjunto de tecnologías apropiadas al contexto de la aplicación. Es importante mencionar que a lo largo del presente capítulo se describieron las tecnologías más relevantes que forman parte de las plataformas de los sistemas CONEST y CONEST Desarrollo, tales como HTML, CSS, *Ruby on Rails y MySQL*.

Capítulo 2 Análisis de las interfaces de usuario desplegadas por el sistema CONEST

En este capítulo se presenta el estudio de las interfaces de usuario (IU) ofrecidas por el sistema CONEST, el cual se llevó a cabo a través de la observación y exploración de las IU de los diferentes módulos del sistema y consistió en dos fases principales: clasificación de IU y creación de patrones de interacción. Durante la fase de clasificación se determinaron las IU del sistema que se proponen para el catálogo de patrones de interacción del sistema CONEST Desarrollo, y en la fase de creación de patrones de interacción se presenta un conjunto de patrones de interacción relacionados con los componentes de IU del sistema CONEST determinados en la fase anterior.

El sistema CONEST fue creado en el año 2007 por la División de Control de Estudios con el fin de automatizar la mayoría de los procesos llevados a cabo por esta división. En la actualidad, este sistema se ha extendido de manera desproporcionada, por lo que el grupo de desarrollo decidió crear una nueva versión de este sistema, incorporando nuevas tecnologías e integrando los conocimientos, la madurez y la experiencia de los líderes de este proyecto adquirida durante la implementación de las versiones previas. De esta manera se tiene como objetivo mejorar la apariencia de las IU, haciendo énfasis en el atributo de la usabilidad como estándar de calidad del software.

Actualmente el sistema CONEST posee una gran cantidad de IU que se distribuyen entre sus diferentes módulos, incrementando la complejidad de llevar a cabo el análisis de sus IU. Por esta razón este análisis se llevó a cabo en 3 diferentes iteraciones, tal como puede observarse en la Figura 2.1. Durante la primera iteración, los componentes de interfaz de usuario se organizaron por cada uno de los procesos gestionados a través del sistema, lo cual dejó como resultado diferentes componentes aislados que no se relacionaban directamente con ninguno de éstos procesos. En la siguiente iteración se organizaron los componentes por cada uno de los módulos del sistema, teniendo como resultado una organización apropiada y general de los componentes. En la última iteración de este proceso se decidió generalizar los componentes que se encontraban

repetidos en los diferentes módulos del sistema con el fin de evitar la repetición de componentes y a los cuales se denominó Componentes Generales.



Figura 2.1: *Iteraciones del proceso de organización de IU*

Con el fin de organizar los componentes de IU del sistema de manera eficiente y sencilla en este trabajo, se propone un esquema que clasifica los componentes de IU y está basado en la taxonomía de los patrones de interacción, los cuales son una solución exitosa a un problema recurrente en un contexto dado. De esta manera es posible establecer una analogía entre los componentes de interfaz y los patrones de interacción asociados a través de este esquema de clasificación. Según el grado de complejidad de la interfaz de usuario la clasificación se divide en componentes simples, compuestos y tarea. A continuación se presenta la descripción de cada uno de ellos.

- **Componente simple:** esta categoría incluye todos aquellos componentes de interfaz que permiten la implementación de funcionalidades básicas, las cuales llevan a cabo acciones tales como descarga de archivos y redireccionar a una página. Algunos ejemplos de esta clasificación son botones, formularios, pestañas y enlaces.
- **Componente compuesto:** esta categoría comprende todos aquellos grupos de componentes simples relacionados, que permiten la construcción de funcionalidades

que requieren de una lógica compleja dentro de una IU, las cuales comprenden aquellas funcionalidades que requieren la interacción de múltiples componentes simples. Algunos ejemplos de este tipo de componentes son: formularios de inicio de sesión, buscadores y menús.

- **Componente tarea:** dentro de esta categoría se encuentran los componentes que combinan múltiples componentes compuestos y que permiten la construcción de las IU de un sistema. Adicionalmente estos componentes pueden generalizarse de tal manera que pueden reutilizarse en diferentes IU, promoviendo la reutilización de código y disminuyendo los tiempos de implementación de las IU. Algunos ejemplos de estos componentes son páginas principales, perfiles de usuario, calendarios, carritos de compras, entre otros.

A través del estudio de los módulos de este sistema, se determina el conjunto de IU que se incorporarán al catálogo de patrones de interacción del sistema CONEST Desarrollo, por medio de la notación propuesta en esta sección, creando una analogía con los patrones de interacción y facilitando su desarrollo.

2.1. Organización de las IU del sistema CONEST

El sistema CONEST en su versión actual provee alrededor de 50 funcionalidades en total, con un número aproximado de 100 componentes de IU asociados. A lo largo de esta investigación se estudiaron las funcionalidades más importantes de cada uno de los módulos existentes en este sistema, con el fin de explorar de manera eficiente las IU del sistema CONEST. Esta organización incorpora un conjunto de componentes generales que se encuentran presentes en todos los módulos y se realizó en función de sus 3 módulos principales: Estudiante, Docente y Administrativo. A continuación se describen los componentes resultantes de esta clasificación.

2.1.1. Componentes generales

Durante la elaboración de esta clasificación se determinó un total de 15 componentes de IU comunes a todos los módulos y que se denominaron Componentes Generales, de los cuales se describen a continuación los siguientes componentes: menú principal, directorio actual, perfil de usuario, *widget* y consultar período académico.

1. **Menú principal:** es un componente compuesto en forma de barra de navegación, la cual contiene las pestañas del menú principal que proveen las funcionalidades que

ofrece cada módulo del sistema, tales como inscripción, calificación, constancias, gestión de entidades, entre otros. (Ver Figura 2.2).

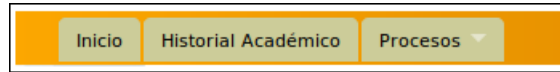


Figura 2.2: *Componente general: Menú Principal*

2. Directorio actual: este componente simple consiste en una pestaña ubicada en la parte superior de la interfaz de usuario, la cual permite conocer el directorio actual en el que se encuentra el usuario. (Ver Figura 2.3).

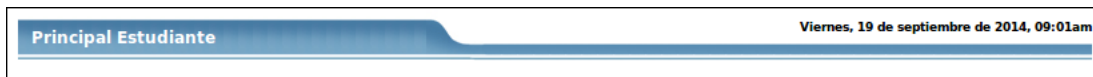


Figura 2.3: *Componente general: Directorio actual*

3. Perfil de usuario: este componente de tipo tarea permite gestionar el perfil del usuario. Consiste en una barra lateral, la cual contiene la información básica del usuario y un enlace que permite modificar datos personales, fotografía del perfil, entre otros. (Ver Figura 2.4).

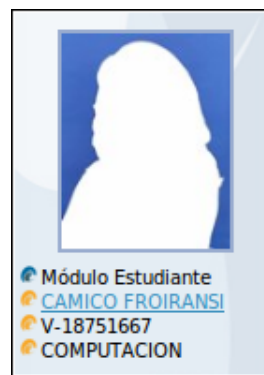


Figura 2.4: *Componente general: Perfil de usuario*

4. Widget: este componente de tipo tarea permite cargar un conjunto de *Widgets* dentro de la interfaz de usuario principal de cada módulo. Consiste en una tabla, la cual contiene en el encabezado el nombre de la funcionalidad y facilita el acceso directo a funcionalidades frecuentemente utilizadas. (Ver Figura 2.5).



Figura 2.5: Componente general: Widget

5. Consultar período académico: este componente de tipo tarea permite a los usuarios consultar la planificación del período académico actual. Consiste en un conjunto de tablas, las cuales a través de diferentes enlaces, permiten la consulta de los diferentes campos relacionados con la planificación del período académico, tales como asignaturas, requisitos, grupo docente, horarios, entre otros. (Ver Figura 2.6).

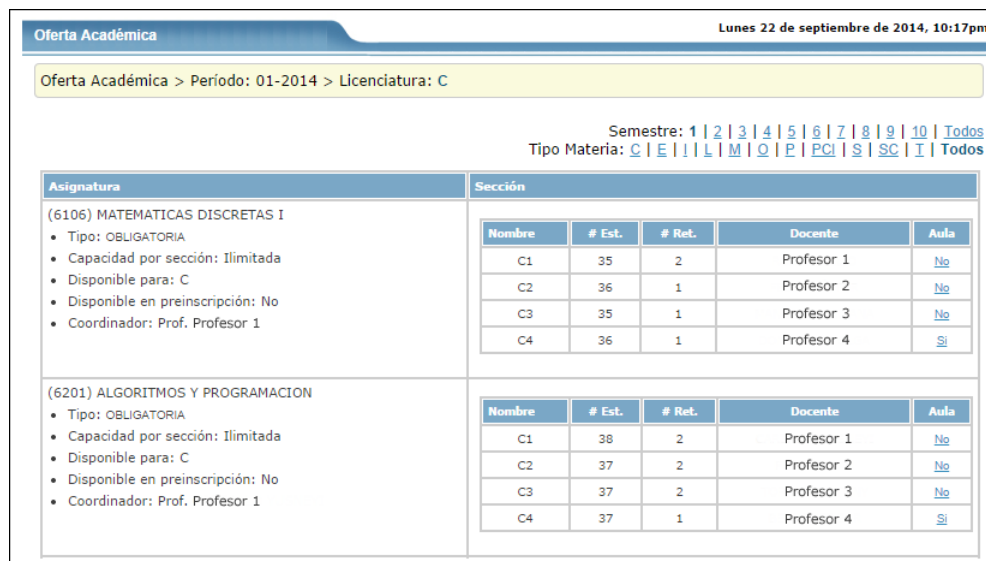


Figura 2.6: Componente general: Consultar período académico

2.1.2. Módulo estudiante

Este módulo incorpora todas las funcionalidades que ofrece el sistema CONEST para la comunidad de los 3500 estudiantes aproximadamente que pertenecen a la Facultad de Ciencias. Incorpora un conjunto de funcionalidades que representan un reto para el grupo de desarrollo, ya que existen procesos en los que participan la mayoría de estos estudiantes, tal como el proceso de inscripción, durante el cual todos utilizan el sistema en un período corto de tiempo, por lo que el sistema debe responder de forma eficaz y eficiente a las exigencias de este proceso, el cual se repite dos veces al año.

A continuación se describe brevemente algunos de los componentes de IU que se encuentran en este módulo.

1. Asignaturas inscritas en la carga académica actual: es un componente simple que contiene la lista de la carga académica en curso del estudiante. Consiste en una tabla que contiene la información relacionada con las asignaturas (código de la asignatura, nombre, sección, estado y acciones) e incorpora el botón de retiro de asignaturas (Ver Figura 2.7).

Código	Nombre	Sección	Estatus	Acciones
6109	CALCULO CIENTIFICO	C2	SIN CALIFICAR	Retirar
6204	LENGUAJES DE PROGRAMACION	C1	SIN CALIFICAR	Retirar
6344	PLANIFICACION ESTRATEGICA DE SISTEMAS	C1	SIN CALIFICAR	Retirar
8203	TECNICA Y RECURSOS DE APRENDIZAJE	M1	RETIRADA	-

Figura 2.7: Módulo estudiante: Carga académica actual

2. Preinscribir: es un componente simple que inicia el proceso de preinscripción de un estudiante, el cual consiste en un botón que permite realizar esta acción. Adicionalmente este componente puede generalizarse y ser utilizado para iniciar los procesos de inscripción y modificación de inscripción (Ver Figura 2.8).



Figura 2.8: Módulo estudiante: Realizar preinscripción

3. Agregar asignatura a preinscripción: es un componente tarea que está formado por una lista con las asignaturas disponibles para la preinscripción y permite agregar asignaturas a la lista de asignaturas a preinscribir. Consiste en una tabla que contiene botones que permiten agregar una asignatura al proceso de preinscripción. Este componente puede generalizarse y utilizarse durante para agregar asignaturas a los procesos de inscripción y modificación de inscripción (Ver figura 2.9).

Tipo	Asignatura	UC	Agregar
E	(6047) ANAL. DEL DESEMP. COMPUT.	5	
E	(6545) INTELIGENCIA DE NEGOCIO	5	
E	(6027) SIST. OPERATIVOS AVANZAD.	5	

Figura 2.9: Módulo estudiante: Agregar asignatura a preinscripción

4. Eliminar asignatura de preinscripción: es un componente tarea que permite eliminar asignaturas de la lista de asignaturas a preinscribir. Consiste en una tabla que contiene botones asociados a una asignatura y que permite eliminarlas del proceso de preinscripción. Adicionalmente este componente puede generalizarse para ser utilizado para eliminar asignaturas de los procesos de inscripción y modificación de inscripción (Ver Figura 2.10).

Tipo	Asignatura	UC	Eliminar
E	(6027) SIST. OPERATIVOS AVANZAD.	5	

Total de Creditos: 5

Figura 2.10: Módulo estudiante: Eliminar asignatura de preinscripción

5. Consultar requisitos de asignaturas: es un componente tarea que permite consultar los requisitos de una asignatura que no se encuentra en el componente agregar asignatura a preinscripción. Consiste en un buscador que proporciona la información relacionada a una asignatura a través de una ventana modal.

Adicionalmente puede generalizarse para ser utilizado durante los procesos de inscripción y modificación de inscripción (Ver Figura 2.11).

Consultar asignaturas

Si no aparece su asignatura, aquí puede consultar los requisitos.

Asignatura

Código o nombre de la asignatura

Consultar

Figura 2.11: Módulo estudiante: Consultar requisitos de asignatura

- Horario parcial: componente simple que permite visualizar el horario parcial de las asignaturas inscritas a través de una tabla. Adicionalmente puede ser generalizado para ser utilizado durante los procesos de inscripción y modificación de inscripción (Ver Figura 2.12).

Horario

Hora	Lun	Mar	Mié	Jue	Vie
7 - 8					
8 - 9					
9 - 10		6047-C1		6047-C1	
10 - 11		6047-C1		6047-C1	
11 - 12	6109-C2		6109-C2		6109-C2
12 - 13	6109-C2		6109-C2		6109-C2
13 - 14					
14 - 15					
15 - 16					
16 - 17					
17 - 18					
18 - 19					

Figura 2.12: Módulo estudiante: Horario parcial

- Seleccionar sección en asignatura: componente complejo que contiene una lista de secciones con su respectivo horario y que permite seleccionar la sección a inscribir a través de una ventana modal que contiene las secciones existentes y su horario asociado. (Ver Figura 2.13).



Figura 2.13: Módulo estudiante: Seleccionar sección en asignatura

8. Formalizar preinscripción: es un componente simple formaliza el proceso de preinscripción a través de un botón que lleva a cabo esta acción. Adicionalmente se puede generalizar este componente para ser utilizado durante los procesos de inscripción y modificación de inscripción (Ver Figura 2.14).

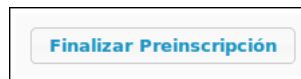


Figura 2.14: Módulo estudiante: Formalizar preinscripción

9. Retirar asignatura: es un componente complejo que consiste en una tabla con la lista de asignaturas inscritas en el semestre actual y permite hacer el retiro de estas asignaturas. (Ver Figura 2.15).



Figura 2.15: Módulo estudiante: Retirar asignatura

10. Horario del período académico actual: es un componente simple que organiza y permite visualizar el horario de la carga académica del estudiante a través de una tabla, incluyendo la información de la asignatura, tal como sección, profesor y aula. (Ver Figura 2.16).

Horario					
Hora	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
7:00 - 9:00	(6344) - PLAN. ESTRATEG. DE SISTEMAS Profesor 1	(6204) - LENGUAJES DE PROGRAMACION Profesor 2	(6344) - PLAN. ESTRATEG. DE SISTEMAS Profesor 1	(6204) - LENGUAJES DE PROGRAMACION Profesor 2	
9:00 - 11:00					(6204) - LENGUAJES DE PROGRAMACION Profesor 2
11:00 - 1:00	(6109) - CALCULO CIENTIFICO Profesor 3		(6109) - CALCULO CIENTIFICO Profesor 3		(6109) - CALCULO CIENTIFICO Profesor 3
1:00 - 3:00				(8203) - TEC. Y RECURSOS DE APREN Profesor 4	

Figura 2.16: Módulo estudiante: Horario del período académico actual

2.1.3. Módulo docente

Este módulo incluye todas las funcionalidades pertenecientes al sistema CONEST para la comunidad de los 400 docentes aproximadamente de la Facultad de Ciencias, los cuales poseen diferentes características y destrezas en la utilización de las aplicaciones Web, por lo que es necesario que este sistema ofrezca un conjunto de interfaces de usuario acorde con este perfil de usuario. Algunas de estas funcionalidades son: calificación, consulta del historial académico, consulta de un estudiante, gestión de datos personales, entre otros.

A continuación se describe brevemente algunos de los componentes de IU de este módulo.

1. Calificar: es un componente simple que permite iniciar el proceso de calificación de una asignatura, el cual consiste en un botón que permite dar inicio a este proceso. (Ver Figura 2.17).

Asignaturas por Calificar (01-2014)							
Asignatura	Sección	Rol	Est.	Ret.	Cal.	Pend.	Acciones
(6211) INTERACCION HUMANO-COMPUTADOR	C2	DOC.	27	0	0	27	Calificar

Figura 2.17: Módulo docente: Realizar calificación

- Consultar solicitudes de modificación de inscripción: es un componente tarea que permite consultar la lista de las solicitudes de modificación de inscripción, el cual consiste en una tabla organizada cronológicamente por la fecha de las solicitudes y contiene diferentes enlaces a las funcionalidades existentes relacionadas a este componente, tales como resumen y descarga de la solicitud. (Ver Figura 2.18).

Consulta de Solicitudes de Modificación de Inscripción						Viernes, 19 de septiembre de 2014, 11:53am
Solicitudes encontradas: 63						
Nro	Carrera	Estudiante	Descripción	Solicitud	Estado	
1	C	1234567 Estudiante 1	INSCRIPCIÓN EXTEMPORÁNEA - (0206) ACTIVIDAD CORAL	19/09/2014	NUEVA	
2	C	8901234 Estudiante 2	INSCRIPCIÓN EXTEMPORÁNEA - (6422) TRABAJO ESPECIAL GRADO (M)	01/07/2014	NUEVA	

Figura 2.18: Módulo docente: Consultar solicitudes de modificación de inscripción

- Solicitar jurado para Trabajo Especial de Grado: componente de tipo tarea que permite al docente realizar la solicitud de jurado para examinar un Trabajo Especial de Grado. Consiste en una barra de búsqueda que permite agregar a los estudiantes a la solicitud de jurado. (Ver Figura 2.19).

Solicitud de Jurado para TEG		Viernes, 19 de septiembre de 2014, 11:58am
Buscar		
Estudiante		
<input type="text" value="1234567 Estudiante"/>	<input type="button" value="+"/>	<input checked="" type="checkbox"/> Carrera: COMPUTACION <input checked="" type="checkbox"/> Asignatura: (6420) TRAB.ESP.GRADO(APLI.INTE)

Figura 2.19: Módulo docente: Solicitar jurado para Trabajo Especial de Grado

- Consultar solicitudes de jurado para Trabajo Especial de Grado: componente de tipo tarea que permite consultar las solicitudes de jurado del docente, el cual consta de una tabla organizada de manera cronológica y que incorpora diferentes

CAPÍTULO 2. ANÁLISIS DE LAS INTERFACES DE USUARIO DE CONEST

enlaces a las funcionalidades relacionadas a este componente, tales como resumen de solicitud, modificación de solicitud, entre otros. (Ver Figura 2.20).

Carrera	Asignatura	Estudiante	Solicitud	Aprobación C.E.	Planilla Notas
C	(6412) TRAB.ESP.GRADO(SIST.INF.)	1234567 Estudiante	15/07/2014	16/07/2014	Solicitar

Figura 2.20: *Módulo docente: Consultar solicitudes de jurado para Trabajo Especial de Grado*

5. Consultar planilla de notas de Trabajo Especial de Grado: este componente de tipo tarea permite al docente consultar la planilla de notas de un estudiante. Consiste en una tabla organizada cronológicamente que incorpora enlaces a las diferentes funcionalidades disponibles para este componente, tales como resumen de planilla, modificación, entre otros. (Ver Figura 2.21).

Carrera	Asignatura	Estudiante	Solicitud	Aprobación C.E.	Planilla Notas
C	(6420) TRAB.ESP.GRADO(APLI.INTE)	1234567 Estudiante	29/06/2014	02/07/2014	07/07/2014

Figura 2.21: *Módulo docente: Consultar planilla de notas de Trabajo Especial de Grado*

6. Formalizar calificación: componente simple que permite formalizar la calificación de una asignatura, el cual consiste en un botón para llevar a cabo esta acción. (Ver Figura 2.22).



Figura 2.22: *Módulo docente: Formalizar Calificación*

2.1.4. Módulo administrativo

El módulo administrativo incluye todas las funcionalidades que ofrece este sistema a la comunidad de los aproximadamente 100 empleados administrativos. Adicionalmente incorpora funcionalidades que permiten gestionar y controlar el estado de algunas funcionalidades disponibles para otros módulos, como por ejemplo las fechas disponibles para realizar los procesos de inscripción, calificación y fechas de retiro de asignaturas.

