



**UNIVERSIDAD DE
SAN BUENAVENTURA
SECCIONAL MEDELLÍN**
Calidad Humana y Profesional

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN
CON INGENIERIA WEB PARA LA ADMINISTRACIÓN DE PROYECTOS
DE INVESTIGACIÓN FACULTAD INGENIERIA DE LA UNIVERSIDAD
SAN BUENAVENTURA MEDELLIN

HAROLD ENOC FLOREZ PEREZ

DORIAN NORBERTO ARDILA PUENTES

UNIVERSIDAD DE SAN BUENAVENTURA SECCIONAL MEDELLÍN

FACULTAD DE INGENIERÍAS

INGENIERÍA DE SISTEMAS

MEDELLIN

2014



**UNIVERSIDAD DE
SAN BUENAVENTURA
SECCIONAL MEDELLÍN**
Calidad Humana y Profesional

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN
CON INGENIERIA WEB PARA LA ADMINISTRACIÓN DE PROYECTOS
DE INVESTIGACIÓN FACULTAD INGENIERIA DE LA UNIVERSIDAD
SAN BUENAVENTURA MEDELLIN**

HAROLD ENOC FLOREZ PEREZ

DORIAN NORBERTO ARDILA PUENTES

Proyecto presentado para optar al título de Ingeniero de Sistemas

Asesor

Carlos Arturo Castro Castro

Jefe de Línea de Investigación en Ingeniería del Software y
Telecomunicaciones

UNIVERSIDAD DE SAN BUENAVENTURA SECCIONAL MEDELLÍN

FACULTAD DE INGENIERÍAS

INGENIERÍA DE SISTEMAS

MEDELLIN

2014

Nota de Aceptación:

Aprobado por el comité de grado en cumplimiento de los requisitos exigidos por la Universidad de San Buenaventura para optar al título de Ingenieros de Sistemas.

Firma Asesor

Firma Calificador

CONTENIDO

	Pág.
1. INTRODUCCION	7
2. JUSTIFICACION	8
3. PLANTEAMIENTO DEL PROBLEMA	9
4. OBJETIVO GENERAL	10
4.1 OBJETIVOS ESPECÍFICOS	10
5. MARCO REFERENCIAL	11
5.1 ESTADO DE ARTE	11
5.2 MARCO CONCEPTUAL	12
5.2.1 Modelamiento ágil	12
5.2.2 Testing ágil	12
5.2.3 Técnicas Ágiles en el desarrollo de Software	13
5.2.4 Ingeniería de Software	13
5.2.5 Figura 1. Modelo Metodología Ágil	14
6. FASES METODOLÓGICAS	15
7. CRONOGRAMA	17
8. APLICACIÓN DEL MODELO INGENIERIA DE SOFTWARE BASADO EN LA METODOLOGÍA SCRUM	18
8.1 Definición de Roles	19
8.2 Personas y roles del proyecto Snies	20
8.3 Sprint	20
8.4 Cuadro de reuniones	21
8.5 Cuadro Reuniones Técnicas	22
8.6 Pila de producto (Sprint Backlog)	22
8.7 Requisitos No Funcionales	26
8.8 Muestras de hojas de Excel enviadas	27
8.9 LISTADO DE ACTIVIDADES	28
8.10 DETALLE DE LA SOLUCIÓN	32
8.10.1 Diseño y Arquitectura de la solución	32
8.10.2 Arquitectura N-Capas	32
8.10.2.1 Capa de Acceso a Datos	32
8.10.2.2 Capa de Lógica de Negocio	32
8.10.2.3 Capa de Presentación	33

8.10.3 Diagrama de la Arquitectura	33
8.11 CASOS DE USO	34
8.12 Requisitos Funcionales Acceso al Sistema	34
8.12.1.1 Casos de Uso	34
8.12.2 Administrar de Permisos	36
8.12.3 Administración de Maestros	37
8.12.3.1 Requisitos Funcionales Administración Maestros	37
8.12.3.2 Casos de Uso Administrar maestros	38
8.12.4 Generación de Reporte	41
8.12.4.1 Requisitos Funcionales Generación de Reporte	41
8.12.4.2 Casos de Uso	41
8.13 PROTOTIPOS DESARROLLADOS	43
8.14 Pruebas Funcionales	47
8.15 Manual de usuario y manual de instalación	47
9. REFERENCIAS BIBLIOGRAFICAS	48
10. GLOSARIO	49
11. CONCLUSIONES	51