A continuación se describe brevemente algunos de los componentes de IU que se encuentran en este módulo.

1. Crear evento en el calendario: este componente de tipo tarea permite agregar un evento al calendario académico actual. Consiste en un formulario, el cual incluye la información del evento y un calendario que permite establecer la fecha del mismo. Este calendario se encuentra disponible en todos los módulos del sistema (Ver Figura 2.23).



Figura 2.23: *Módulo administrativo: Crear evento en el calendario*

2. Agregar período académico: este componente de tipo tarea permite agregar un nuevo período académico, indicando la fecha de inicio y la fecha de fin, tipo de semestre, entre otros. Consiste en un botón que despliega una ventana modal que contiene un formulario con la información necesaria para la creación de un nuevo período académico. (Ver Figura 2.24).



Figura 2.24: Módulo administrativo: Agregar período académico

3. Gestionar planificación del período académico: es un componente de tipo tarea que a través de diferentes componentes simples y compuestos permite modificar los parámetros de un período académico. Consiste en diferentes tablas que contienen los enlaces a las funcionalidades relacionadas con este componente, tales como agregar asignatura, modificar número de secciones, modificar datos de una asignatura, entre otros. (Ver Figura 2.25).

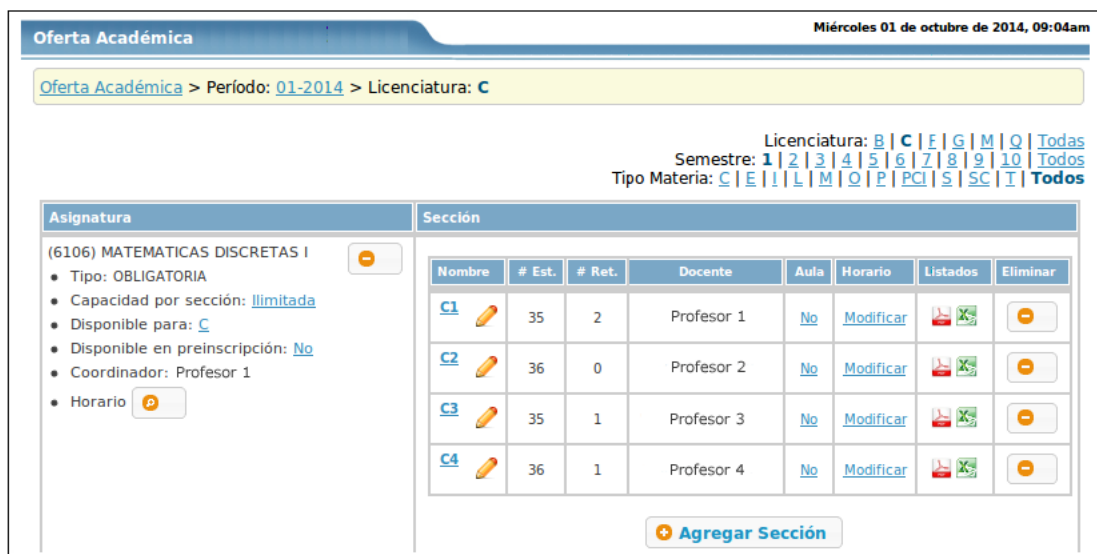


Figura 2.25: Módulo administrativo: Gestionar planificación del período académico

4. Búsqueda rápida: este componente compuesto forma parte del módulo de solicitudes estudiantiles y permite la búsqueda de estudiantes a través de un buscador. Asimismo proporciona enlaces directos para filtrar las solicitudes por escuela. (Ver Figura 2.26).



Figura 2.26: *Módulo administrativo: Búsqueda rápida*

5. Agregar asignatura a período académico: este componente tarea permite agregar una asignatura a un período académico a través de una ventana modal, la cual solicita información inherente a la acción, tales como nombre de la asignatura, capacidad de sección, entre otros. (Ver Figura 2.27).



Figura 2.27: *Módulo administrativo: Agregar asignatura a período académico*

6. Actualizar disponibilidad de asignatura: este componente tarea permite modificar la disponibilidad de la asignatura para las diferentes escuelas de la facultad a través de una ventana modal. (Ver Figura 2.28).

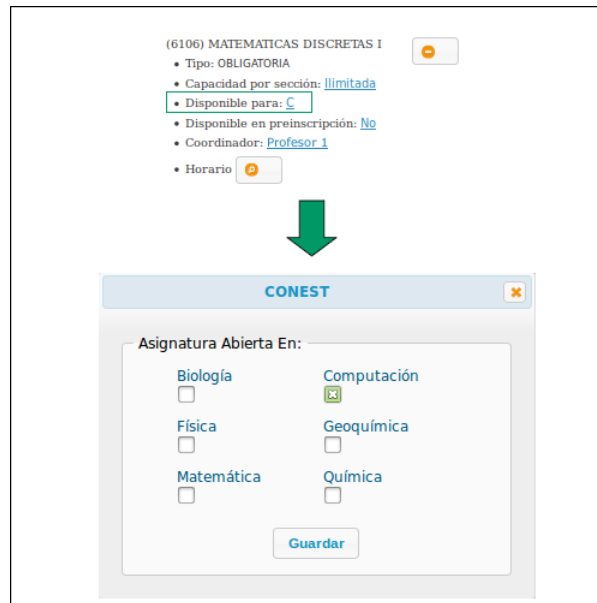


Figura 2.28: Módulo administrativo: Actualizar disponibilidad de asignatura

7. Actualizar requisito de preinscripción de asignatura: este componente tarea permite, a través de una ventana modal, cambiar el requisito de preinscripción de una asignatura. Adicionalmente permite elegir un algoritmo de selección de estudiantes diferente para cada asignatura. (Ver Figura 2.29).

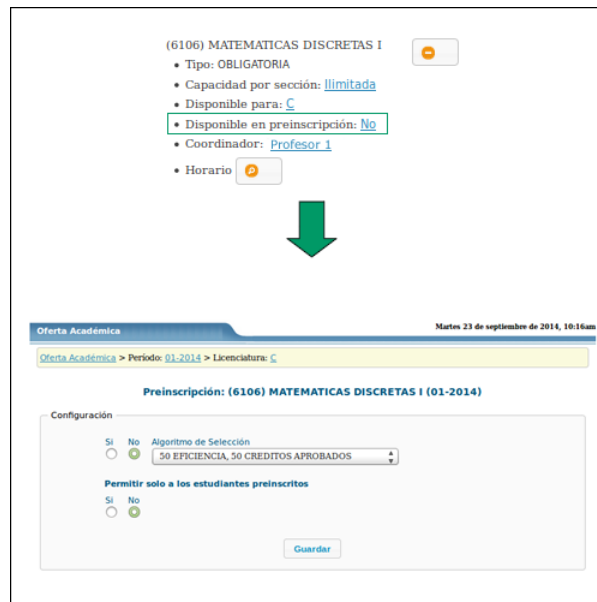


Figura 2.29: Módulo administrativo: Actualizar requisito de preinscripción de asignatura

8. Generar mapa de aulas: este componente de tipo tarea consiste en una ventana modal que permite llevar a cabo el proceso de asignación de aulas a las asignaturas pertenecientes al período académico, generando un archivo PDF con el mapa de aulas.

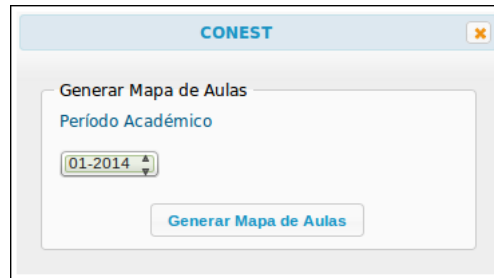


Figura 2.30: Módulo administrativo: Generar mapa de aulas

9. Modificar horario de asignatura: este componente tarea permite modificar el horario de una asignatura a través de una tabla que contiene los horarios disponibles. (Ver Figura 2.31).

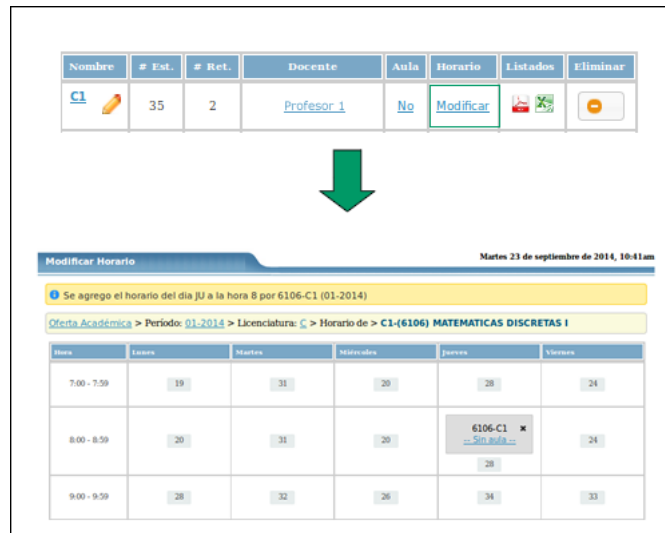


Figura 2.31: Módulo administrativo: Modificar horario de asignatura

10. Anular solicitud de jurado de Trabajo Especial de Grado: este componente de tipo tarea permite eliminar una solicitud de jurado de Trabajo Especial de Grado a través de una ventana modal que contiene los estudiantes relacionados a la solicitud que se desea eliminar. (Ver Figura 2.32).

CAPÍTULO 2. ANÁLISIS DE LAS INTERFACES DE USUARIO DE CONEST



Figura 2.32: Módulo administrativo: Anular solicitud de jurado de Trabajo Especial de Grado

11. Aprobar solicitud de jurado de Trabajo Especial de Grado: este componente de tipo tarea permite aprobar la solicitud de jurado. Consiste en una ventana modal que contiene un formulario con los campos relacionados a la aprobación de la solicitud. Para realizar esta actividad es necesario proveer información como fecha del Consejo de Escuela en el cual se realizó la aprobación y correlativo. (Ver Figura 2.33).

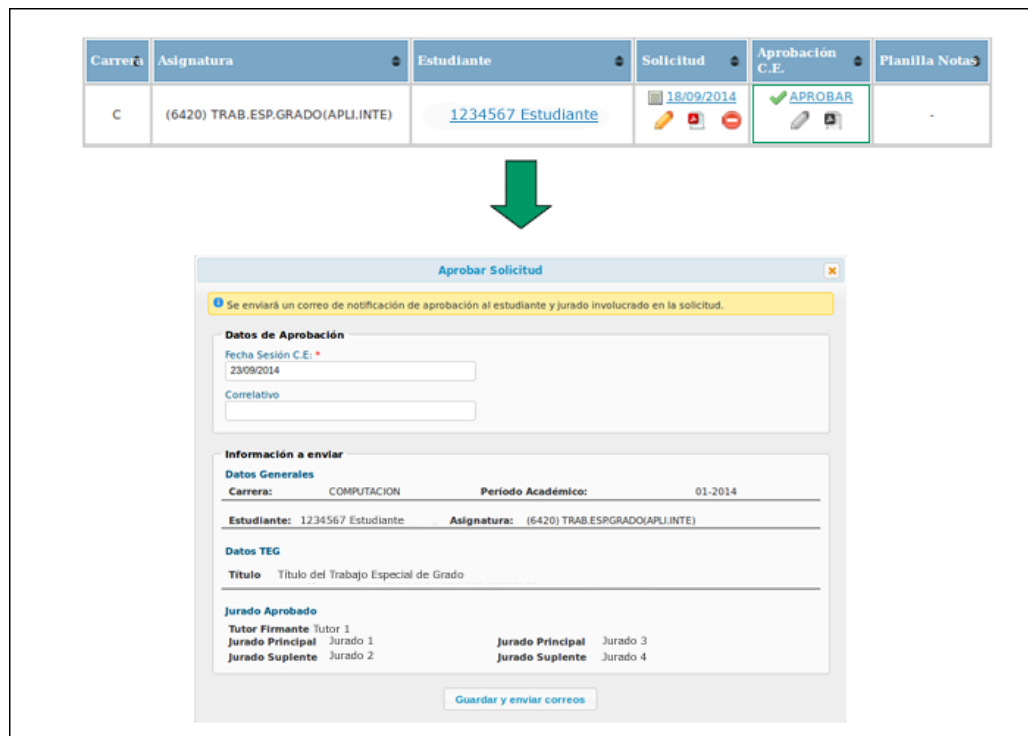


Figura 2.33: Módulo administrativo: Aprobar solicitud de jurado para Trabajo Especial de Grado

CAPÍTULO 2. ANÁLISIS DE LAS INTERFACES DE USUARIO DE CONEST

12. Resumen de solicitud: este componente tarea proporciona un resumen de las diferentes solicitudes existentes en el sistema a través de una ventana modal, entre las cuales se encuentran las solicitudes de jurado de Trabajo Especial de grado y las solicitudes de modificación de inscripción. (Ver Figura 2.34).

Carrera	Asignatura	Estudiante	Solicitud	Aprobación C.E.	Planilla Notas
C	(6420) TRAB.ESP.GRADO(APLI.INTE)	1234567 Estudiante	18/09/2014	✓ APROBAR	

Ver Resumen Solicitud

Datos Generales

Fecha de Solicitud: 18/09/2014 Sección: TE4362
 Carrera: COMPUTACION Período Académico: 01-2014

Estudiante: 1234567 Estudiante Asignatura: (6420) TRAB.ESP.GRADO(APLI.INTE)

Datos TEG

Título: Título del Trabajo Especial de Grado

Datos Tutores

Tutor Firmante: Tutor 1

Datos Jurados

Jurado Principal	Jurado 1	Jurado Principal	Jurado 3
Jurado Suplente	Jurado 2	Jurado Suplente	Jurado 4

Figura 2.34: Módulo administrativo: Resumen de solicitud

13. Gestionar solicitudes reincorporación: este componente de tipo tarea permite gestionar las solicitudes de reincorporación, lo cual incluye la aprobación o rechazo de las mismas. Consiste en una tabla organizada cronológicamente que contiene una lista de las solicitudes y enlaces que permiten consultar y anular las solicitudes. (Ver Figura 2.35).

Martes 23 de septiembre de 2014, 11:49am

Cédula o Nombre del Estudiante

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

Período Académico: [01-2014](#)

Solicitudes encontradas: 19

Nro	Carrera	Estudiante	Solicitud	Estado
1	C	1234567 Estudiante	20/07/2014	✓ PROCESAR
2	C	8901234 Estudiante 2	20/07/2014	✓ PROCESAR

Figura 2.35: Módulo administrativo: Gestionar solicitudes de modificación de inscripción

14. Gestionar estudiante: este componente de tipo tarea permite gestionar, a través de un formulario general, la información de un estudiante de la facultad. Esta información está clasificada como datos personales, historial académico, bitácora, correo electrónico y restablecimiento de contraseña. Adicionalmente este componente puede generalizarse para ser utilizado en la gestión de docentes y personal administrativo. (Ver Figura 2.36).

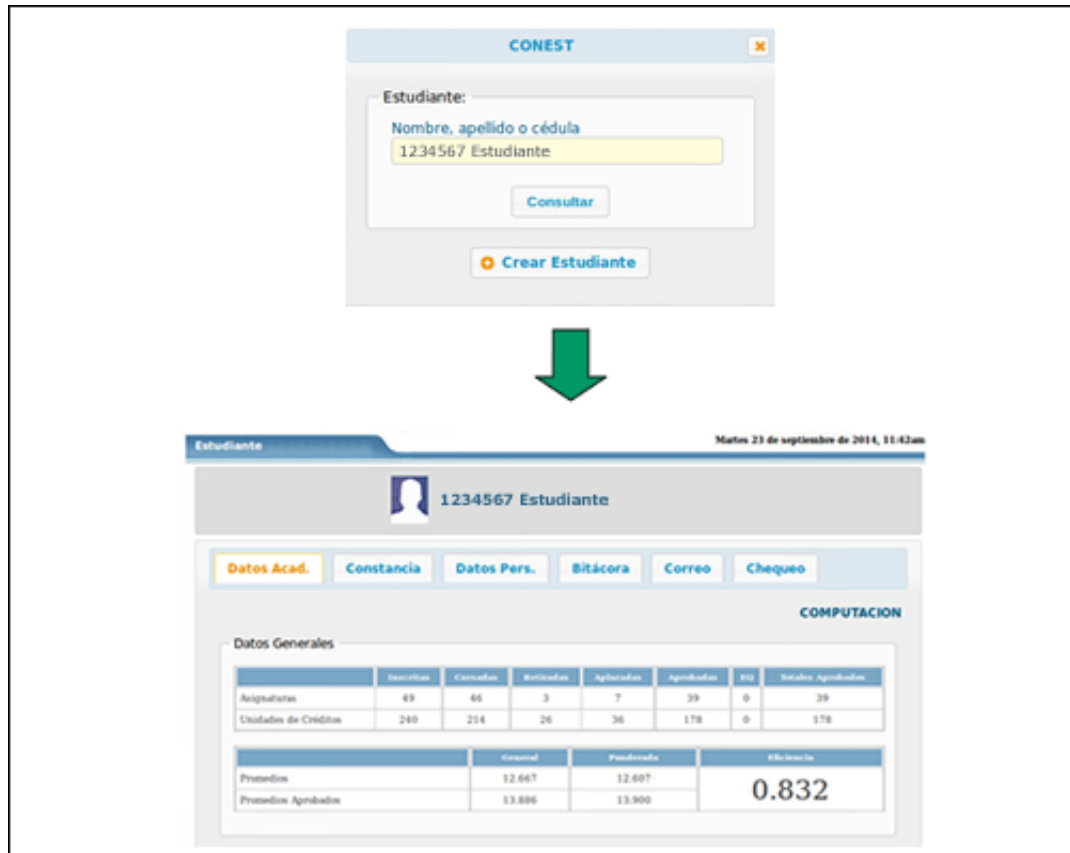


Figura 2.36: Módulo administrativo: Gestionar estudiante

15. Gestionar proceso de preinscripción: este componente de tipo tarea permite gestionar el proceso de preinscripción a través de una ventana modal. Permite cambiar el período académico y estado del proceso, además de mostrar el resultado del proceso a los estudiantes. (Ver Figura 2.37).



Figura 2.37: *Módulo administrativo: Gestionar proceso de preinscripción*

Es importante mencionar que esta clasificación de interfaces de usuario se realizó en distintos intervalos de tiempo, debido al proceso de desarrollo de funcionalidades nuevas y la modificación de las funcionalidades existentes dentro del sistema CONEST, el cual se llevó a cabo de manera paralela a esta investigación, por lo que en algunos casos la descripción de los componentes que se encuentra a lo largo de esta investigación puede variar con respecto al estado actual de los mismos. Adicionalmente es importante mencionar que este estudio contribuyó a la mejora de diferentes componentes de interfaz existentes y sirvió de base para la creación de componentes nuevos, como lo es el caso de los componentes de interfaz relacionados a la planificación de la oferta académica, en el cual se incorporaron algunas recomendaciones descritas en este trabajo.

Luego de realizar el estudio y clasificación de 60 componentes de interfaz de usuario del sistema CONEST (aproximadamente) a continuación se presentan los patrones de interacción asociados a los componentes estudiados en esta sección.

2.2. Creación de patrones de interacción

En general, los patrones de interacción pueden construirse a partir de los modelos de casos de uso y objetos del dominio, los cuales describen las funcionalidades y objetos de una aplicación, así como la consideración de los requerimientos no funcionales relacionados con la interfaz de usuario. En el caso del sistema CONEST, es un procedimiento complejo, debido al número de interfaces de usuario que posee y el gran número de casos de uso existentes. De igual forma el modelo de objetos del dominio resulta difícil de utilizar para este procedimiento, debido a la gran cantidad de objetos existentes en el sistema. Los patrones de interacción no se realizaron durante la fase de análisis del proceso de desarrollo del sistema CONEST, sino que se utilizaron para formalizar los componentes determinados en el análisis y exploración de las IU

CAPÍTULO 2. ANÁLISIS DE LAS INTERFACES DE USUARIO DE CONEST

existentes en la nueva versión de CONEST. Por esta razón los patrones de interacción del sistema CONEST se construyeron a partir de la clasificación de las interfaces de usuario realizada en este capítulo y se basan en la taxonomía de los patrones de interacción, la cual se describe en la Figura 2.38.







Categoría	Descripción	Notación
Patrones de Dominio	Captura aspectos de interfaz correspondientes a sistemas del mismo tipo.	
Patrones del Sistema	Están asociados al propósito del sistema y están relacionados con un dominio, heredando de éste los patrones allí definidos. En general, este tipo de patrón describe lo esencial y abre posibilidades a definir otros patrones.	
Patrones de Tareas	Asociados a la representación de las funcionalidades en la interfaz. Provee detalles de alto nivel acerca de la interfaz de las tareas que los usuarios ejecutan.	
Patrones de Elementos Compuestos	Pueden estar asociados al sistema o a las tareas, se conocen también como patrones de interacción, debido a que, en general, involucran un estilo de interacción tal como menús, formas, manipulación directa, etc.	
Patrones de Elementos Simples	Pueden ser referenciados por patrones de dominio o sistema, como ejemplo se tiene el patrón relativo al medio, donde se muestra la metáfora elegida o el patrón de barra de herramientas.	
Patrones de Usuarios	Describen los perfiles de los usuarios potenciales de la aplicación.	

Figura 2.38: *Taxonomía de los patrones de interacción*

Para llevar a cabo el proceso de creación del catálogo de patrones de interacción del sistema CONEST fue necesario realizar varias iteraciones de este proceso, tal como puede observarse en la Figura 2.39, con el fin de obtener un diagrama claro y legible de las relaciones entre los patrones de interacción que describen las IU del sistema CONEST. En la primera iteración se construyó el catálogo de patrones de interacción incluyendo todos los patrones y relaciones existentes en el sistema, lo que tuvo como resultado una imagen difícil de leer y en muchos casos carente de sentido. En la segunda iteración, se agruparon los patrones de acuerdo a las IU donde se encontraban dentro del sistema, el resultado de esta agrupación permitió simplificar significativamente el número de patrones y relaciones visibles dentro del catálogo. Durante la tercera iteración, para simplificar la cantidad de líneas que determinan las relaciones entre los patrones de interacción del catálogo de patrones de interacción del sistema, se decidió aplicar un código de colores que simula el funcionamiento de un semáforo y permite

simplificar las líneas de relación a través de la permisología de los usuarios del sistema. Los patrones denotados con el color verde pueden ser accedidos cualquier rol de usuario del sistema. Los patrones denotados con el color amarillo pueden ser accedidos por los docentes y administradores. Por último, los patrones denotados con el color rojo, sólo pueden ser accedidos por los administradores.



Figura 2.39: *Iteraciones de la construcción del catálogo de componentes de IU*

El catálogo de componentes resultante de las iteraciones descritas anteriormente puede observarse en la Figura 2.40 y describe una versión simplificada de las relaciones existentes entre los patrones de interacción del sistema, ya que cada patrón de tipo tarea descrito en esta figura agrupa un conjunto de patrones de tipo tarea relacionados entre sí.

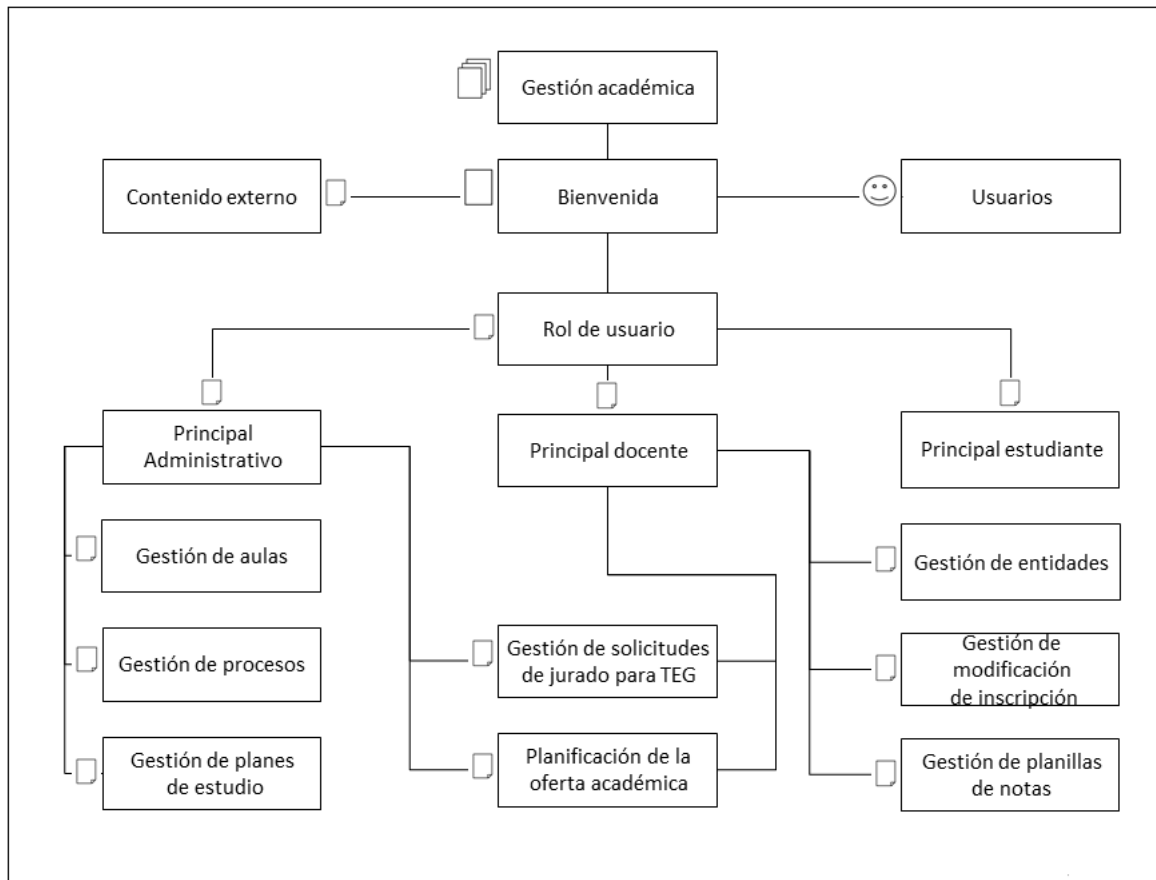


Figura 2.40: *Catálogo de patrones del sistema CONEST*

Este catálogo de patrones es una versión simplificada, ya que el original posee un gran número de patrones y relaciones que dificultan su lectura. Con el fin de facilitar la comprensión de las relaciones presentes en el sistema, para este trabajo se propone dividir el catálogo de patrones en dos niveles. En el primer nivel se presentan los patrones que describen funcionalidades generales, y posteriormente se describen en detalle los patrones que los componen estas funcionalidades. Adicionalmente se propone identificar los patrones de tipo tarea por medio de una notación de semáforo, la cual define los tipos de usuario que utilizan los patrones de manera similar a las restricciones que denotan los colores de un semáforo. El color verde indica que un patrón, puede ser accedido por cualquier usuario del sistema. El color amarillo indica que sólo los usuarios de tipo docente y administrativo pueden acceder a este patrón. Por último, los patrones denotados con el color rojo sólo pueden ser accedidos por el usuario administrador del sistema. Una vez explicada esta notación, a continuación se describen los patrones de tipo tarea del segundo nivel.

En la Figura 2.41, se detallan los patrones que componen el patrón principal estudiante, entre los cuales se tienen preinscripción, inscripción y modificación de inscripción, los

cuales a su vez se relacionan con el patrón consultar requisitos de asignatura.

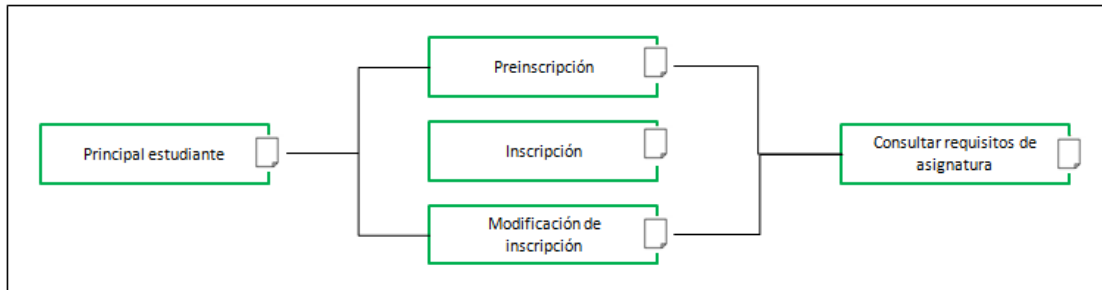


Figura 2.41: *Principal estudiante*

En la Figura 2.42 se observa el patrón gestionar planes de estudio, el cual está relacionado con el patrón gestionar asignatura.

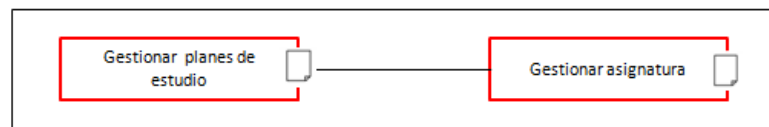


Figura 2.42: *Gestionar de planes de estudio*

En la Figura 2.43, se detalla el patrón principal docente, el cual se relaciona con los componentes calificar, consultar preinscripción y consultar horarios libres.

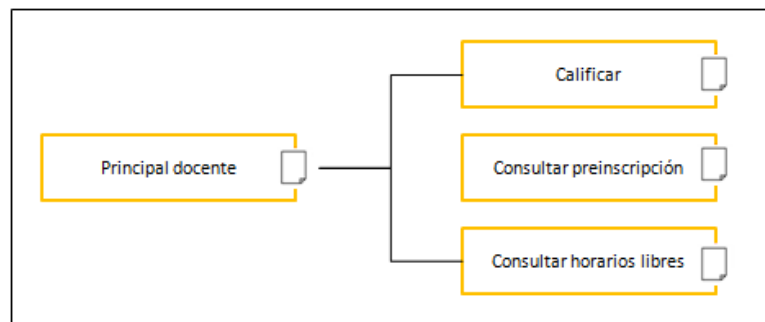


Figura 2.43: *Principal docente*

En la Figura 2.44 se observan los patrones que se relacionan con el patrón gestionar procesos, entre los cuales se tienen configurar planificación, preinscripción, inscripción, modificación, calificación y graduación

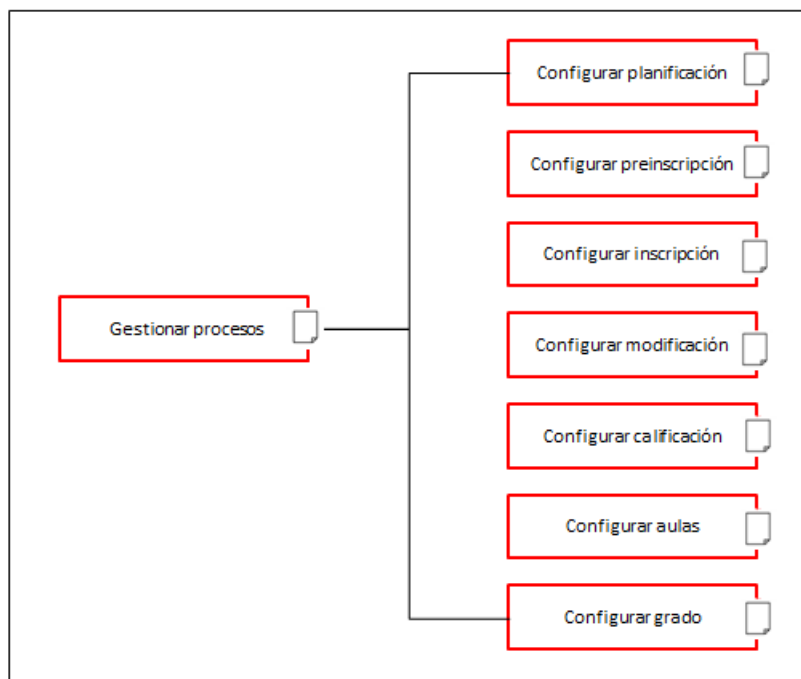


Figura 2.44: *Gestionar procesos*

En la Figura 2.45 puede observarse el patrón gestionar aulas, el cual está relacionado con los patrones generar mapa de aulas y aulas sin asignar.

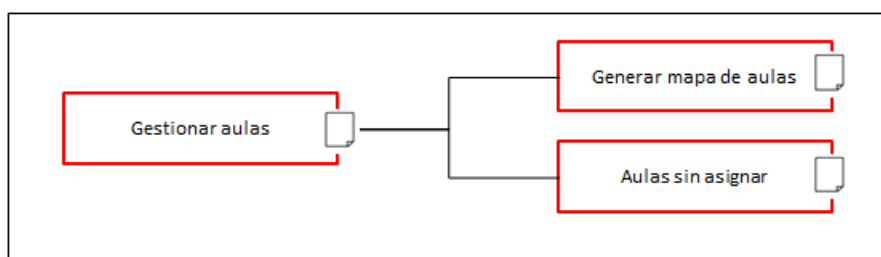


Figura 2.45: *Gestionar aulas*

En la Figura 2.46 se observan los patrones relacionados con el patrón gestionar planilla de notas, los cuales son solicitar, resumen, editar, anular y enviar correo.

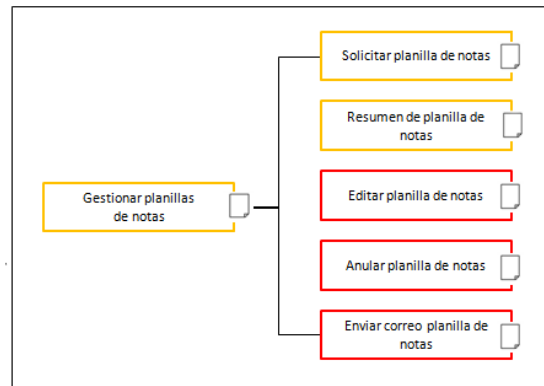


Figura 2.46: *Gestionar planillas de notas*

En la Figura 2.47, se detallan los patrones relacionados con el patrón principal administrativo, entre los cuales se presentan cambiar período académico, gestionar calendario, crear nuevo período académico y registrar título.

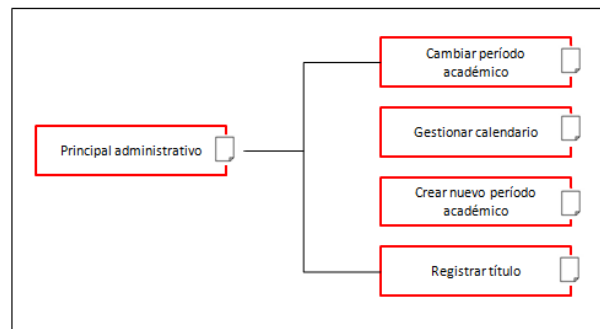


Figura 2.47: *Principal administrativo*

En la Figura 2.48 se observan los patrones que se relacionan con el patrón gestionar modificación de inscripción, entre los cuales se tienen resumen, anular y procesar modificación de inscripción.

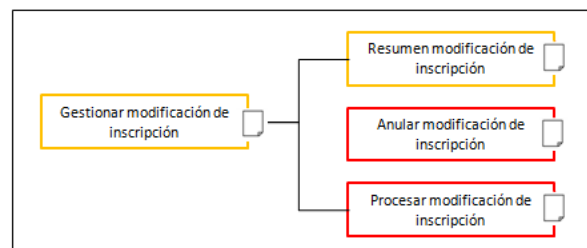


Figura 2.48: *Gestionar modificación de inscripción*

En la Figura 2.49 se observan los patrones relacionados con el patrón gestionar solicitud jurado TEG, entre los cuales se tienen agregar jurado externo, solicitar jurado, resumen de solicitud, editar solicitud, resumen de aprobación, aprobar solicitud, anular solicitud, enviar correo de aprobación, editar aprobación y enviar correo de jurado TEG.

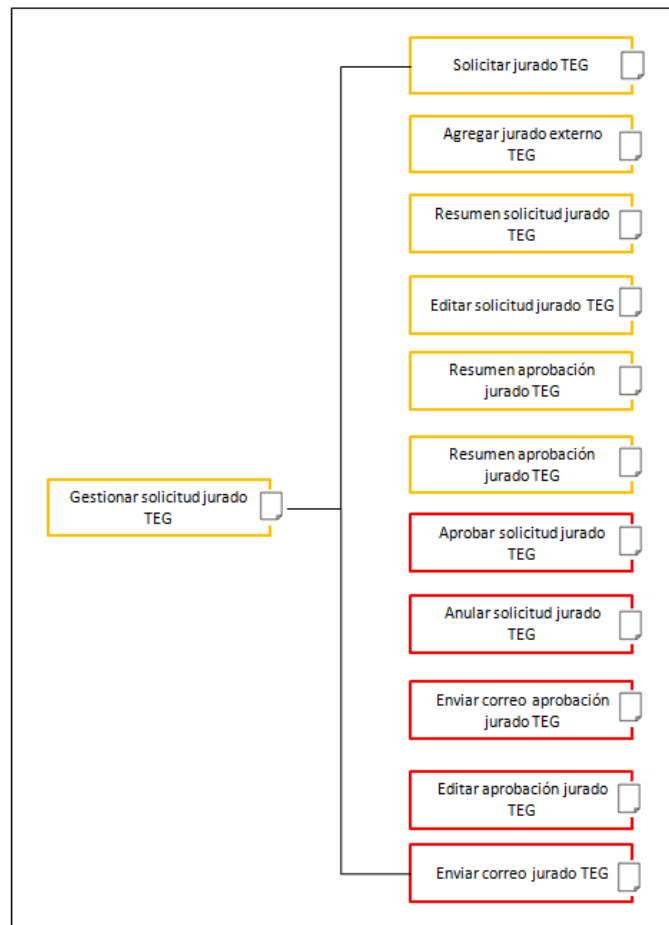


Figura 2.49: *Gestionar solicitud jurado TEG*

En la Figura 2.50 se observan los patrones relacionados con el patrón gestionar planificación por licenciatura, entre los cuales se encuentran consultar planificación de la oferta académica, listado de estudiantes por sección, ver horario asignatura, crear nueva sección, modificar horario asignatura, modificar campo simple, crear nueva asignatura, modificar capacidad asignatura, modificar disponibilidad asignatura, y modificar preinscripción asignatura.

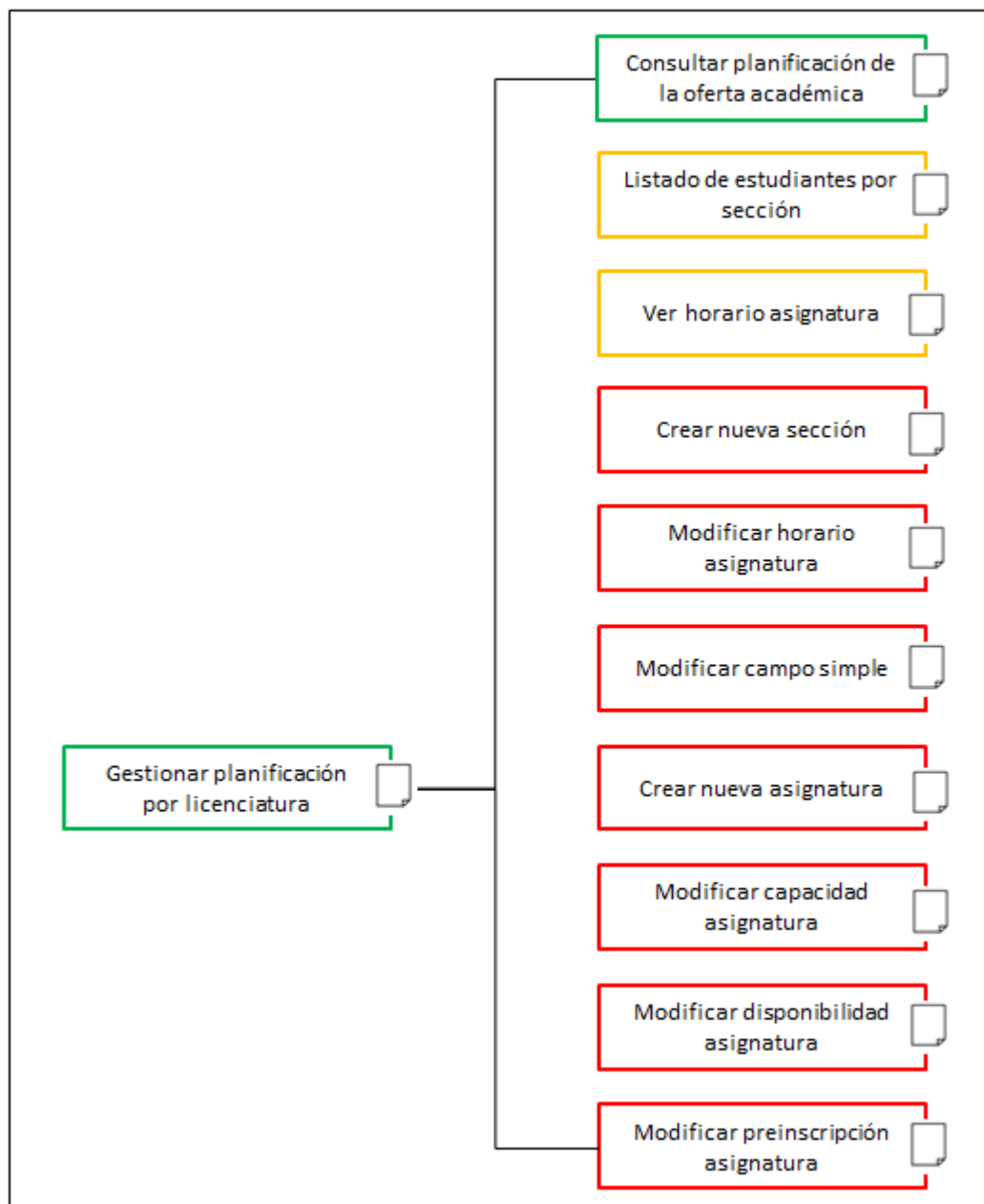


Figura 2.50: *Gestionar planificación oferta académica*

Por último, en la Figura 2.51 se observan los patrones que se relacionan el patrón gestionar entidad, entre los cuales se encuentran datos personales, historial académico, cambiar contraseña, chequeo de requisitos, cambiar foto, historial de fotos, consultar correo, constancias, bitácora y reestablecer contraseña.