FIGURAS

	Pág.
1. Figura 1. Modelo Metodología Ágil	14
2. Figura 2. Cronograma proyecto Snies	17
3. Figura 3. Historias de Usuario	20
4. Figura 4. Sprint 1	23
5. Figura 5. Sprint 2	23
6. Figura 6. Sprint 3	24
7. Figura 7. Sprint 4	24
8. Figura 8. Sprint 5	24
9. Figura 9. Sprint 6	25
10. Figura 10. Sprint 7	25
11. Figura 11. Sprint 8	25
12. Figura 12. Sprint 9	26
13. Figura 13. Sprint 10	26
14. Figura 14. Hoja Excel Proyectos	27
15. Figura 15. Hoja Excel Flujo de caja	27
16. Figura 16. Historia de Usuario Aprobado	29
17. Figura 17. Historia de Usuario Aprobado	30
18. Figura 18. Historia de Usuario Aprobado	30
19. Figura 19. Historia de Usuario Aprobado	31
20. Figura 20. Diagrama de la Arquitectura	33
21. Figura 21. Módulo de Acceso	43
22. Figura 22. Interfaz Página Principal	44
23. Figura 23. IES	44
24. Figura 24. Agregar IES	45
25. Figura 25. Validación IES	45
26. Figura 26. Listado de las IES Registradas	45
27. Figura 27. Modificar Listado de IES Registradas	46
28. Figura 28. Eliminación Listado IES Registradas	47

1. INTRODUCCION

Las metodologías ágiles para el desarrollo de aplicaciones Web, se basan en una serie de procesos donde su principal objetivo es la priorización de los requisitos expresados por el cliente, posteriormente se realizan entregas de acuerdo al orden de priorización pactado. Se debe desplegar una serie de actividades asegurando los entregables enfocados a la fase correspondiente dentro del ciclo de vida de los proyectos; la primera es la conceptualización donde se realiza el levantamiento de la información o requerimientos, una segunda fase de elaboración en donde se realiza el análisis y diseño del mismo, tercera fase llamada construcción en donde se realiza el proceso de implementación del software y artefactos de testing, la última fase es la de transición en donde se realiza la instalación. (Innova, 2013)

Actualmente se pretende adaptar esta metodología para solucionar el problema en particular que se le presenta al área de Investigaciones de la Universidad San Buenaventura sede Medellín, el cual consiste en la carencia de un sistema de información centralizado, organizado, mediante el cual se pueda realizar la generación de un reporte de acuerdo a la información que se encuentra en el repositorio de una base de datos estructurada; se pretende implementar ingeniería web ágil mediante la metodología Scrum en donde se establecen unos logros a cortos plazo en donde se define el alcance para la consecución y ejecución del proyecto, además enfocado en un marco de trabajo basado en el Lenguaje Unificado de Modelado - UML. (Sutherland, 2011)

Durante el desarrollo del proyecto, se encontraran elementos como: requisitos (funcionales y no funcionales), casos de usos prototipos, cronograma de actividades, modelo entidad relacion entre otros, que ayudaran a entender la solución planteada al problema.

2. JUSTIFICACIÓN

La facultad de Ingeniería de la Universidad de San Buenaventura sede Medellín, para el envío de información de los proyectos de investigación al ministerio nacional de educación superior, debe realizar la recolección de datos mediante un flujo de operación extenso debido a la cantidad de reprocesos que conllevan a la ejecución del mismo.

Se plantea diseñar e implementar un sistema de información que permita realizar la administración de los proyectos de investigación de tal modo que permita almacenar, crear, consultar, modificar y eliminar datos de manera organizada, generación de reporte dependiendo del perfil con el que cuente el usuario, actividad cuya disponibilidad no dependa del trabajo de los demás personas involucradas.

3. PLANTEAMIENTO DEL PROBLEMA

Se carece de un sistema que permita gestionar y administrar la información de los proyectos de investigación internos y externos que actualmente se llevan a cabo en la facultad de ingeniería de la Universidad san buenaventura sede Medellín.

En la actualidad la información de los proyectos de investigación se registra en un formato de Excel, proceso que es realizado por la coordinadora de investigaciones Helena Pérez Garces, la cual está generando los siguientes inconvenientes:

- Construcción manual de los reportes con el fin de ser enviados al ministerio nacional de Educación Superior generando mayor probabilidad de errores y demoras.
- Errores ortográficos y de digitación de la información al ser depositada en archivos de Excel.
- La disponibilidad del documento Excel repositario de la información, pues se encontraba en un servidor compartido, donde era necesario esperar que la persona que lo estuviera trabajando finalizara para poderlo utilizar.
- Incompatibilidades información que es reescrita por diferentes fuentes.

4. OBJETIVO GENERAL

Desarrollar un sistema de información que permita administrar y gestionar de forma adecuada la información de los proyectos de investigación de la facultad de ingeniería de la Universidad de San Buenaventura seccional Medellín.

4.1 OBJETIVOS ESPECÍFICOS

- Realizar levantamiento de requisitos con el fin de obtener las necesidades puntuales expresadas por el usuario.
- Crear los prototipos del desarrollo de la aplicación Web.
- Especificar los casos de uso del desarrollo de las funcionalidades que conforman la aplicación.
- Realizar Testing a los procesos o actividades desarrolladas utilizando metodología de Testing ágil.
- Brindar acompañamiento al usuario durante la etapa de estabilización. (1 días)

5. MARCO REFERENCIAL

5.1 ESTADO DE ARTE

Las metodologías ágiles están siendo utilizadas en los últimos años para la elaboración y ejecución de proyectos a corto y mediano plazo las cuales brindan una mayor flexibilización y moldeabilidad en el desarrollo de los sistemas, donde se realiza una colaboración constante y enfoque mayor al usuario o cliente mediante técnicas utilizadas como las historias de usuario donde se describe los requisitos funcionales y no funcionales. En la actualidad se han dado a conocer proyectos a nivel mundial, nacional y regional como desarrollos en la Banca los cuales han optado por escoger metodologías scrum, teniendo como apoyo las múltiples incompatibilidades obtenidas de las experiencias de proyectos realizados tiempo atrás, en donde se repetían soluciones al mismo inconveniente, solicitudes de requerimientos del cliente extensas las cuales podrían simplificarse o priorizando los desarrollos llegando a un acuerdo de entendimiento o negociación con el mismo. (Díaz, 2008)

Empresas grandes como Firestone, Bancolombia, informan el éxito de sus proyectos mediante la aplicación de metodologías ágiles los cuales conllevan a la obtención de resultados próximos, han realizado implementación de portales web ágiles, sistemas de gestión de calidad, procesos internos automáticos enfocados con el negocio del cliente, entre otros. (PSL, 2012)

Observamos en la actualidad que a pesar de las diferentes metodologías existentes para la ejecución de sistemas de información para los proyectos de investigación, se ha visto la balanza en el mercado y las compañías desarrolladoras la inclinación por realización de proyectos prácticos, ágiles, que permitan sufrir cambios en cualquier momento sin que afecte los entregables de acuerdo al cronograma, que genere una constante interacción con el cliente para poderlo hacer parte del proyecto en donde sería de gran importancia. (Medina, 2011)

Fase Heurística:

Los sistemas de información son indispensables en nuestra sociedad actual ya que nos permite realizar una gestión organizada de los datos, automatización de procesos operativos para realizar toma de decisiones en algún proceso determinado basado en experiencias obtenidas empíricamente. A continuación mostraremos algunos sistemas de información y su funcionalidad:

Sisav (Sistema de Información del Sector Agropecuario del Valle del Cauca) Sistema enfocado al agro en donde muestra en una guía los recursos informativos nacionales y mundiales de dominio público con el objetivo de incentivar a la producción e innovación.