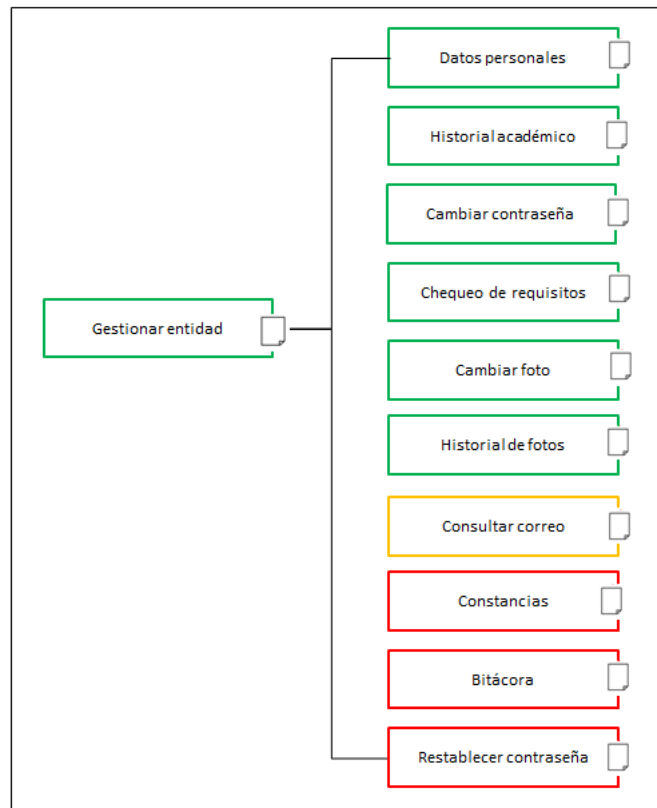



Figura 2.51: *Gestionar entidad*

Es posible observar a través del análisis que se realizó de las interfaces de usuario del sistema CONEST que el número de patrones de interacción resultante es muy grande, debido a la complejidad y el tamaño del sistema CONEST, por esta razón, se enunciaron algunos patrones de interacción con el objetivo de ilustrar el proceso de creación de los mismos, los cuales se presentan a continuación.

Tabla 2.1: Patrón de interacción - Gestión académica

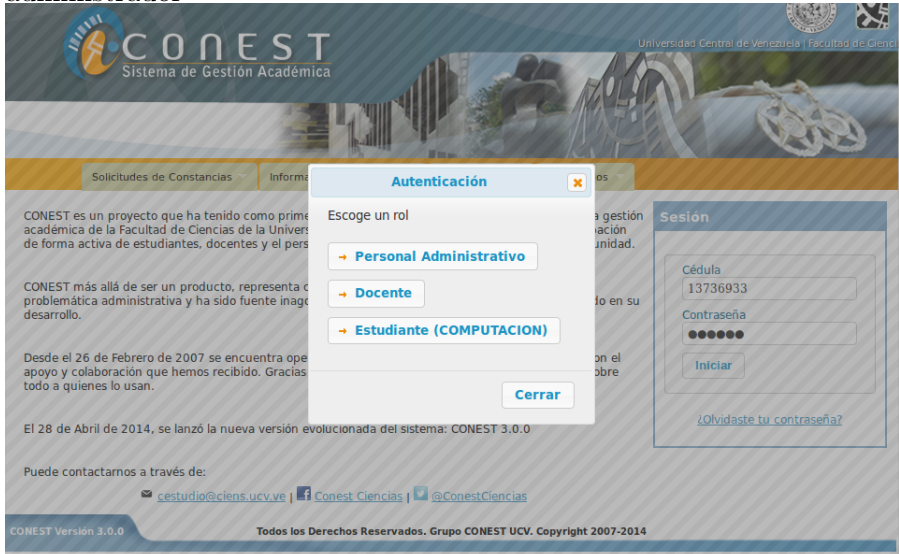
Nombre	Gestión académica 
Descripción	La comunidad de la Facultad de Ciencias de la Universidad Central de Venezuela requiere llevar a cabo los procesos y actividades académicas relacionadas con la gestión de los estudiantes, docentes y personal administrativo de manera eficiente, eficaz y automatizada.
Patrones relacionados	CONEST

CAPÍTULO 2. ANÁLISIS DE LAS INTERFACES DE USUARIO DE CONEST

Tabla 2.2: Patrón de interacción - Estudiante


Nombre	Estudiante □
Descripción	Los estudiantes de la Facultad de Ciencias son jóvenes con edades comprendidas entre 15-30 años, de ambos sexos y con conocimientos generales de la utilización del computador, los cuales frecuentemente utilizan las tecnologías Web, por lo que se interesan por los avances tecnológicos aplicados en el sistema.

Tabla 2.3: Patrón de interacción - Rol de usuario

Nombre	Rol de usuario □
Problema	<p>El usuario requiere determinar el rol con el cual desea acceder a CONEST. Los roles de usuario existentes son: estudiantes, docente y administrador</p> 
Solución	Mostrar una ventana modal en la cual el usuario puede escoger el rol de usuario a utilizar
Contexto	El usuario introdujo sus datos de autenticación y posteriormente debe elegir un rol de usuario con el cual realizar la autenticación en el sistema
Fuerzas	Los usuarios pueden acceder al sistema con un rol a la vez, para acceder con otro rol requieren de una acción adicional
Usabilidad	Reducción de carga cognitiva, facilidad de utilización
Consecuencias	Se le otorga el acceso al usuario a la interfaz de usuario principal del rol seleccionado
Patrones relacionados	Principal estudiante, Principal docente, Principal administrador

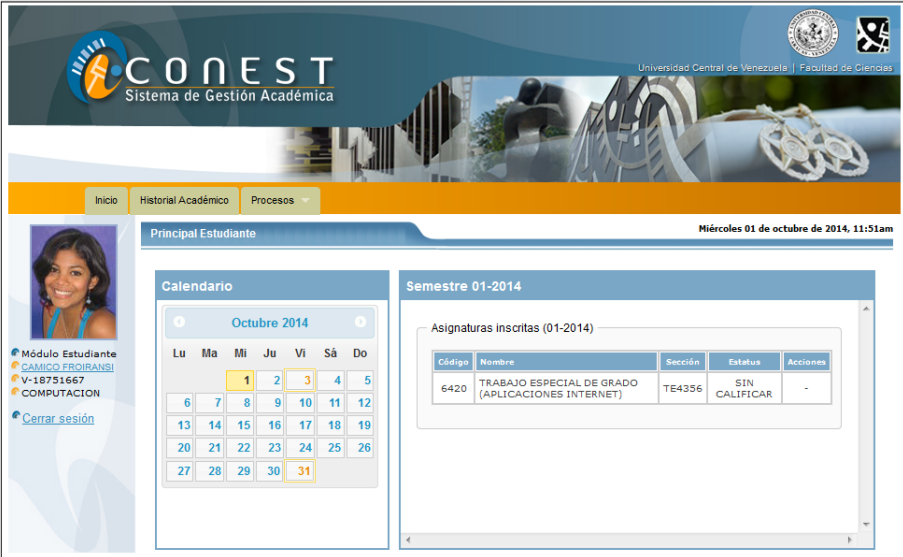
CAPÍTULO 2. ANÁLISIS DE LAS INTERFACES DE USUARIO DE CONEST

Tabla 2.4: Patrón de interacción - CONEST

Nombre	CONEST □
Problema	La comunidad de la Facultad de Ciencias requiere un sistema que permita la automatización de los procesos relacionados con la gestión académica de la Facultad de Ciencias, tales procesos son: inscripción, calificación y grado.
Solución	<p>Crear un sistema de gestión académica automatizado denominado CONEST, que incorpore nuevas tecnologías y técnicas, tales como AJAX, SPA (<i>Single Page Application</i>), y funcionalidades adicionales, como gestión de aulas, estudiantes, docentes y asignaturas, entre otros, que permitan obtener un sistema usable y mantenible a través del tiempo que incorpore algunos aspectos de accesibilidad. Adicionalmente este sistema debe incorporar la divulgación de información al público general, incorporando otras funcionalidades privadas a través de la autenticación del usuario.</p> 
Contexto	El usuario se encuentra en la página de inicio del sistema CONEST
Fuerzas	Los principales procesos llevados a cabo por la División de Control de Estudios pueden realizarse en línea a través del sistema CONEST, disminuyendo los recursos utilizados
Usabilidad	Facilidad de uso, Tolerancia a errores, Satisfacción al usuario
Consecuencias	El usuario puede consultar las diferentes ventanas modales informativas, que contienen la información general relacionada con las actividades llevadas a cabo por la división de control de estudios, tales como solicitudes de constancias, información sobre los procesos de inscripción, retiro de asignaturas, entre otros. También puede autenticarse para acceder a funcionalidades adicionales, tales como: preinscripción, gestión de historial académico, solicitud de constancias y gestión de solicitudes estudiantiles .
Patrones relacionados	Rol de usuario, Principal estudiante, Principal docente, Principal administrador, Modal informativa, Olvido de contraseña, Enlace de contacto

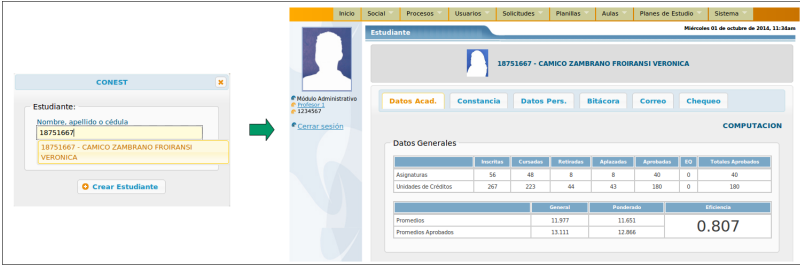
CAPÍTULO 2. ANÁLISIS DE LAS INTERFACES DE USUARIO DE CONEST

Tabla 2.5: Patrón de interacción - Principal estudiante

Nombre	Principal estudiante □										
Problema	Un estudiante de la Facultad de Ciencias requiere utilizar las funcionalidades (Preinscripción, Inscripción, Modificación de inscripción, Historial académico, perfil de usuario, Planificación) del sistema CONEST de una manera organizada										
Solución	<p>Mostrar una interfaz de usuario principal para el rol de estudiante que encapsule de manera organizada las funcionalidades que puede realizar el estudiante, tales como consultar horario actual, realizar preinscripción, inscripción y modificación de inscripción, realizar retiro de asignaturas y modificar perfil de usuario.</p>  <p>The screenshot displays the 'Principal Estudiante' interface. At the top, there is a navigation bar with 'Inicio', 'Historial Académico', and 'Procesos'. Below this, a user profile section shows a photo and details for 'CAMILIO FROIRANSI' (V-18751667) in the 'COMPUTACION' module. The main content area is divided into two panels: 'Calendario' (Calendar) for October 2014 and 'Semestre 01-2014' (Semester 01-2014). The calendar shows dates from 1 to 31. The semester panel contains a table of enrolled courses:</p> <table border="1" data-bbox="909 1081 1331 1155"> <thead> <tr> <th>Código</th> <th>Nombre</th> <th>Sección</th> <th>Estatus</th> <th>Acciones</th> </tr> </thead> <tbody> <tr> <td>6420</td> <td>TRABAJO ESPECIAL DE GRADO (APLICACIONES INTERNET)</td> <td>TE4356</td> <td>SIN CALIFICAR</td> <td>-</td> </tr> </tbody> </table>	Código	Nombre	Sección	Estatus	Acciones	6420	TRABAJO ESPECIAL DE GRADO (APLICACIONES INTERNET)	TE4356	SIN CALIFICAR	-
Código	Nombre	Sección	Estatus	Acciones							
6420	TRABAJO ESPECIAL DE GRADO (APLICACIONES INTERNET)	TE4356	SIN CALIFICAR	-							
Contexto	El usuario se encuentra autenticado en el sistema con el rol de estudiante										
Fuerzas	El sistema ofrece una interfaz de usuario unificada y adaptada cronológicamente a la ocurrencia de los procesos durante el semestre										
Usabilidad	Satisfacción al usuario, eficiencia, divulgación progresiva										
Consecuencias	Una vez que el usuario se encuentre en la página principal del sistema podrá realizar las siguientes actividades: consultar perfil de usuario, planificación por licenciatura, historial académico (del mismo estudiante). Adicionalmente puede llevar a cabo los procesos de preinscripción, inscripción y modificación de inscripción										
Patrones relacionados	Preinscripción, Inscripción, Modificación de inscripción, Perfil de usuario, Consultar usuario, Planificación por licenciatura										

CAPÍTULO 2. ANÁLISIS DE LAS INTERFACES DE USUARIO DE CONEST

Tabla 2.6: Patrón de interacción - Consultar usuario

Nombre	Consultar usuario □																																																								
Problema	Un usuario del sistema requiere consultar información (datos personales, historial académico, correo electrónico, chequeo de requisitos, bitácora, constancias, historial de fotos) de otro usuario.																																																								
Solución	<p>Mostrar una interfaz de usuario que recopile toda la información relacionada con el usuario a consultar. Esta información es limitada según los privilegios del usuario. Una vez consultada una entidad se pueden llevar a cabo diferentes actividades, dependiendo del rol de usuario.</p>  <p>The screenshot shows two parts of the system. On the left, a search form titled 'CONEST' allows entering a student's name, ID, or email. A search for '18751667 - CAMICO ZAMBRANO FROIRANSI VERONICA' is shown. On the right, the profile page for this student is displayed, including a navigation menu (Datos Acad., Constancia, Datos Pers., Bitácora, Correo, Chequeo) and a 'COMPUTACION' table with the following data:</p> <table border="1"> <thead> <tr> <th colspan="7">COMPUTACION</th> </tr> <tr> <th>Datos Generales</th> <th>Asignaturas</th> <th>Completadas</th> <th>Reprobadas</th> <th>Aprobadas</th> <th>Requisitos</th> <th>Promedio</th> </tr> </thead> <tbody> <tr> <td>Asignaturas</td> <td>56</td> <td>48</td> <td>8</td> <td>0</td> <td>40</td> <td>0.71</td> </tr> <tr> <td>Unidades de Créditos</td> <td>267</td> <td>223</td> <td>44</td> <td>43</td> <td>180</td> <td>0</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Promedio</td> <td></td> <td>13.977</td> <td></td> <td>13.651</td> <td></td> <td>0.807</td> </tr> <tr> <td>Promedio Aprobados</td> <td></td> <td>13.111</td> <td></td> <td>12.866</td> <td></td> <td></td> </tr> </tbody> </table>	COMPUTACION							Datos Generales	Asignaturas	Completadas	Reprobadas	Aprobadas	Requisitos	Promedio	Asignaturas	56	48	8	0	40	0.71	Unidades de Créditos	267	223	44	43	180	0															Promedio		13.977		13.651		0.807	Promedio Aprobados		13.111		12.866		
COMPUTACION																																																									
Datos Generales	Asignaturas	Completadas	Reprobadas	Aprobadas	Requisitos	Promedio																																																			
Asignaturas	56	48	8	0	40	0.71																																																			
Unidades de Créditos	267	223	44	43	180	0																																																			
Promedio		13.977		13.651		0.807																																																			
Promedio Aprobados		13.111		12.866																																																					
Contexto	El usuario se encuentra autenticado en el sistema con el rol de administrador o docente.																																																								
Fuerzas	La interfaz de usuario agrupa toda la información relacionada al usuario consultado, por lo que reduce la cantidad de búsquedas adicionales.																																																								
Usabilidad	Divulgación progresiva, satisfacción al usuario, eficiencia																																																								
Consecuencias	<p>Una vez que se consulta un usuario se pueden acceder a diferentes funcionalidades que contienen de manera organizada la información del usuario. A continuación se detallan estas funcionalidades:</p> <ul style="list-style-type: none"> -Rol de estudiante: pueden consultar los datos personales, historial académico, chequeo de requisitos, cambio de contraseña, cambio de foto e historial de fotos del mismo usuario. -Rol de docentes: pueden consultar los datos personales, historial académico, chequeo de requisitos, historial de fotos y correo electrónico de cualquier estudiante de la Facultad de Ciencias. -Rol de administrador: puede consultar y modificar toda la información de una entidad, así como puede consultar la bitácora y restablecer la contraseña del usuario. Adicionalmente puede emitir constancias. 																																																								
Patrones relacionados	Consultar historial académico, Reiniciar contraseña																																																								

Por medio del estudio de las interfaces de usuario del sistema CONEST se determinaron los componentes de interfaz de usuario existentes y se creó el catálogo de patrones de interacción relacionados, el cual representa el insumo que será utilizado para la creación del sistema CONEST Desarrollo y permitiendo obtener una descripción del comportamiento general de las IU que provee el sistema CONEST.

Capítulo 3 Implementación del sistema CONEST Desarrollo

En este capítulo se describe el proceso de desarrollo del sistema CONEST desarrollo, el cual se llevo a cabo a través de la metodología AgilUs. Este capítulo se divide en cuatro fases: Requerimientos, Análisis, Prototipaje y Entrega. En primer lugar se describen las actividades y artefactos elaborados en la fase de requerimientos, entre los cuales se tienen tormenta de ideas, evaluación de sistema existentes, perfil de usuario y requerimientos funcionales y no funcionales. En segundo lugar se describen las actividades elaboradas durante la fase de análisis, tales como modelos de casos de uso y objetos del dominio, prototipo en papel, guía de estilo y patrones de interacción. En tercer lugar se describe el proceso de prototipaje del sistema CONEST Desarrollo, en el cual se describen las dos iteraciones llevadas a cabo durante esta fase y las actividades elaboradas en cada una de ellas. Por último se describe la fase de entrega, en la cual se colocó en producción el sistema CONEST Desarrollo, sobre el cual se aplicaron las pruebas de aceptación del sistema.

Para la implementación de este Trabajo Especial de Grado se llevó a cabo una iteración del ciclo de desarrollo de la metodología AgilUs, y en la fase de prototipaje se llevaron a cabo dos iteraciones, tal y como se observa en la Figura 3.1.

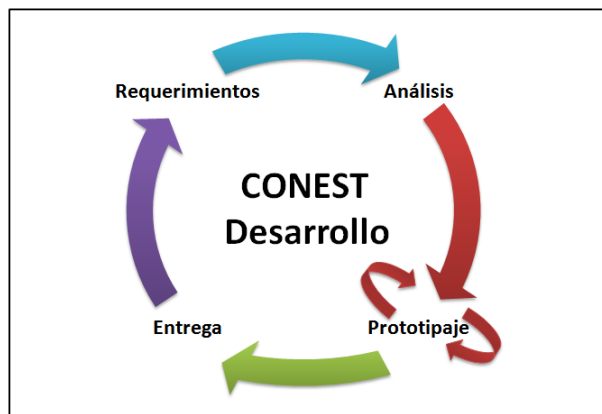


Figura 3.1: Iteraciones de la implementación del sistema CONEST Desarrollo

- Estándares: es necesario definir los estándares que deben cumplir los desarrolladores a la hora de crear nuevas interfaces de usuario.
- Guías de estilo: se debe establecer una guía de estilo general que contenga las características de la apariencia que poseen las interfaces de usuario provistas por este sistema, como por ejemplo: esquema de colores, tipografía, entre otros.
- Interfaz de usuario: para la creación de nuevas funcionalidades en el sistema CONEST es importante la creación de interfaces de usuario estandarizadas.
- Guías de instalación: desarrollar funcionalidades en el sistema CONEST requiere un ambiente de desarrollo específico, por lo que es necesario proveer una guía de instalación del ambiente.
- Patrón de interacción: para definir la interacción de las diferentes pantallas del sistema es necesario generar un catálogo de patrones de interacción del sistema.
- Guías de diseño: para agrupar los patrones de interacción junto con la apariencia de las interfaces de usuario se propone la creación de una guía de diseño.
- Componentes: la agrupación de los elementos de interfaz se debe realizar a través de un catálogo de componentes.
- Desarrolladores: los usuarios potenciales de esta aplicación son los nuevos desarrolladores del sistema CONEST.

Con el fin de conocer soluciones exitosas en el contexto del problema de esta investigación, se estudiaron sistemas similares que permitieron proveer al usuario una guía de diseño general en el desarrollo de aplicaciones Web e interfaces de usuario. Esta evaluación es descrita a continuación.