Eduteka (Sistema de información web implementado para docentes y directivos interesados en mejorar la calidad de aprendizaje de las Tics)

Bimbo SCW (Sistema de información, gestión de la cadena de suministros de alimentos) En este caso, al ver el crecimiento significativo de la compañía se notó la necesidad de realizar un sistema de información que permitiera optimizar la cadena de suministro para mantener la frescura de sus alimentos los cuales son perecederos.

SOFIA (Sistema Optimizado para la Formación Integral del Aprendizaje) Es un sistema de información educativo centralizado donde se habilitan los cursos para la gestión de la formación profesional y no formal.

5.2 MARCO CONCEPTUAL

5.2.1 Modelamiento ágil:

Son metodologías efectivas basadas en la practicidad y flexibilidad que se le dan a los recursos para la ejecución de proyectos o desarrollos de forma ligera y sencilla. Un ejemplo de un modelamiento ágil sería la consecución de información con el cliente por medio de un papel en el que se describa la necesidad puntual. Algunas de las metodologías ágiles son XP, DSDM o SCRUM. Para las metodologías ágiles el principal agente es la gente o usuarios ya que de la mano se logran establecer comunicaciones estrechas donde se llega a la idea de involucrarlo o hacerlo parte activa del proyecto para poder ser exitoso. Es un modo de desarrollo que se adapta a las circunstancias de la evolución del proyecto. (Ambler, 2002)

5.2.2 Testing ágil:

Son prácticas de pruebas ágiles las cuales se basan en las metodologías ágiles, se integran los tester al grupo de desarrollo para trabajar mancomunadamente con el cliente para la realización de pruebas funcionales. En las metodologías ágiles, el tester al verse implicado el grupo de desarrollo y las personas o clientes, juega un papel importante ya que trabaja en la consecución de ideas o automatización de pruebas, usabilidad de la aplicación, capacidad de aceptación al cambio de requerimientos, colaboración con los desarrolladores como los clientes o usuarios. (Araújo, 2004)

5.2.3 Técnicas Ágiles en el desarrollo de Software:

Está enfocado a los equipos de desarrollo y programación mentalizados en ofrecer "calidad" en sus entregables, todo esto se logra satisfaciendo las necesidades del cliente.

Mediante el desarrollo del proyecto se debe contemplar la ejecución de cada iteración la cual comprende:

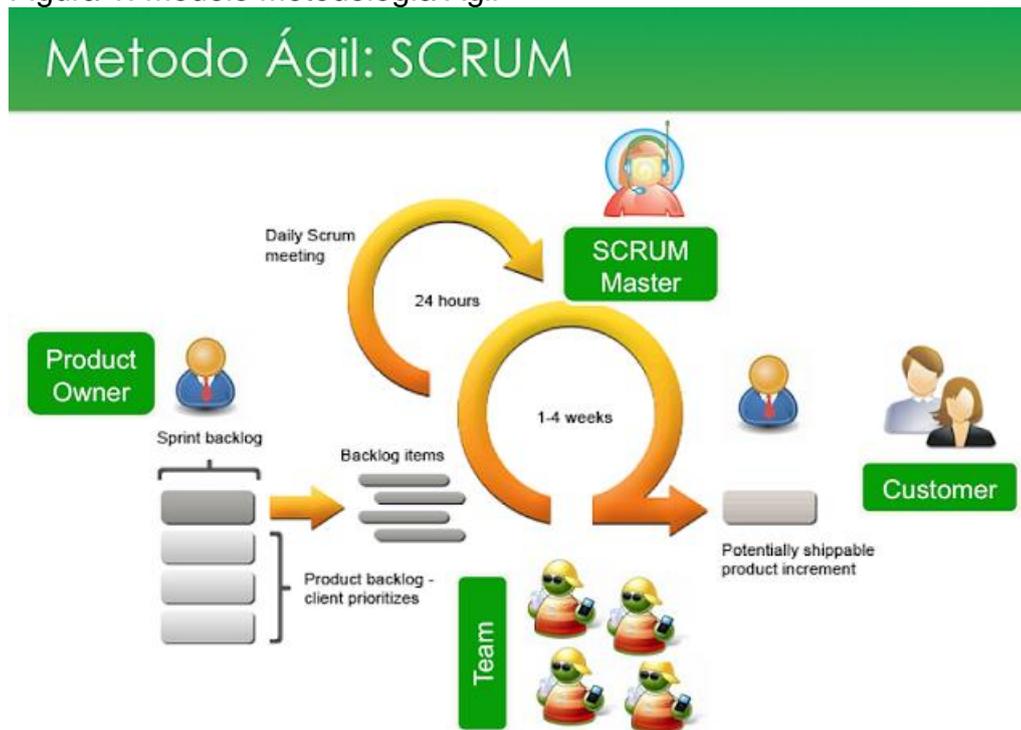
- Planificación de la iteración en la cual se realizan los estudios de riesgos.
- Análisis de los casos de uso.
- Codificación y testing
- Evaluación de los diferentes prototipos en función de pruebas y criterios definidos.
- Preparación de la entrega la cual comprende la documentación alcanzada y las pruebas funcionales. (Adelaida Valbín Landeo, 2010)

5.2.4 Ingeniería de Software:

Según la definición del IEEE, citada por [\[Lewis 1994\]](#) "**software** es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo". Según el mismo autor, "un producto de software es un producto diseñado para un usuario" En base a este concepto, la Ingeniería de Software es la rama de la ingeniería que provee un enfoque sistemático disciplinado y cuantificable al desarrollo, operación y posterior mantenimiento del software, con el fin de obtener un resultado final o producto que sea eficaz, eficiente y que pueda ser pagado. (Wikipedia, 2012)

Se realiza la adaptación de la metodología Scrum en el proyecto de grado debido a que ésta se realiza como un marco del trabajo mediante la cual se puede asignar varios procesos y técnicas, éste modelo está diseñado para ser flexible y adaptable a cualquier proyecto a corto y mediano plazo.

Figura 1. Modelo Metodología Ágil



En la figura 1 se describe gráficamente la metodología implementada en el proyecto, metodología Scrum.

6. FASES METODOLÓGICAS

El presente proyecto de investigación, pretende mejorar la sistematización de los datos concernientes a los proyectos de investigación pertenecientes a la facultad de ingenierías de la Universidad de San Buenaventura Sede Medellín, lo cual se verá reflejado en la optimización de tiempo y facilidad en el proceso de generación de reportes para ser enviados al eje central la Universidad San Buenaventura sede Bogotá donde una vez validada la información es enviada al Ministerio Nacional de Educación Superior.

Para el desarrollo de la aplicación, se estipularon las siguientes fases metodológicas:

Fase 1: Realizar el análisis de requisitos con el fin de obtener las necesidades puntuales expresadas por el usuario

El desarrollo de esta fase, se ejecuta mediante el cumplimiento de las siguientes actividades:

- Identificación de los usuarios
- Planeación y ejecución de reuniones con el usuario para la exposición de la necesidad.
- Actividades de identificación de las necesidades puntuales, con el fin de comprender la situación problema, los procesos actuales que desean ser automatizados, y recolección de información adicional. Esto se realiza por medio de documentos guías externos, correos con cuestionarios pregunta, entre otros.

Fase 2: Especificar los casos de uso del desarrollo de las funcionalidades que conforman la aplicación.

Para realizar el diseño de la interfaz gráfica de usuario, se toman como base los principios de usabilidad web, definiendo así las siguientes tareas:

- El diseño de los casos de Uso se realiza mediante la especificación detallada de cada uno de los requisitos. Dicha especificación se realiza por medio de los casos UML, con la utilización de plantillas que permiten a cualquier lector del caso de uso el entendimiento del mismo.

Fase 3: Crear los prototipos del desarrollo de la aplicación Web

El diseño de arquitectura de la solución es el diagrama mediante el cual se pretende mostrar la relación de los componentes de la aplicación en conjunto con el diseño de la base de datos de la misma. Para esto es necesario:

- Validación de los posibles componentes externos que se relacionarán con la aplicación a desarrollar.

- Es necesario tener los casos de uso completos, para determinar el modelo de la aplicación y su interacción con todos los elementos involucrados.
- Definición de los campos, tipos de datos, llaves primarias y descripción de los mismos, con el fin de diseñar el modelo de base de datos de la aplicación.