3.1.2. Evaluación de sistemas existentes

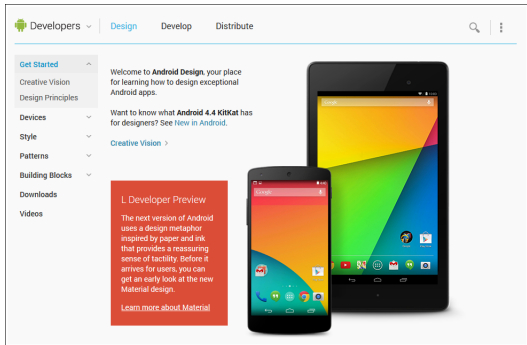
Se realizó el análisis de guías de diseño existentes orientadas a diferentes tipos de aplicaciones, con la finalidad de detallar las principales funcionalidades ofrecidas y las ventajas que provee su uso. En primer lugar se estudiaron las guías de diseño de Android, iOS y Google, con el objetivo de determinar las características que toman en cuenta estas organizaciones para el desarrollo de aplicaciones móviles. En segundo lugar, se estudiaron las guías de diseño orientadas a aplicaciones Web ó *frameworks* CSS, con el fin de determinar las tendencias actuales en el desarrollo de aplicaciones Web.

Los resultados del análisis de las guías de diseño orientadas a dispositivos móviles se describen a continuación.

CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA CONEST DESARROLLO

1. Android es el sistema operativo basado en el *kernel* de Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas. En la Tabla 3.1 se detalla el estudio de la guía de diseño de este sistema operativo.

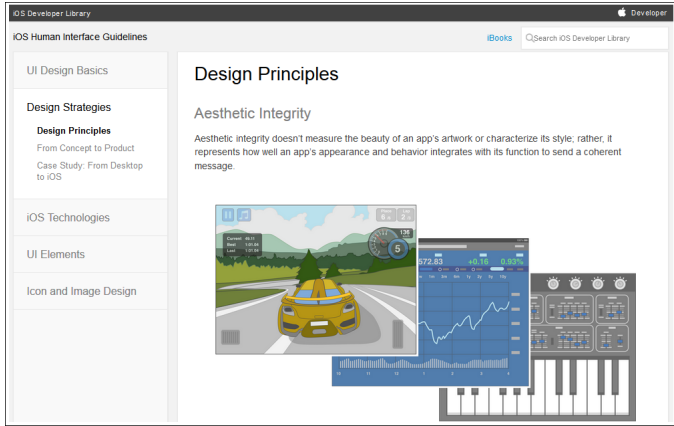
Tabla 3.1: Evaluación de sistemas existentes: Android Design

Nombre	Android Design
Descripción	Android Design es una aplicación web que incorpora una guía de diseño que provee a los desarrolladores de esta plataforma los recursos y funcionalidades necesarios para la construcción de aplicaciones de calidad. Adicionalmente la comunidad de Android provee guías de desarrollo y distribución de aplicaciones.
Funcionalidades	Provee una guía de inicio que incorpora la visión creativa detrás de las aplicaciones de esta plataforma y los principios de diseño a seguir. Facilita una guía para el diseño de aplicaciones para diferentes dispositivos y tamaños de pantallas. Contiene una guía de estilo que incluye temas, tipografías, colores, iconos, entre otros. Propone un conjunto de patrones de comportamiento asociados a diferentes componentes de interfaz de usuario ofrecidas por el sistema. Facilita principios de accesibilidad y herramientas que permiten crear aplicaciones accesibles. Por último, incorpora un conjunto de componentes de interfaz que pueden utilizarse en el desarrollo de cualquier aplicación en Android.
Personalización	Es posible descargar diferentes paquetes predefinidos, tales como esquemas de colores, íconos y ejemplos de aplicaciones.
Comunicación	Provee comunicación con la comunidad de desarrolladores de Android a través de la red social Google+ y a través de un foro de soporte técnico ubicado dentro de su página Web.
Apariencia	 <p>The screenshot shows the Android Design website interface. It features a navigation bar with 'Developers', 'Design', 'Develop', and 'Distribute' tabs. A sidebar on the left contains a 'Get Started' section with links to 'Creative Vision', 'Design Principles', 'Devices', 'Style', 'Patterns', 'Building Blocks', 'Downloads', and 'Videos'. The main content area displays a welcome message: 'Welcome to Android Design, your place for learning how to design exceptional Android apps.' Below this, there is a promotional card for 'L Developer Preview' with the text: 'The next version of Android uses a design metaphor inspired by paper and ink that provides a reassuring sense of familiarity. Before it arrives for users, you can get an early look at the new Material Design. Learn more about Material.' The background of the main content area shows two Android devices (a smartphone and a tablet) displaying colorful, abstract designs.</p>
Idiomas	Inglés, Español, Ruso, Chino
Opinión como usuario	Es una guía de diseño extensa que provee todas las herramientas necesarias para el desarrollo de aplicaciones en esta plataforma.

CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA CONEST DESARROLLO

2. iOS es un sistema operativo móvil de la empresa Apple Inc. Facilita a los desarrolladores un conjunto de buenas prácticas para mejorar la experiencia del usuario por medio de una guía de diseño que incorpora diferentes tecnologías disponibles para esta plataforma, tales como iCloud, VoiceOver, entre otros. En la Tabla 3.2 se describe esta guía de diseño.

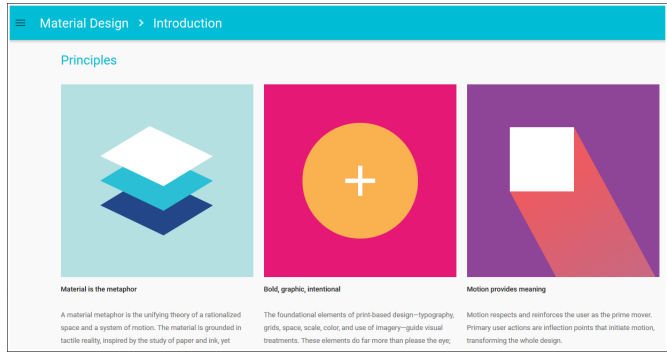
Tabla 3.2: Evaluación de sistemas existentes: iOS Human Interface Guideline

Nombre	iOS Human Interface Guideline
Descripción	La guía de diseño para interfaces de usuario del sistema operativo iOS consiste en un conjunto de buenas prácticas para el diseño de aplicaciones para esta plataforma.
Funcionalidades	Facilita los principios básicos para el diseño de interfaces de usuario en iOS, tales como la interacción y la retroalimentación, animaciones, terminología, navegación, entre otros. Proporciona un conjunto de recomendaciones y estrategias para el diseño de aplicaciones exitosas. Provee un manual de referencia de las tecnologías que proveen los dispositivos Mac, tales como acelerómetro, teclados, sonidos, entre otros. Incorpora algunos ejemplos de elementos de interfaz básicos como menús, barras de navegación y vistas temporales. Por último anexa las medidas recomendadas para los principales componentes de interfaz presentes en la mayoría de las aplicaciones, tales como botones, imágenes, entre otros.
Personalización	Provee la guía de diseño en formato PDF.
Comunicación	No provee información a través de las redes sociales.
Apariencia	
Idiomas	Inglés
Opinión como usuario	Es una guía de diseño limitada que provee recomendaciones para el diseño de componentes de interfaz de aplicaciones de esta plataforma.

CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA CONEST DESARROLLO

3. Google Inc. es una empresa estadounidense especializada en productos y servicios de software. La guía de diseño propuesta por Google incorpora manuales de diseño de componentes de interfaz, programas para desarrolladores Web, entre otros. Esta guía de diseño es descrita en la Tabla 3.3.

Tabla 3.3: Evaluación de sistemas existentes: Google Design

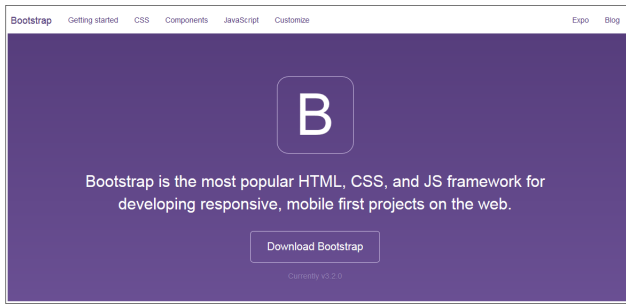
Nombre	Google Design
Descripción	Google Design es una guía de diseño propuesta por la compañía Google, en la cual describen las directrices de diseño que utilizan en el desarrollo de sus proyectos. Google propone entre sus pautas centrarse en el usuario para alcanzar exitosamente la consecución de los objetivos.
Funcionalidades	Google Design provee entre sus características una guía de diseño que incluye una guía de estilo que considera los aspectos más importantes a la hora de desarrollar aplicaciones Web, tales como el esquema de colores, componentes de interfaz, patrones de gestos y comportamiento, entre otros. Adicionalmente provee un breve manual para el diseño de material gráfico, en el cual se encuentra un conjunto de buenas prácticas en el diseño de metáforas y componentes de interfaz.
Personalización	Esta guía de diseño provee un conjunto de descargas, tales como plantillas para diferentes dispositivos, esquemas de colores para utilizar a través de Photoshop, componentes de interfaz, entre otros.
Comunicación	Esta guía de diseño actualmente se encuentra en Google+ como vía de comunicación con la comunidad que la utiliza.
Apariencia	
Idiomas	Inglés
Opinión como usuario	Esta guía de diseño contiene las directrices necesarias para iniciar el desarrollo de un producto. Facilita la descarga de diferentes módulos, tales como plantillas de diferentes dispositivos, tipografías, paletas de colores, entre otros. Resulta intuitivo y fácil de utilizar.

CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA CONEST DESARROLLO

Los resultados del análisis de las guías de diseño orientadas a aplicaciones Web se describen a continuación.

1. Bootstrap es un *framework* de software libre para el diseño de aplicaciones web y contiene plantillas de diseño con diferentes elementos de diseño. En la Tabla 3.4 se observa la descripción de este *framework*.

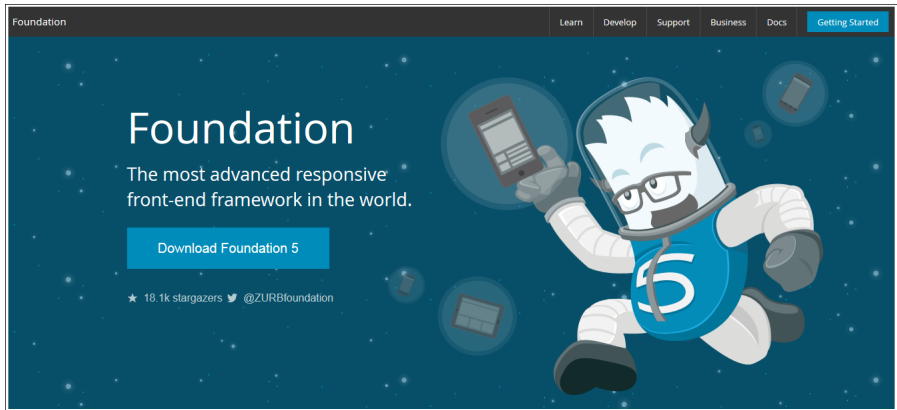
Tabla 3.4: Evaluación de sistemas existentes: Bootstrap

Nombre	Bootstrap
Descripción	Bootstrap es una herramienta de código abierto que permite llevar a cabo el desarrollo de aplicaciones web y esta diseñado para que este desarrollo sea fácil y rápido, tomando en cuenta todos los niveles de experiencia y tipos de dispositivos
Funcionalidades	Provee una guía de inicio para la descarga y utilización del <i>framework</i> y una guía de los estilos CSS que define la apariencia de las interfaz de usuario de las aplicaciones. Adicionalmente provee un catálogo que describe los estilos de sus componentes de interfaz, los cuales combinan elementos HTML con el CSS que facilita el <i>framework</i> , lo que resulta en un componente de interfaz de usuario predefinido que puede ser reutilizado en las aplicaciones que utilicen el <i>framework</i> .
Personalización	Provee una interfaz de usuario que permite personalizar los módulos que se incluyen en la descarga. Es posible modificar los estilos del CSS y los componentes de interfaz de usuario predefinidos que contiene el <i>framework</i> .
Comunicación	Facilita un <i>blog</i> con información, noticias y actualizaciones que se realizan en los módulos del <i>framework</i> y utiliza la red social <i>Twitter</i> .
Apariencia	
Idiomas	Chino, Francés, Alemán, Italiano, Coreano, Ruso, Español, Ucraniano
Opinión como usuario	Provee la mayoría de los elementos necesarios para la creación de interfaces de usuario, lo cual representa una ventaja, ya que no es necesario complementar el framework con otros elementos externos pero puede limitar la creatividad del desarrollador

CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA CONEST DESARROLLO

2. Foundation es un *framework* para crear aplicaciones web. Contiene HTML y plantillas de diseño basadas en CSS para la tipografías, botones y otros componentes de interfaz, así como extensiones opcionales de JavaScript. En la Tabla 3.5 se observa en detalle las características del framework.

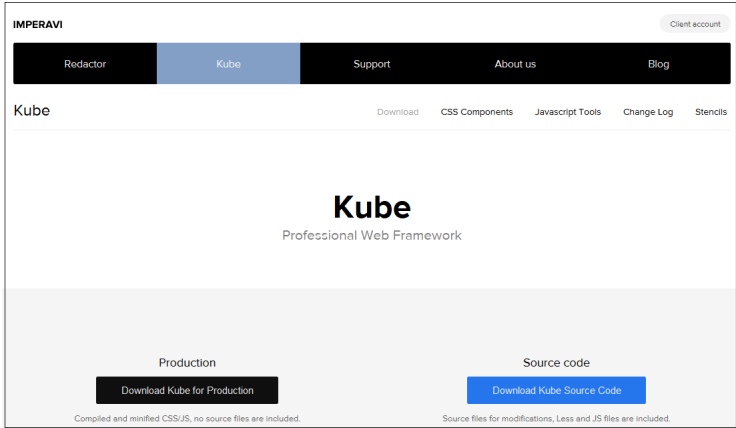
Tabla 3.5: Evaluación de sistemas existentes: Foundation

Nombre	Foundation
Descripción	Foundation es un <i>framework</i> CSS que permite la creación de sitios Web y aplicaciones de una manera más organizada, sencilla, rápida y eficaz. La herramienta cuenta con múltiples complementos disponibles para usar de manera fácil e intuitiva, así como también estilos preestablecidos para hacer el desarrollo más dinámico.
Funcionalidades	Facilita una guía de inicio que describe todas las características del código detrás del <i>framework</i> y las funcionalidades que provee, adicionalmente incluye una lista de ejemplos de sitios web diseñados utilizando esta herramienta. Posee una documentación extensa que incluye ejemplos de codificación e implementación.
Personalización	Provee un conjunto variado de descargas predeterminadas, que incluyen la descarga del <i>framework</i> completo, una versión simplificada con los módulos esenciales, una descarga personalizada donde el usuario determina los módulos que desea descargar y si desea modificar su contenido original.
Comunicación	Foundation utiliza las redes sociales Facebook y Twitter. Proporciona una sección de soporte técnico, donde provee diferentes vías de comunicación con el equipo de desarrollo del framework.
Apariencia	
Idiomas	Inglés
Opinión como usuario	Es un <i>framework</i> sencillo y adaptable a las necesidades del usuario, ya que provee el esquema general para la creación de aplicaciones Web y una mayor flexibilidad en el diseño de aplicaciones Web.

CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA CONEST DESARROLLO

3. Kube es un *framework* minimalista y de gran alcance para la construcción de sitios Web profesionales de forma rápida. En la Tabla 3.6 se observa la descripción en detalle de Kube.

Tabla 3.6: Evaluación de sistemas existentes: Kube

Nombre	Kube
Descripción	Kube es un <i>framework</i> CSS avanzado y flexible que otorga el poder de elección y creatividad al desarrollador, mientras toda la tecnología se encuentra detrás del <i>framework</i> .
Funcionalidades	Kube proporciona estilos CSS con funcionalidades dinámicas de <i>JavaScript</i> . Provee una guía de documentación de los componentes de interfaz del <i>framework</i> , la cual contiene ejemplos en diferentes contextos y muestra la implementación del código asociado.
Personalización	Existen dos formas de descargar el <i>framework</i> desde la página Web. La primera contiene el código fuente que permite la modificación de los archivos originales que componen el <i>framework</i> . La segunda opción es descargar una versión simplificada que únicamente contiene los archivos compilados del <i>framework</i> .
Comunicación	Kube utiliza la red social <i>Twitter</i> . Adicionalmente provee soporte técnico a través de un <i>blog</i> ubicado dentro de su página Web y a través de correo electrónico.
Apariencia	 <p>The screenshot shows the Kube website interface. At the top, there is a navigation bar with links for 'Redactor', 'Kube', 'Support', 'About us', and 'Blog'. Below the navigation bar, the main heading reads 'Kube' followed by 'Profesional Web Framework'. At the bottom of the page, there are two prominent buttons: 'Download Kube for Production' (black) and 'Download Kube Source Code' (blue). Small text at the very bottom indicates that the production version includes compiled and minified CSS/JS, while the source code version includes source files for modifications, Less, and JS files.</p>
Idiomas	Inglés
Opinión como usuario	Es un <i>framework</i> que provee la estructura general necesaria para el desarrollo de aplicaciones sin establecer una apariencia predefinida de la misma. Esta estructura permite ahorrar tiempo en el diseño general de las aplicaciones Web y permite al desarrollador realizar estilos y funcionalidades que se ajusten a las necesidades de la organización.

Es importante hacer énfasis en que este estudio realizado entre las guías de diseño

CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA CONEST DESARROLLO

orientadas aplicaciones móviles y las orientadas a aplicaciones Web permite determinar los lineamientos y propuestas que proponen éstas tendencias actuales. Aunque estas guías están diseñadas para ser utilizadas en ambientes diferentes, sugieren buenas prácticas y actividades en común que conllevan la construcción de guías de diseño que incorporan herramientas de utilidad para el desarrollo de productos de software. A continuación, en la Tabla 3.7 se realiza una comparación de las guías de diseño orientadas a aplicaciones móviles y aplicaciones Web estudiadas en esta sección, con el fin de determinar las similitudes entre las características y funcionalidades ofrecidas por cada una de éstas.

Tabla 3.7: Cuadro comparativo de sistemas existentes

Guía de diseño	Aplicaciones móviles			Aplicaciones Web		
	Android	iOS	Google	Bootstrap	Foundation	Kube
Provee componentes de interfaz pre-definidos, tales como formularios, botones y menús	✓	✓	✓	✓	✓	✓
Incorpora guías de estilo	✓	✗	✓	✓	✓	✓
Facilita ejemplos de implementación de funcionalidades	✗	✗	✗	✓	✓	✓
Proporciona guías o pautas sobre el comportamiento de los componentes de interfaz	✓	✓	✓	✓	✓	✗
Provee una guía de inicio para el desarrollo de aplicaciones	✗	✗	✗	✓	✓	✓
Permite la descarga de diferentes módulos y porciones de código, tales como archivos CSS y JavaScript	✓	✗	✓	✓	✓	✓

Luego de realizar el análisis de las guías de diseño de Android, iOS y Google, y de los *frameworks* CSS Bootstrap, Foundation y Kube, es posible concluir que en general para la construcción de software es necesario proveer a los programadores un conjunto de directrices generales que permitan mantener estandarizados los productos de software y

que provean al desarrollador una forma rápida y fácil de iniciar el proceso de desarrollo. Si bien estas guías no siempre deben cumplirse al pie de la letra, permiten a los desarrolladores más inexpertos tener un punto de partida para completar las actividades que deseen llevar a cabo y reducir la curva de aprendizaje del ambiente de desarrollo de un sistema. Las características tomadas en cuenta a partir de éste análisis que serán incorporadas a la implementación del sistema CONEST Desarrollo incluyen una guía de inicio, una guía de estilo y una sección de descargas, las cuales complementarán el catálogo de componentes de interacción del sistema CONEST.

3.1.3. Perfiles de usuario

Los usuarios potenciales de CONEST Desarrollo son los miembros de la comunidad de la Facultad de Ciencias, con conocimientos en el manejo de las Tecnologías de Información y Comunicación, en especial las aplicaciones Web. Los principales usuarios tienen el poder de tomar decisiones en la administración general del sistema, por lo que requieren conocer los procesos inherentes a la Gestión Académica de la Facultad de Ciencias. Adicionalmente deben contar con conocimientos generales en la gestión del sistema CONEST, con el fin de tomar decisiones en la gestión de usuarios y componentes pertenecientes al sistema CONEST Desarrollo.

3.1.4. Requerimientos funcionales y no funcionales

Luego de realizar el análisis de requerimientos a través de los artefactos de tormentas de ideas, evaluación de sistemas existentes y perfiles de usuario, descritos anteriormente, se obtuvo una lista de requerimientos funcionales para el desarrollo del sistema, los cuales se describen a continuación.

- Crear un inicio de sesión para autenticarse en el sistema CONEST Desarrollo.
- Consultar una guía de inicio en la cual se encuentre la información de la configuración del ambiente de desarrollo.
- Listar componentes de interfaz del catálogo del sistema CONEST.
- Consultar la guía de estilo donde se encuentre la información de la guía de estilo del sistema CONEST.
- Gestionar componentes de interfaz y usuarios del sistema CONEST Desarrollo.

Una vez establecidos los requerimientos funcionales del sistema, los requerimientos no funcionales que serán implementados para este desarrollo son: satisfacción al usuario, reducción de carga cognitiva, usabilidad y tolerancia a errores. Para este desarrollo se hará énfasis en el requerimiento no funcional de satisfacción al usuario. El cual será evaluado en la etapa de entrega a través de las pruebas de aceptación, las cuales son un artefacto propuesto por el método para esta etapa del ciclo de desarrollo.

3.2. Fase II: Análisis

En la etapa de Análisis de la metodología AgilUs se utilizaron los artefactos: modelo de casos de uso y objetos del dominio, prototipo en papel, guías de estilo y patrones de interacción, los cuales se describen a continuación.

3.2.1. Modelo de casos de uso

El modelo de casos de uso es una técnica de especificación de requerimientos del usuario. A continuación, en la Figura 3.3 se puede observar el modelo de casos de uso del sistema CONEST Desarrollo.

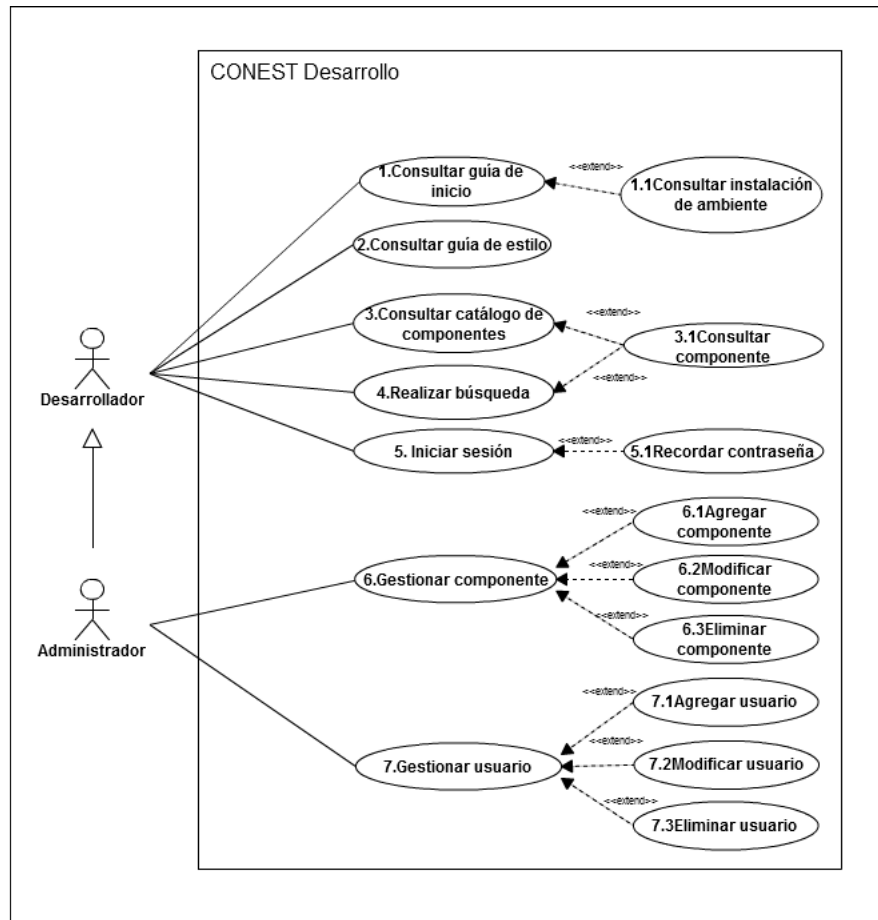


Figura 3.3: Modelo de casos de uso CONEST Desarrollo

Para este sistema se definen dos actores principales. El primer actor es el Administrador, el cual lleva a cabo la gestión de las entidades del sistema. El segundo actor es el Desarrollador, el cual realiza las operaciones de consulta de guías de inicio, guías de estilo, catálogo de componentes y componentes de IU.

Una vez presentados el diagrama de casos de uso se describen las especificaciones de los casos de uso, con el fin de representar las interacciones de los actores con el sistema.

- Consultar guía de inicio: permite al usuario consultar la guía de inicio del sistema CONEST, la cual incorpora los requerimientos generales necesarios para el ambiente de desarrollo del sistema.
- Consultar instalación de ambiente: contiene un directorio con las tecnologías utilizadas en la implementación del sistema CONEST, las cuales van desde el sistema operativo hasta el sistema manejador de base de datos.

- Consultar guía de estilo: proporciona al usuario la guía de estilo del sistema CONEST Desarrollo, la cual incluye el esquema de colores, tipografía, logotipos del sistema, entre otros.
- Consultar catálogo de componentes: permite al usuario consultar el catálogo de componentes de interfaz del sistema.
- Consultar componente: permite al usuario consultar un componente de interfaz perteneciente al catálogo de componentes del sistema.
- Realizar búsqueda: permite la búsqueda de componentes de interfaz en el sistema proporcionando la información relacionada.
- Iniciar sesión: provee al usuario la interfaz de autenticación en el sistema a través de un formulario de inicio de sesión.
- Recordar contraseña: facilita al usuario recuperar su contraseña en caso de olvido.
- Gestionar componente: permite realizar la gestión de un componente perteneciente al catálogo del sistema. Puede realizar diferentes acciones, tales como agregar, modificar o eliminar un componente.
- Agregar componente: permite agregar un nuevo componente al catálogo de componentes del sistema.
- Modificar componente: permite modificar un componente perteneciente al catálogo de componentes del sistema.
- Eliminar componente: facilita al usuario eliminar un componente del catálogo del sistema.
- Gestionar usuario: facilita la gestión de un usuario dentro del sistema. Puede realizar acciones tales como agregar, modificar o eliminar un usuario.
- Agregar desarrollador: permite agregar un nuevo usuario al sistema.
- Modificar usuario: permite modificar un usuario perteneciente al sistema.
- Eliminar usuario permite eliminar un usuario perteneciente al sistema.

3.2.2. Modelo de objetos del dominio

Un modelo de objetos del dominio describe los objetos del dominio de la aplicación y las relaciones estáticas entre estos. En la Figura 3.4 puede observarse los objetos que se especifican en este modelo, los cuales son: administrador, desarrollador, guía de estilo, guía de inicio, catálogo de componentes, componentes y descargas.

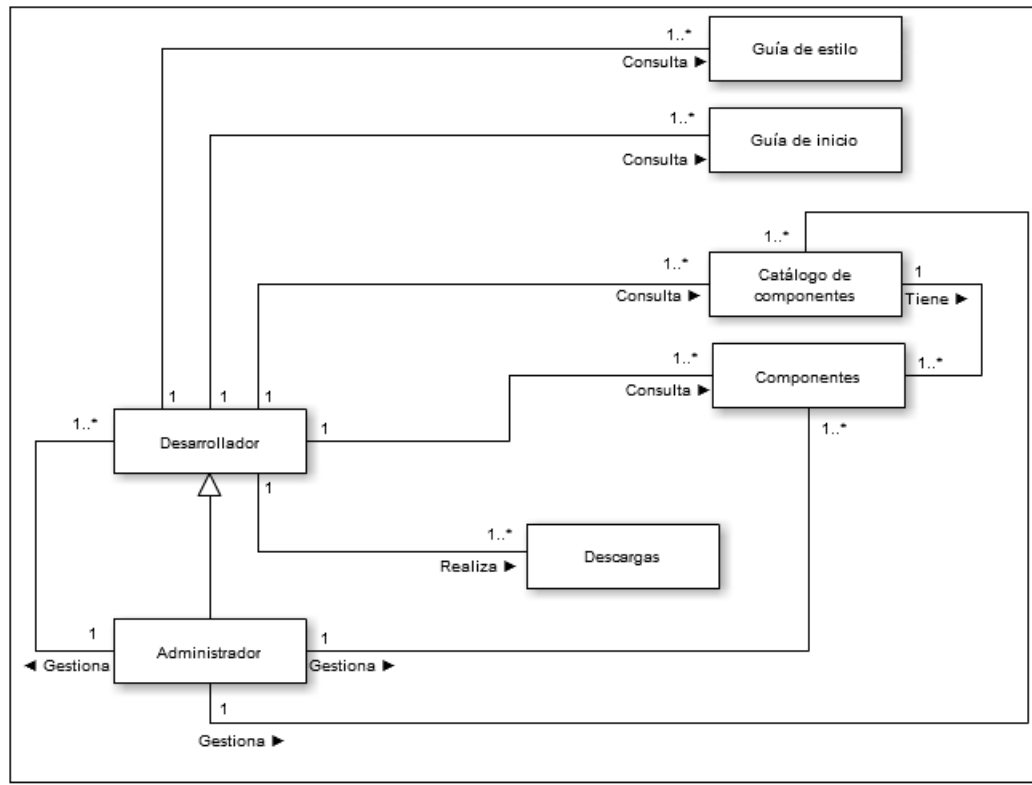


Figura 3.4: Modelo de objetos del dominio CONEST Desarrollo

Un desarrollador puede consultar dentro del sistema una guía de inicio, una guía de estilo, un catálogo de componentes y un componente en específico. Adicionalmente puede realizar descargas de archivos del sistema. Un administrador puede gestionar usuarios y componentes del sistema, y a su vez puede realizar las mismas acciones que el desarrollador.

3.2.3. Prototipo en papel

Un prototipo en papel permite visualizar a un bajo nivel, la aplicación que se desea obtener. A continuación, se presenta los prototipos en papel asociados a las interfaces de usuario del sistema CONEST Desarrollo.

En la Figura 3.5 se puede observar la vista general de la autenticación, la cual es la primera interfaz de usuario del sistema. Está compuesta por el logotipo del sistema CONEST Desarrollo y el formulario de autenticación. Cuenta con algunos enlaces adicionales, tales como la información de contacto del grupo de desarrollo (correo electrónico) y el enlace para la recuperación de la contraseña en caso de olvido.

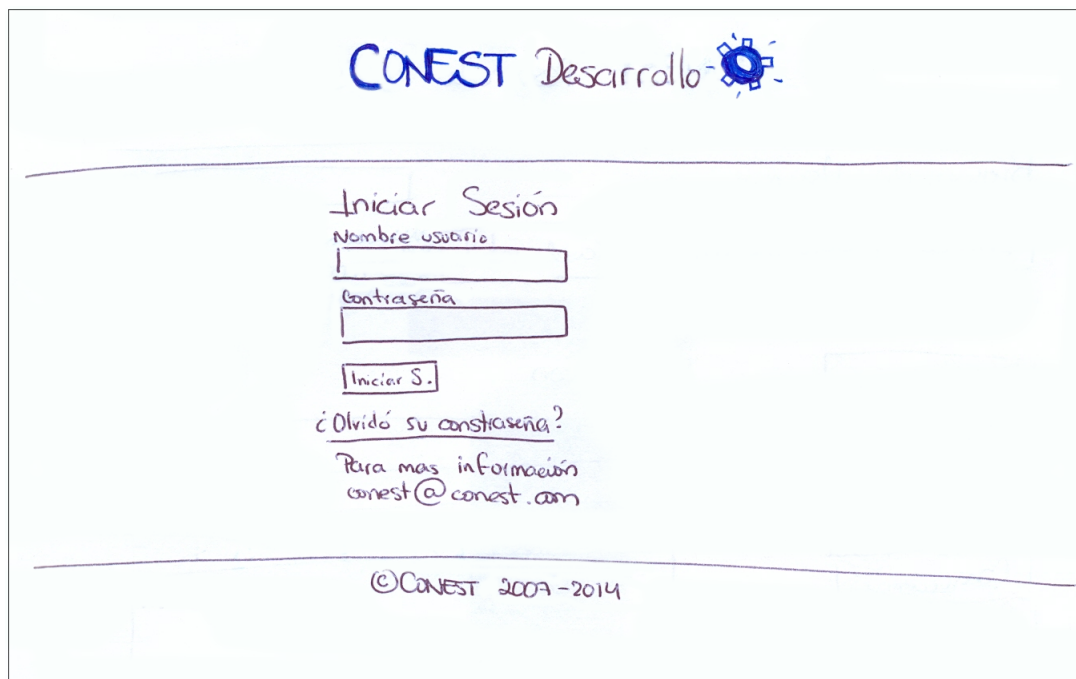


Figura 3.5: Prototipo en papel - Inicio sesión

En la Figura 3.6 se detalla la organización de los elementos en la guía de inicio, la cual cuenta con un menú principal donde se ubican los enlaces a las otras funcionalidades del sistema. Para la descripción de los pasos a seguir durante la instalación de algún programa en específico se provee un menú lateral, el cual carga la información relacionada, tales como versiones, requerimientos del sistema, entre otros.

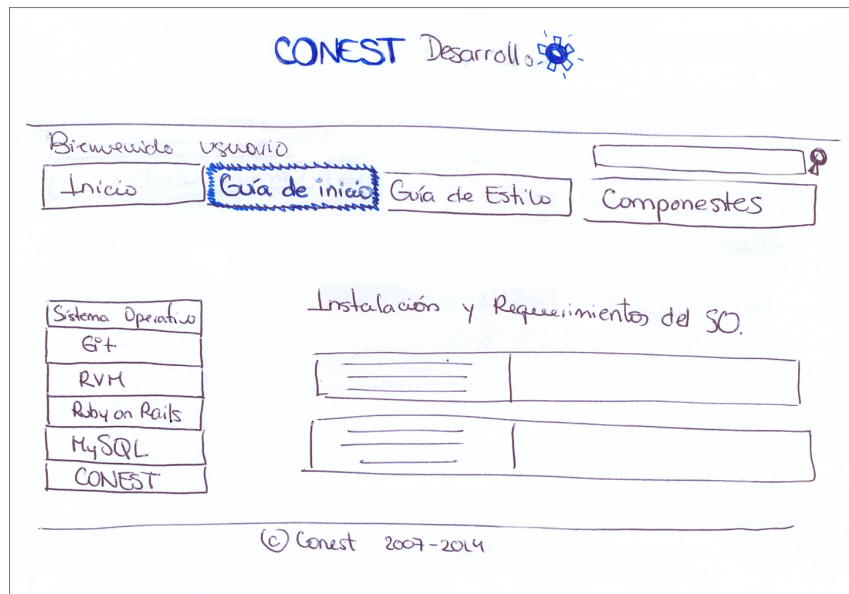


Figura 3.6: Prototipo en papel - Guía de inicio

En la Figura 3.7 puede observarse el prototipo de la interfaz de usuario que permite consultar un componente, el cual a través de un menú lateral, provee la información contenida en un patrón de usuario, donde se encuentran diferentes campos relacionados con los patrones de interacción, tales como problema, solución, contexto, entre otros.

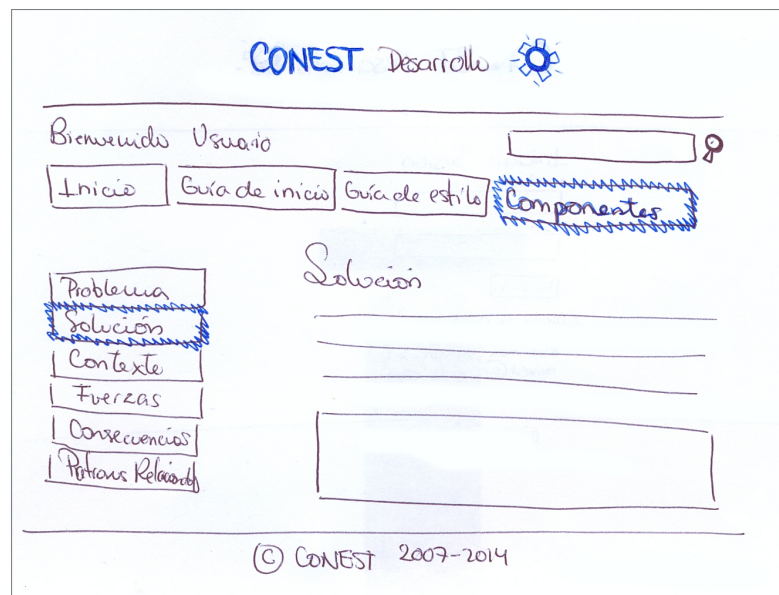


Figura 3.7: Prototipo en papel - Componentes

Por último, en la Figura 3.8, puede observarse el prototipo de interfaz de usuario de

la gestión de usuarios en el sistema. Esta interfaz de usuario está asociada al usuario administrador del sistema y provee las funcionalidades relacionadas a la gestión de usuarios del sistema, tales como agregar usuario, modificar usuario y eliminar usuario.

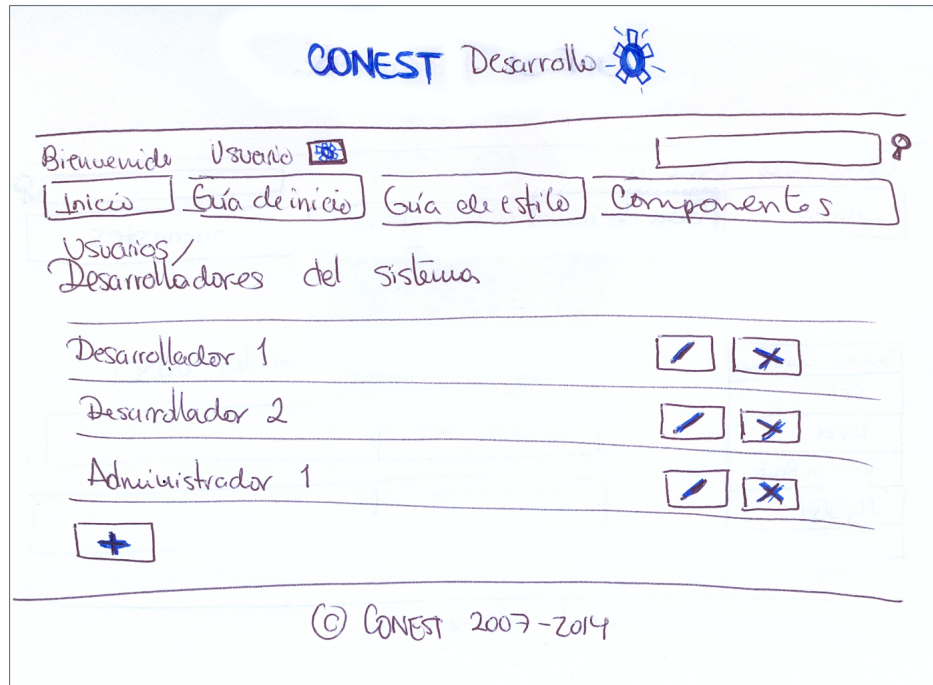




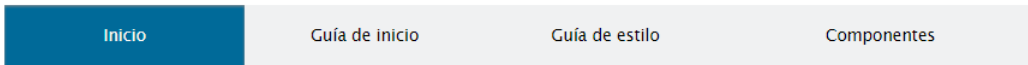
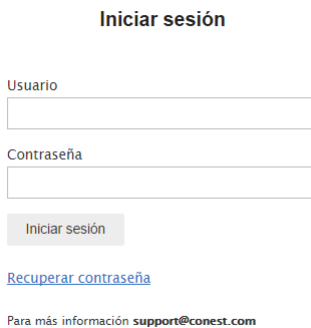
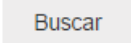
Figura 3.8: Prototipo en papel - Gestión de usuarios

3.2.4. Guía de estilo

A continuación se presenta la guía de estilo propuesta para el sistema CONEST Desarrollo, tomando en cuenta la tormenta de ideas y los requerimientos no funcionales. Tal y como se observa en la Figura 3.8, el esquema de colores se basa en el esquema de colores que compone el logotipo del sistema CONEST, esto con el fin de crear una relación visual entre ambos sistemas.

CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA CONEST DESARROLLO

Tabla 3.8: Guía de estilo de CONEST Desarrollo

Guía de estilo de CONEST Desarrollo
Logo 
Colores Principales  RGB(0,106,158) RGB(255,153,0) RGB(0,51,102) RGB(240,241,242) RGB(224,227,229)
Tipografía Título: Sans-Serif 24px Contenido: Sans-Serif 12px
Navegación Estilo de menú principal:  Estilo de formulario:  Para más información support@conest.com
Botones: 

3.2.5. Patrones de interacción

En la Figura 3.9 se describe el catálogo de patrones del sistema, el cual describe la interfaz de un sistema interactivo, a través de una colección estructurada de patrones que guía al diseñador desde los problemas de gran escala a los problemas de pequeña escala, según Acosta (2011). Este catálogo de patrones de interacción describe las relaciones existentes entre los patrones existentes en el sistema.

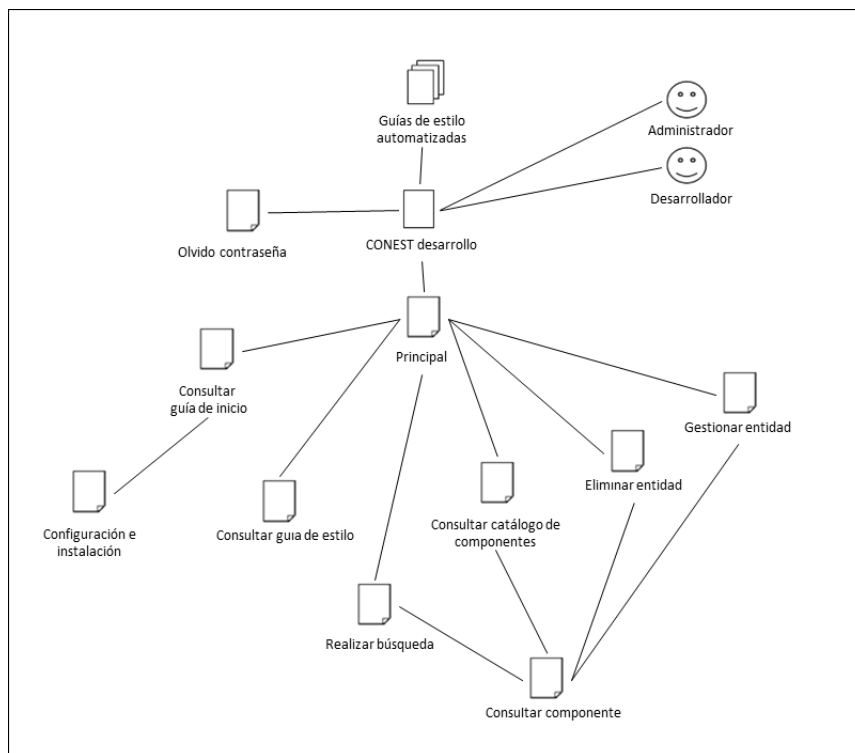



Figura 3.9: *Lenguaje de patrones de interacción CONEST Desarrollo*

A continuación se describen los patrones de interacción del sistema CONEST Desarrollo detallados en la Figura anterior.

Tabla 3.9: Patrón de interacción - Guías de estilo automatizadas

Nombre	Guías de estilo automatizadas 
Problema	El grupo de desarrollo de CONEST requiere agrupar los sistemas que permiten gestionar de manera centralizada el desarrollo de componentes e interfaces de usuario de CONEST
Patrones relacionados	CONEST Desarrollo

CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA CONEST DESARROLLO

Tabla 3.10: Patrón de interacción - Desarrollador


Nombre	Desarrollador 
Descripción	Los desarrolladores de la aplicación CONEST Desarrollo comúnmente son estudiantes de la Licenciatura en Computación, con edades comprendidas entre 20-30 años, de ambos sexos, con conocimientos en el desarrollo de aplicaciones Web.

Tabla 3.11: Patrón de interacción - Administrador




Nombre	Administrador 
Descripción	Los administradores de la aplicación CONEST Desarrollo son personas pertenecientes a la comunidad de la División de Control de Estudios ó de la Escuela de Computación, con edades comprendidas entre los 25-50 años, de ambos sexos, con conocimientos en el manejo de aplicaciones Web. Este usuario debe tener el poder de tomar decisiones en cuanto a la inclusión de nuevos componentes, administradores y desarrolladores en el sistema.

Tabla 3.12: Patrón de interacción - Recordar contraseña

Nombre	Recordar contraseña 
Problema	El usuario olvidó su contraseña y requiere recuperarla
Solución	<p>Crear una interfaz de usuario permita la recuperación de la contraseña del usuario a través de la administración del correo electrónico asociado con la cuenta de usuario del sistema</p> 
Contexto	El usuario se encuentra en la página de autenticación del sistema
Usabilidad	Tolerancia a errores
Consecuencias	Una vez consultada la guía de inicio se puede consultar la configuración e instalación del ambiente y las convenciones y estándares
Patrones relacionados	CONEST Desarrollo


CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA CONEST DESARROLLO

Tabla 3.13: Patrón de interacción - CONEST Desarrollo

Nombre	CONEST Desarrollo □
Problema	El grupo de desarrollo de CONEST requiere un ambiente donde se puedan organizar todas las directrices y buenas prácticas a seguir para desarrollar componentes e interfaces de usuario para el sistema CONEST
Solución	<p>Crear una aplicación Web que centralice la información necesaria para el desarrollo de nuevas interfaces de usuario en CONEST, tales como tecnologías utilizadas en el desarrollo de aplicaciones Web, sistemas manejadores de bases de datos, guías de estilo, catálogos de componentes, entre otros.</p> 
Contexto	El usuario se encuentra en la página de inicio de CONEST Desarrollo y debe autenticarse
Fuerzas	Una aplicación que contenga las directrices y convenciones para desarrollar una aplicación Web permite estandarizar las interfaces de usuario y disminuir el tiempo de desarrollo, incrementando la satisfacción del desarrollador
Usabilidad	Eficiencia, Satisfacción al usuario
Consecuencias	Una vez autenticado en el sistema, el usuario podrá consultar las funcionalidades ofrecidas por la aplicación
Patrones relacionados	Principal, Olvido contraseña

CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA CONEST DESARROLLO

Tabla 3.14: Patrón de interacción - Consultar guía de estilo

Nombre	Consultar guía de estilo □
Problema	El usuario requiere consultar la guía de estilo del sistema CONEST
Solución	<p>Crear una interfaz de usuario que contenga la información de la guía de estilo de CONEST, tal como esquema de colores, tipografía, logotipos, entre otros</p>  <p>The screenshot shows a web interface for 'CONEST Sistema de Gestión Académica'. At the top right, there is a blue header with the CONEST logo and a gear icon. Below this, a navigation bar contains five items: 'Guía de inicio', 'Guía de estilo' (highlighted in blue), 'Catálogo de componentes', 'Descargas', and 'Configuración'. Below the navigation bar, the text 'Bienvenid@ Froiransi Camico' is visible. Underneath, the heading 'Guía de estilo' is followed by the word 'Logo' and a large version of the CONEST logo and name.</p>
Contexto	El usuario se encuentra autenticado en el sistema
Fuerzas	Por medio de la creación de una guía de estilo automatizada se reduce el tiempo de búsqueda y transcripción de la información que se requiere
Usabilidad	Facilidad de uso, Divulgación Progresiva
Patrones relacionados	Principal


CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA CONEST DESARROLLO

Tabla 3.15: Patrón de interacción - Realizar búsqueda

Nombre	Buscar □
Problema	El usuario requiere consultar los componentes que coincidan con un conjunto de palabras proporcionadas por el usuario a través del campo de búsqueda
Solución	<p>Crear un campo de búsqueda en el catálogo de componentes de interfaz de usuario del sistema, que permita consultar los componentes pertenecientes al catálogo del sistema que contengan en la descripción del nombre, problema o solución el conjunto de palabras suministradas por el usuario en el campo de búsqueda</p>  <p>The screenshot shows the CONEST Sistema de Gestión Académica interface. At the top, there is a navigation menu with options: 'Guía de inicio', 'Guía de estilo', 'Catálogo de componentes' (highlighted), 'Descargas', and 'Configuración'. Below the menu, the search results for 'estudiante' are displayed. A search bar contains the text 'estudiante' and a magnifying glass icon. Below the search bar, there is a link for 'Resumen de solicitud' and a paragraph of text: 'El usuario requiere una interfaz que provea un resumen de la solicitud (Trabajo especial de grado), se requiere que la interfaz muestre fecha de solicitud, carrera, sección, período académico, estudiantes, asignatura, título del trabajo especial, tutor firmante, tutor, jurados principales y jurados suplentes.' At the bottom of the page, there is a copyright notice: '© Conest 2007-2014'.</p>
Contexto	El usuario se encuentra autenticado en el sistema
Fuerzas	Por medio de la implementación de un campo de búsqueda global se reduce la cantidad de tareas necesarias para realizar una búsqueda
Usabilidad	Satisfacción al usuario, Eficiencia, Divulgación progresiva,
Consecuencias	Una vez realizada una búsqueda el usuario puede consultar los componentes que cumplan con los requerimientos proporcionados por el usuario
Patrones relacionados	Principal, Consultar componente



CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA CONEST DESARROLLO

Tabla 3.16: Patrón de interacción - Eliminar entidad

Nombre	Eliminar entidad □
Problema	El administrador requiere eliminar una entidad (usuario o componente) del sistema
Solución	<p>Crear una ventana modal que permita confirmar la eliminación de una entidad</p>  <p>The screenshot shows the CONEST Sistema de Gestión Académica interface. A modal dialog box is centered on the screen with the text "¿Está seguro que desea eliminar este usuario?" and two buttons: "Cancelar" and "Aceptar". The background interface includes a header with the CONEST logo and "Sistema de Gestión Académica", a user profile for "Froiransi Camico", and a table titled "Gestionar usuarios" with columns for name, email, and role. The table lists three users: Sergio Rivas, Froiransi Camico, and Jossie Zambrano, all with the role of "Administrador".</p>
Contexto	El usuario se encuentra autenticado como administrador
Fuerzas	Por medio de la creación de una ventana modal se reduce la cantidad de operaciones necesarias para eliminar una entidad
Usabilidad	Tolerancia a errores, Facilidad de uso
Consecuencias	Una vez eliminada una entidad, el usuario volverá a la página principal del sistema
Patrones relacionados	Principal



CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA CONEST DESARROLLO

Tabla 3.17: Patrón de interacción - Consultar guía de inicio

Nombre	Consultar guía de inicio 
Problema	El usuario requiere consultar el proceso de configuración e intalación del ambiente de desarrollo del sistema CONEST, así como los requerimientos de hardware y software del sistema, es decir, el sistema operativo, los <i>frameworks</i> y programas utilizados para la implementación de interfaces de usuario en CONEST
Solución	<p>Crear una interfaz de usuario que contenga las tecnologías necesarias para el ambientes de desarrollo y los pasos a seguir para su instalación. Estas tecnologías incluyen el sistema operativo, el <i>framework</i> del lado del servidor <i>Ruby on Rails</i>, el manejador de base de datos MySQL y el manejador de versiones <i>Git</i></p> 
Contexto	El usuario está autenticado en el sistema
Fuerzas	Creando una lista de tecnologías necesarias para la creación de interfaces se asegura que todos los desarrolladores tengan el mismo ambiente de desarrollo, disminuyendo los conflictos de versiones y actualizaciones de los mismos
Usabilidad	Divulgación progresiva, Facilidad de uso
Patrones relacionados	Principal

CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA CONEST DESARROLLO

Tabla 3.18: Patrón de interacción - Consultar catálogo de componentes

Nombre	Consultar catálogo de componentes 
Problema	El usuario requiere consultar el catálogo de componentes de interfaz de usuario del sistema
Solución	<p>Crear una interfaz de usuario que muestre en forma de lista los componentes pertenecientes al catálogo del sistema</p>  <p>The screenshot shows the CONEST Sistema de Gestión Académica interface. At the top, there is a header with the logo and the text 'CONEST Sistema de Gestión Académica'. Below the header, there is a navigation bar with four items: 'Guía de inicio', 'Guía de estilo', 'Catálogo de componentes' (which is highlighted in blue), and 'Descargas'. To the right of the navigation bar is a search bar with a magnifying glass icon. Below the navigation bar, the main content area is titled 'Catálogo de componentes' and contains a list of links: 'Sistemas de gestión académica', 'CONEST', 'Principal estudiante', and 'Rol de usuario'.</p>
Contexto	El usuario se encuentra autenticado en el sistema
Usabilidad	Eficiencia, Usabilidad, Divulgación progresiva
Consecuencias	Una vez listado el catálogo de componentes de interfaz del sistema, el usuario puede consultar un componente perteneciente a la lista
Patrones relacionados	Principal, Consultar componente

CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA CONEST DESARROLLO

Tabla 3.19: Patrón de interacción - Consultar componente




Nombre	Consultar componente 
Problema	El usuario requiere consultar la información de un componente de interfaz perteneciente al catálogo del sistema
Solución	<p>Proporcionar una interfaz de usuario que contenga de manera organizada la información relacionada con el componente</p>  <p>The screenshot shows the CONEST Sistema de Gestión Académica interface. At the top, there is a navigation menu with options: 'Guía de inicio', 'Guía de estilo', 'Catálogo de componentes' (highlighted), 'Descargas', and 'Configuración'. Below the menu, the text 'Componente: Consultar usuario' is displayed. Underneath, there is a label 'Nombre' and a button labeled 'Consultar usuario' with a magnifying glass icon.</p>
Contexto	El usuario se encuentra autenticado en el sistema
Usabilidad	Eficiencia, Usabilidad, Divulgación progresiva
Patrones relacionados	Principal

Tabla 3.20: Patrón de interacción - Gestionar entidad

Nombre	Gestionar entidad □
Problema	El administrador requiere agregar o modificar una entidad (usuario o componente) al sistema
Solución	<p>Crear una interfaz de usuario que contenga los campos relacionados con una entidad y le permita al administrador gestionarla. Una entidad contempla tanto componentes del catálogo como usuarios del sistema</p>  <p>The screenshot shows the CONEST Sistema de Gestión Académica interface. At the top, there is a header with the logo and the text 'CONEST Sistema de Gestión Académica'. Below the header, there is a navigation bar with links for 'Guía de inicio', 'Guía de estilo', 'Catálogo de componentes', and 'Descargas'. A search bar is also present. The main content area is titled 'Gestionar componentes' and contains two sections: 'Usuarios' and 'Componentes'. The 'Componentes' section lists 'Sistemas de gestión académica' and 'Principal estudiante', each with edit and delete icons. A plus sign is visible at the bottom of the list, indicating an option to add new components. The footer shows '© Conest 2007-2014'.</p>
Contexto	El usuario se encuentra autenticado en el sistema como administrador
Fuerzas	Por medio de la creación de una plantilla general que permita agregar y modificar una entidad, se disminuye el tiempo necesario para gestionar entidades en el sistema
Usabilidad	Facilidad de uso, Tolerante a errores
Consecuencias	Una vez creada o modificada una entidad, los cambios se verán reflejados en el sistema y el usuario será direccionado a la página principal
Patrones relacionados	Principal

3.3. Fase III: Protipaje

La plataforma del sistema CONEST Desarrollo involucró un conjunto variado de tecnologías y herramientas de Software Libre, tal como se observa en la Figura 3.10. Este sistema se desarrolló sobre la plataforma del Sistema Operativo *Linux Mint*, el cual es un sistema operativo basado en UNIX, en el que se encuentra instalado el *framework* para el desarrollo de aplicaciones Web del lado del servidor *Ruby on Rails* y el Sistema Manejador de Bases de Datos MySQL. Adicionalmente para definir la apariencia de las

IU del sistema se utilizó el *framework* CSS Kube y se incorporaron algunos módulos del *framework* CSS Bootstrap.



Figura 3.10: *Ambiente de desarrollo*






Durante las dos iteraciones llevadas a cabo durante esta fase, se implementaron diferentes prototipos ejecutables relacionados con las diferentes funcionalidades que provee el sistema CONEST Desarrollo. A continuación se describen las IU desarrolladas relacionadas con los diferentes prototipos ejecutables.

3.3.1. Iteración 1: Funcionalidades de gestión

En la primera iteración se desarrolló la autenticación del sistema, la cual se llevó a cabo de manera diferente a la implementada en el sistema CONEST, debido a la cantidad de usuarios a la que está dirigida este sistema. Esta autenticación requiere la administración de un nombre de usuario designado por un administrador del sistema.

La implementación de las diferentes funcionalidades dentro de este sistema se llevó a cabo a través de la utilización de diferentes metáforas dentro del contexto de la aplicación, con el fin de promover la simplicidad de las IU. A continuación, en la Tabla 3.21 se presentan los diferentes iconos utilizados dentro de la aplicación, junto con una breve descripción de la acción que se realiza a través de esta metáfora.

Tabla 3.21: Metáforas de la aplicación CONEST Desarrollo

Figura	Descripción
	Este icono en forma de lupa permite realizar una búsqueda dentro del sistema, a través del buscador general ubicado en todas las IU del sistema
	Este icono en forma de + permite al usuario, incorporar un nuevo usuario o componente al sistema
	Este icono en forma de lápiz permite modificar a través de una ventana modal los datos relacionados con un usuario o componente del sistema
	Este icono en forma de papelera permite eliminar un usuario o componente existente en el sistema
	Este icono permite iniciar la descarga de un archivo o documento

Posteriormente se desarrollaron las IU relacionadas con la gestión de usuarios y componentes del sistema, a través del enlace ubicado en el menú principal y la IU para la gestión de componentes incorpora las funcionalidades de agregar, modificar y eliminar una entidad.

Las pruebas heurísticas se realizan con el fin de encontrar problemas de usabilidad en la IU de un sistema (Nielsen, 1995). Para ello, se utilizan las heurísticas creadas por Jakob Nielsen, las cuales se describen a continuación.

Las heurísticas que se tomaron en cuenta para evaluación son las siguientes:

- H1. Visibilidad de sistema: el sistema debe informar a los usuarios del estado del sistema, dando una retroalimentación apropiada en un tiempo razonable.
- H2. Utilizar el lenguaje de los usuarios: el sistema debe utilizar el lenguaje de los usuarios, con palabras o frases que le sean conocidas, en lugar de los términos que se utilizan en el sistema, para que no resulte difícil utilizar el sistema.
- H3. Control y libertad para el usuario: en caso que los usuarios elijan una opción del sistema por error, éste debe contar con las opciones de deshacer y rehacer para proveer al usuario de una salida fácil sin tener que utilizar un diálogo extendido.
- H4. Consistencia y estándares: el usuario debe seguir las normas y convenciones de la plataforma sobre la que está implementando el sistema, para que no se tenga que preguntar el significado de las palabras, situaciones o acciones del sistema.

CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA CONEST DESARROLLO

- H5. Prevención de errores: es más importante prevenir la aparición de errores que de generar buenos mensajes de error.
- H6. Minimizar la carga de la memoria del usuario: en la interfaz se deben mantener los objetos, acciones y opciones visibles para que el usuario no pierda el tiempo recordando la información de una parte del diálogo a otra.
- H7. Flexibilidad y eficiencia de uso: las instrucciones para el uso del sistema tienen que ser visibles o fácilmente accesibles cada vez que se necesiten, para que al usuario inexperto se le facilite aprender a utilizar el sistema. Es importante que el sistema permita personalizar acciones frecuentes, para acelerar el uso del sistema.
- H8. Los diálogos estéticos y diseño minimalista: la interfaz no debe contener información que no sea relevante o se utilice raramente, pues cada unidad adicional de información en un diálogo compite con las unidades relevantes de la información y disminuye su visibilidad relativa.
- H9. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores: los mensajes de error deben expresarse en un lenguaje claro y deben indicar exactamente el problema.
- H10. Ayuda y documentación: en caso que el sistema necesite disponer de ayuda y documentación, ésta tiene que ser fácil de encontrar, centrada en las tareas del usuario, tener información de las etapas a realizar y que no sea demasiado extensa.

La escala de evaluación va de 0 a 4 con los siguientes valores: (0) No es un problema de Usabilidad, (1) Problema cosmético, (2) Problema de poca importancia, (3) Problema grave y (4) Catástrofe es obligatorio arreglarlo.

En la Tabla 3.22 se presentan los resultados de la evaluación heurística aplicada durante esta iteración.

Tabla 3.22: Resultados de la evaluación Heurística

Problema	Heurística	Valoración	Solución
La bienvenida del usuario se realiza a través del nombre de usuario del sistema	H2	1	Colocar el nombre del usuario en lugar del usuario del sistema
Los mensajes del sistema en el inicio de sesión no se encuentran alineados con el formulario	H4,H8	2	Alinear los mensajes con el contenido del formulario de inicio de sesión
La funcionalidad de configuración del sistema no es reconocible a simple vista	H4,H6	3	Modificar el icono de configuración del sistema
Los enlaces para regresar al estado anterior no son reconocibles	H6	0	Modificar el enlace de regresar al estado anterior por un icono relacionado con esta acción
Las secciones de la guía de inicio no poseen numeración	H6	1	Numerar los elementos que se encuentran en la guía de inicio
No puede apreciarse el formato del lenguaje del código relacionado a la solución de los patrones de interacción enriquecidos	H4,H5	1	Modificar las hojas de estilo del sistema
No es posible indicar el tipo de patrón de interacción	H6	2	Incluir un campo dentro del formulario de los patrones donde se indique el tipo de patrón
En el formulario para agregar un patrón nuevo no existe documentación o ejemplos para los campos del formulario	H10	2	Incorporar ejemplos en los campos del formulario de los patrones
El catálogo de patrones de interacción no incluye una descripción de los mismos	H6	1	Agregar una breve descripción de cada patrón en el catálogo de componentes

Los problemas encontrados durante la aplicación de esta evaluación heurística se solventaron oportunamente, permitiendo al grupo de desarrollo evaluar las sugerencias del especialista y realizar los ajustes necesarios que permiten obtener un producto de software que se ajuste a las necesidades del usuario, obteniendo un software refinado y con altos estándares de calidad.

3.3.2. Iteración 2: Funcionalidades de consulta

Durante la segunda iteración de la fase de protipaje se desarrollaron las IU relacionadas con la consulta de los componentes, sección de descargas y las guías de inicio y estilo del sistema CONEST.

La estructura de la guía de inicio, está compuesta por la descripción de los requerimientos necesarios para la instalación del ambiente de desarrollo del sistema CONEST. Esta descripción incluye los pasos a seguir para la instalación del sistema operativo, el manejador de versiones *Git*, la plataforma para el manejo de las versiones del lenguaje *Ruby* denominada *RVM*, el *framework* del lado del servidor *Ruby on Rails*, el Sistema Manejador de Base de Datos *MySQL* y por último el manejo del sistema CONEST.

De igual forma, durante esta iteración se desarrolló un sistema de descarga de archivos importantes dentro del ambiente de desarrollo del sistema CONEST, el cual incorpora un documento con los estándares de programación, el diagrama relacional de la base de datos y el lenguaje de patrones del sistema.

3.4. Fase IV: Entrega

En esta fase se aplican diferentes pruebas al sistema, con el fin de certificar que el sistema desarrollado cumple satisfactoriamente con los requerimientos funcionales y no funcionales, además verifica que posee los niveles de usabilidad presentes en un software de calidad. Adicionalmente, las pruebas de aceptación verifican el grado de satisfacción de los usuarios finales en relación a los requerimientos funcionales de un sistema.

Es importante mencionar que luego de implementar los módulos del sistema CONEST Desarrollo y realizar las pruebas correspondientes, el sistema se integró con el servidor de producción y se habilitó para el uso del grupo de desarrollo del sistema CONEST, permitiendo de esta manera la incorporación de este sistema como herramienta de apoyo para los desarrolladores pertenecientes a este grupo.

3.4.1. Pruebas de aceptación

Para llevar a cabo la verificación del software, se realizó una prueba de aceptación de todas las funcionalidades que ofrece el sistema en su versión de producción, con el objetivo de detectar posibles fallos e inconsistencias. Se aplicó esta prueba a través de una encuesta con las siguientes cuatro preguntas:

CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA CONEST DESARROLLO

1. ¿El sistema cumple con tus expectativas como desarrollador?
2. ¿Crees que es intuitivo?
3. ¿La apariencia del sistema te resulta agradable?
4. ¿Crees que es de utilidad la información y las funcionalidades que provee este sistema?

El grupo de muestra al que se le aplicó la encuesta se encuentra formado por 5 miembros del grupo de desarrollo del sistema CONEST con diferentes roles dentro de este grupo de trabajo, los cuales llevaron a cabo esta encuesta de manera individual, con una escala de evaluación sobre su apreciación con valores entre 1 y 5, siendo 1 el nivel de apreciación más bajo y 5 el nivel de apreciación más alto.

A continuación, en la Figura 3.11 se pueden observar los resultados de la aplicación de esta prueba.

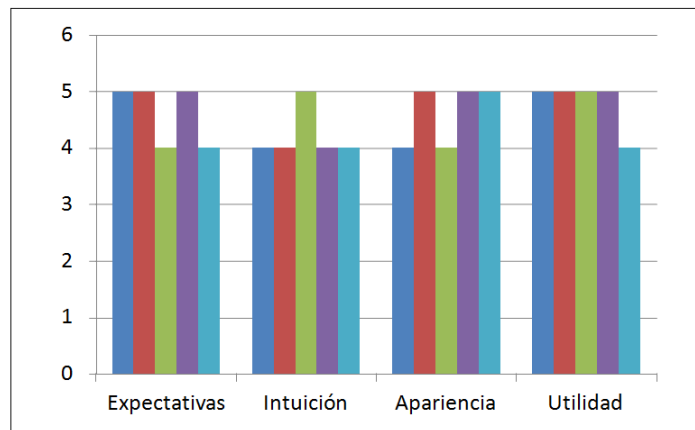


Figura 3.11: Prueba de aceptación CONEST Desarrollo

Como resultado de la aplicación de esta prueba de aceptación es posible concluir que en general el sistema CONEST Desarrollo posee un alto nivel de aceptación por parte de la comunidad de desarrolladores del sistema CONEST, con más del 90% de aceptación, cumpliendo con las necesidades y requerimientos del usuario, a través de un sistema usable y con altos estándares de calidad.

Por medio de la aplicación de la metodología de desarrollo AgilUs fue posible llevar a cabo exitosamente el proceso de desarrollo del sistema CONEST Desarrollo, aplicando durante cada una de las fases, diferentes técnicas y elaborando artefactos para la consecución de la usabilidad.

Conclusiones

El sistema de gestión académica CONEST a través del tiempo se ha convertido en una herramienta de vital importancia para el funcionamiento de las actividades llevadas a cabo por la División de Control de Estudios de la Facultad de Ciencias. En vista de la gran cantidad de usuarios que utilizan este sistema, resulta crucial que la interacción entre el usuario y el sistema genere satisfacción, al gestionar de manera efectiva los requerimientos solicitados, además de tener una apariencia consistente, coherente y organizada en cada una de sus interfaces de usuario.

Al inicio del desarrollo de este Trabajo Especial de Grado se planteó como objetivo general “Automatizar las guías de diseño del ambiente de desarrollo del sistema CONEST”, el cual se logró satisfactoriamente, permitiendo articular las decisiones de diseño del grupo de desarrollo del sistema CONEST, facilitando información útil a los programadores nuevos que se incorporan a este grupo. Como consecuencia surge la creación del sistema CONEST Desarrollo, el cual además de incorporar las guías de diseño del sistema CONEST, ofrece un ambiente de desarrollo centralizado que incorpora todas las herramientas necesarias para el desarrollo de funcionalidades dentro de este sistema.

La relación de los resultados obtenidos durante este trabajo con respecto a los objetivos específicos se describen a continuación.

- El estudio de las herramientas que proveen componentes de interfaz de usuario se llevó a cabo durante la fase de requerimientos del sistema CONEST Desarrollo. Este análisis permitió concretar el conjunto de herramientas incluidas dentro de la implementación de este sistema, tales como las guías de estilo e inicio, el catálogo de componentes basado en patrones de interacción y la sección de descargas. Las cuales incorporan un conjunto de buenas prácticas presentes en la mayoría de los sistemas existentes, permitiendo conocer a fondo las tendencias actuales para el desarrollo de herramientas y guías de diseño automatizadas de diferentes organizaciones, tales como *Android*, *iOS*, *Google*, *Bootstrap*, *Foundation* y *Kube*. Las cuales permiten facilitar el trabajo de los desarrolladores dentro de los ambientes de trabajo e incorporando algunas recomendaciones incluidas al final

de este trabajo, con el fin de ampliar las funcionalidades del sistema CONEST Desarrollo.

- El desarrollo del catálogo de patrones de interacción del sistema CONEST por medio de la aplicación de la metodología AgilUs permitió desarrollar de manera exitosa la solución a la problemática existente, debido a que esta metodología aplica evaluaciones de usabilidad desde etapas tempranas del desarrollo del software, obteniendo un software que genera satisfacción en el usuario y cumpliendo con el objetivo principal de este tipo de software orientado a desarrolladores en un ambiente académico. Adicionalmente, la creación del catálogo de componentes del sistema contribuyó a establecer los estándares en la creación de los componentes de interfaz de usuario, debido a que los patrones de interacción describen soluciones exitosas a problemas recurrentes, describiendo de esta manera los lineamientos a seguir en el desarrollo de funcionalidades similares a las existentes.
- El análisis de las interfaces de usuario del sistema CONEST fue una actividad bastante difícil de llevar a cabo, en vista del desarrollo constante de nuevas funcionalidades y los cambios en la estructura y funcionamiento de las interfaces de usuario que se estudiaron. Sin embargo, este análisis contribuyó en varias ocasiones a detectar errores en la implementación de funcionalidades, así como también a detectar inconsistencias y errores en las interfaces de usuario del sistema.
- La implementación del sistema CONEST Desarrollo, a través de la utilización de los *frameworks* CSS *Kube* y *Bootstrap*, y el *framework* del lado del servidor *Ruby on Rails*, resultó un tanto compleja, debido a que la arquitectura de *Kube* y *Bootstrap* fue difícil de integrar con *Ruby on Rails*. Sin embargo, luego de realizar la integración de estos *frameworks* resultó fácil y rápida la implementación de las funcionalidades del sistema.

Los módulos implementados para este sistema componen las guías de diseño del ambiente de desarrollo del sistema CONEST, las cuales incorporan estándares que deben cumplirse para el desarrollo funcionalidades e interfaces de usuario, disminuyendo las inconsistencias y permitiendo la reutilización de los diferentes componentes de interfaz de usuario como una solución exitosa a problemas similares en contextos específicos. Estas guías de diseño permiten apoyar el proceso de integración de los nuevos desarrolladores de esta plataforma, sentando las bases sobre las cuales estos desarrolladores podrán basar el desarrollo de sus actividades y funcionalidades dentro del sistema.

La descripción de los componentes de interfaz de usuario mediante los patrones de interacción, permitió generar de manera sistemática y sencilla el catálogo de componentes de interfaz del sistema CONEST, asimismo permitió generar el insumo requerido para la implementación del sistema CONEST Desarrollo.

La utilización de la metodología AgilUs permitió implementar de manera exitosa la solución a la problemática presentada, ya que gracias a que esta metodología se centra en potenciar la usabilidad, propone diferentes artefactos y herramientas que permiten crear software usable y con altos estándares de calidad.

La interacción con el sistema CONEST en su versión más reciente, resultó ser un desafío para el desarrollo de esta investigación, ya que es un sistema grande y complejo, el cual fue implementado, en su mayoría, en paralelo con esta investigación, por lo que en más de una ocasión los componentes que se estudiaron durante este trabajo cambiaron su funcionamiento y la forma de interacción entre los componentes, retrasando la investigación y obligando a cambiar constantemente la información relacionada con cada uno de estos. Sin embargo, durante el análisis y clasificación de los componentes de interfaz se realizaron aportes significativos en cuanto a reportes de fallas e inconsistencias en el sistema, mejorando la calidad del producto final de software del sistema CONEST y favoreciendo la importancia de tener un repositorio de componentes y estándares.

De esta manera se puede decir que los objetivos propuestos para este Trabajo Especial de Grado se cumplieron satisfactoriamente, generando un aporte significativo en el contexto de las guías de diseño y sentando las bases para continuar el proyecto CONEST Desarrollo, el cual será de gran utilidad para toda la comunidad de desarrolladores y profesionales pertenecientes al grupo de desarrollo del sistema CONEST.

Recomendaciones y trabajos futuros

Algunas de las recomendaciones que se proponen para incorporar funcionalidades generales del sistema CONEST Desarrollo son:

- Realizar un estudio de los componentes de código pertenecientes al sistema CONEST, con el fin de estandarizar la creación de módulos y funcionalidades de componentes *back-end*, incorporando estos componentes al sistema CONEST Desarrollo a través de un catálogo de componentes de código.
- Crear un sistema de gestión y seguimiento de actividades asignadas, con el fin de promover el cumplimiento de las asignaciones en el tiempo requerido.
- Incorporar una guía que contenga los estándares del formato de los archivos de salida y divulgación de información del sistema. Tales como correos electrónicos, archivos PDF, constancias, entre otros.

Referencias Bibliográficas

- Acosta, A. (2004). Métodos y herramientas para la construcción de la interfaz de usuario utilizando patrones de interacción. Facultad de Ciencias, Universidad Central de Venezuela: Tesis doctoral. [Consulta: 2014, Febrero 16]
- Acosta, A. (2011). AgilUs: un método ágil de desarrollo de software que incorpora la usabilidad. [Documento en línea]. Disponible: <http://www.ciens.ucv.ve/escueladecomputacion/documentos/archivo/121> [Consulta: Febrero 3]
- Acosta, A. (2013). AgilUs: una metodología ágil de desarrollo de software que incorpora la evaluación de la usabilidad. Facultad de Ciencias, Universidad Central de Venezuela. Trabajo de ascenso para ascender a la categoría Titular. [Consulta: 2014, Septiembre 9]
- Cataffi, R. Zambrano, N. (2007). Aspectos técnicos y sociales de la interacción humano-computador en el ámbito de la computación colaborativa. [Documento en línea]. Disponible: <http://www.ciens.ucv.ve/escueladecomputacion/documentos/archivo/42> [Consulta: 2014, Febrero 21]
- Baronetti, M. (2014). Foundation 5. [Página Web en línea]. Disponible: <http://manuelbaronetti.com.ar/blog/2013/11/zurb-landa-foundation-5/> [Consulta: 2014, Agosto 13]
- Ceria, S. (2014). Casos de Uso: Un método práctico para explorar requerimientos. [Documento en línea]. Disponible: http://www-2.dc.uba.ar/materias/isoft1/2001_2/apuntes/CasosDeUso.pdf [Consulta: 2014, Septiembre 10]
- Chong, R., Danubianu, M., Iyer, A., Nandan, C., Nonvinkere, M., Perniu, L., Sharma, N. (2010). Database Fundamentals. [Documento en línea]. Disponible: http://public.dhe.ibm.com/software/dw/db2/express-c/wiki/Database_fundamentals.pdf [Consulta: 2014, Marzo 2]
- Díaz, M. (2014). Metapatrón. [Página Web en línea]. Disponible: <http://www.moisesdaniel.com/es/wri/metapatrones.pdf> [Consulta: 2014, Febrero 19]

Referencias Bibliográficas

- Eguaras, M. (2014). Manual de estilo: ¿Por qué estandarizar y armonizar criterios?. [Página Web en línea]. Disponible: <http://marianaeguaras.com/manual-de-estilo-por-que-estandarizar-y-armonizar-criterios> [Consultado: 2014, Agosto 18]
- Eguiluz, J. (2014). AJAX. [Documento en línea]. Disponible: <http://librosweb.es/ajax/> [Consulta: 2014, Enero 24]
- Ekas, L. (2012). 5 beneficios de las metodologías ágiles en el desarrollo de sistemas de software. [Página Web en línea]. Disponible: https://www.ibm.com/developerworks/community/blogs/rationalspain/entry/5_beneficios_de_las_metodolog_C3_ADas__C3_A1giles_en_el_desarrollo_de_sistemas_de_software5?lang=en [Consulta: 2014, Septiembre 9]
- Foundation (2014). History of Foundation. [Página Web en línea]. Disponible:<http://foundation.zurb.com/learn/about.html> [Consulta: 2014, Enero 27]
- Garcia, B. (2014). Ventajas y desventajas de Bootstrap. [Página Web en línea]. Disponible:<http://bruno.garciaechegaray.com/Bootstrap.The.Melee/presentation/#ventajas-theme-02> [Consulta: 2014, Agosto 14]
- Imperavi, (2014). Kube Professional CSS-framework. [Página Web en línea]. Disponible: <http://imperavi.com/kube/> [Consulta: 2014, Agosto 11]
- Jacobson, I. (2005). Casos de uso:. [Documento en línea]. Disponible: http://www.ivarjacobson.com/uploadedFiles/Pages/Knowledge_Centre/Resources/White_Paper/Resources/Use%20Case%202.0%20-%20Spanish%20Translation.pdf [Consulta: 2014, Septiembre 10]
- Luján, S. (2002). Programación de aplicaciones Web: historia, principios básicos y clientes Web. Editorial Club Universitario. Disponible: <http://gplsi.dlsi.ua.es/almacenes/ver.php?pdf=42> [Consulta: 2014, Enero 9]
- Mac Developer Library (2013). Framework Programming Guide. [Página Web en línea]. Disponible: <https://developer.apple.com/library/mac/documentation/macosx/Conceptual/BPFrameworks/Concepts/WhatAreFrameworks.html> [Consulta: 2014, Enero 25]
- Mejía, A. (2011). Ruby on Rails Architectural Design. [Página Web en línea]. Disponible: <http://adrianmejia.com/blog/2011/08/11/ruby-on-rails-architectural-design/> [Consulta: 2014, Enero 27]
- Microsoft Developer Network (2014). Diagramas de Casos de uso UML: Referencia. [Página Web en línea]. Disponible: <http://msdn.microsoft.com/es-es/library/dd409427.aspx> [Consulta: 2014, Agosto 20]

Referencias Bibliográficas

- Moreno, L. (2010). Perfiles de usuario. [Página Web en línea]. Disponible: <http://labda.inf.uc3m.es/awa/es/node/66> [Consulta: 2014, Septiembre 10]
- Mozilla Developer Nerwork-a (2014). AJAX. [Página Web en línea]. Disponible: <https://developer.mozilla.org/en-US/docs/AJAX> [Consulta: 2014, Enero 10]
- Mozilla Developer Nerwork-b (2014). CSS. [Página Web en línea]. Disponible: <https://developer.mozilla.org/en-US/docs/Web/CSS> [Consulta: 2014, Enero 8]
- Mozilla Developer Nerwork-c (2014). DOM. [Página Web en línea]. Disponible: https://developer.mozilla.org/en-US/docs/DOM/DOM_Reference/Introduction [Consulta: 2014, Enero 9]
- Mozilla Developer Nerwork-d (2014). JavaScript. [Página Web en línea]. Disponible: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> [Consulta: 2014, Enero 8]
- Nielsen, J. (1995). 10 Usability Heuristics for User Interface Design .[Página Web en línea]. Disponible: <http://www.nngroup.com/articles/ten-usability-heuristics/> [Consulta: 2014, Septiembre 1]
- Shklar, L, Rosen, R. (2009). Web Application Architecture Principles, protocols and practices. Editorial John Wiley & Sons, Ltd. [Documento en línea]. Disponible: <http://www.ce.uniroma2.it/courses/PRSI/WebApplication.pdf> [Consulta: 2014, Enero 8]
- Sommerville, I. (2005). Ingeniería de Software, Séptima Edición. Editorial Addison-Wesley Publishing Company. [Consulta: 2014, Enero 28]
- Universidad Central de Venezuela (2009). Reseña Histórica . [Página Web en línea]. Disponible: <http://www.ucv.ve/sobre-la-ucv/resena-historica.html> [Consulta: 2014, Febrero 26]
- Universidad Politécnica de Valencia, (2014). Manual de estilos. [Página Web en línea]. Disponible: http://www.upv.es/entidades/ASIC/manuales/guia_estilos_upv.pdf [Consulta: 2014, Agosto 18]
- Universidad Politécnica de Valencia, (2014). Proceso de desarrollo de software. [Página Web en línea]. Disponible: <http://ldc.usb.ve/~abianc/materias/ci4712/ProcesoSW-Letelier.pdf> [Consulta: 2014, Septiembre 9]
- Villa, L. (2014). Prototipo en papel. [Página Web en línea]. Disponible: http://www.webtaller.com/maletin/articulos/prototipos_de_papel.php [Consulta: 2014, Agosto 17]
- W3C-a (2013). CSS. [Página Web en línea]. Disponible: http://www.w3.org/community/webed/wiki/CSS/Training/What_is_CSS [Consulta: 2014, Enero 10]

Referencias Bibliográficas

- W3C-b (2013). HTML. [Página Web en línea]. Disponible: http://www.w3.org/community/webed/wiki/The_basics_of_HTML [Consulta: 2014, Enero 8]
- W3C-c (2013). Script. [Página Web en línea]. Disponible: <http://www.w3.org/standards/webdesign/script> [Consulta: 2014, Enero 24]
- Wikipedia-a (2013). Convención sobre Configuración. [Página Web en línea]. Disponible: http://es.wikipedia.org/wiki/Convenci%C3%B3n_sobre_Configuraci%C3%B3n [Consulta: 2014, Febrero 10]
- Wikipedia-b (2014). Heurísticas de Nielsen. [Página Web en línea]. Disponible: http://es.wikipedia.org/wiki/Heur%C3%ADsticas_de_Nielsen [Consulta: 2014, Septiembre 2]
- Wikipedia-c, (2014). Requerimientos de sistemas. [Página Web en línea]. Disponible: [http://es.wikipedia.org/wiki/Requerimiento_\(sistemas\)](http://es.wikipedia.org/wiki/Requerimiento_(sistemas)) [Consulta: 2014, Agosto 15]
- Wikipedia-d (2013). No te repitas. [Página Web en línea]. Disponible: http://es.wikipedia.org/wiki/No_te_repitas [Consulta: 2014, Febrero 10]
- Wikipedia-e (2008). Ruby on Rails. [Página Web en línea]. Disponible: http://es.wikipedia.org/wiki/Ruby_on_Rails [Consulta: 2014, Enero 27]
- Wikipedia-f (2013). SASS. [Página Web en línea]. Disponible: [http://es.wikipedia.org/wiki/Sass_\(lenguaje_de_hojas_de_estilo\)](http://es.wikipedia.org/wiki/Sass_(lenguaje_de_hojas_de_estilo)) [Consulta: 2014, Febrero 12]
- Wikipedia-g (2014). Tormenta de ideas. [Página web en línea]. Disponible: http://es.wikipedia.org/wiki/Lluvia_de_ideas [Consulta: 2014, Agosto 12]