Fase 4: Realizar Testing a los procesos o actividades desarrolladas utilizando metodología de Testing ágil

En la consecución del presente objetivo específico, es debe contar con:

- Empezaremos con las pruebas de exploración a medida que se vaya completando el desarrollo.
- Realizar el uso de ejemplos, en cada funcionalidad describimos un ejemplo para ver las posibles acciones que tomaría un usuario final.
- Planeación y ejecución de los casos prueba funcionales, a completitud y consistencia de datos en la aplicación, garantizando la “calidad” en la entrega de lo que se tiene esperado por el cliente.
- Realizar diferentes pruebas de stress para verificar cómo se comporta el sistema ante un número de usuarios determinado.
- Corrección de los bugs encontrados en la ejecución de las pruebas, y nueva ejecución de las pruebas. En base a la metodología Scrum las pruebas deben ser desde el comienzo para que permita realizar correcciones en caliente.

Fase 5: Brindar acompañamiento al usuario durante la etapa de estabilización. (1 días)

Para la consecución de esta tarea, es necesario tener presente los siguientes puntos:

- Realizar asesorías a los diferentes implicados en el proceso de familiarización con el entorno de la aplicación.
- Despejar dudas e inquietudes que puedan presentarse en el momento de la entrega.

7. CRONOGRAMA

A continuación observaremos la ejecución del cronograma empleado para la realización del proyecto Snies.

Figura 2. Cronograma proyecto Snies

Proyecto			
Proyecto SNIES			
N° de sprint	Inicio	Días	Jornada
1	10-abr-13	6	Nocturna
2	16-abr-13	6	Nocturna
3	22-abr-13	7	Nocturna
4	29-abr-13	5	Nocturna
5	4-may-13	5	Nocturna
6	9-may-13	5	Nocturna
7	14-may-13	5	Nocturna
8	19-may-13	5	Nocturna
9	24-may-13	4	Nocturna
10	28-may-13	7	Nocturna

TAREAS		EQUIPO	FESTIVOS
TIPOS	ESTADOS		
Reunion	Terminada	Harold y Dorian	1-may
Analisis	En Curso	Harold y Dorian	13-may
Asesorias	En Curso	Harold y Dorian	
Gestion de Proyecto	En Curso	Harold y Dorian	
Prototipo	En Curso	Harold y Dorian	
Codificacion	En Curso	Harold y Dorian	
Calidad	Sin empezar	Harold y Dorian	
Pruebas	En Curso	Harold y Dorian	

En la figura número 2 observamos el cronograma empleado para el desarrollo del proyecto Snies, se observan los días de inicio, duración de cada sprint, tipos de tareas, conformación del equipo de trabajo y ejecución de tareas.

8. APLICACIÓN DEL MODELO INGENIERIA DE SOFTWARE BASADO EN LA METODOLOGÍA SCRUM

A continuación se describe la implementación de la metodología de trabajo Scrum para el desarrollo del proyecto Snies.

Descripción General de la Metodología Scrum proyecto Snies

Para el desarrollo del proyecto Snies se opta por una de las metodologías ágiles más conocida como Scrum, debemos tener presente que Scrum es un marco de trabajo que facilita la manera o forma de trabajar de las personas del equipo. Esta metodología utiliza un proceso iterativo en donde cada iteración se llama sprint, con los sprints se busca en lo posible que sean lo más corto en cuanto al tiempo de ejecución de las tareas, dichos sprint van de la mano con la planificación, ejecución, pruebas. Esta metodología utiliza ceremonias de tiempo limitado para planificar, inspeccionar, corregir, y adaptar durante cada sprints de trabajo. Antes de la ejecución de cada sprint se debe pactar compromisos de entrega y planificación del trabajo, al final de cada sprint se muestran los resultados al cliente.

Fundamentación

Las principales razones del uso de un ciclo de desarrollo iterativo e incremental de tipo Scrum para la ejecución de este proyecto son:

- Sistema modular: Las características del sistema Snies permiten desarrollar una base funcional mínima y sobre ella ir incrementando las funcionalidades o modificando el comportamiento o apariencia de las ya implementadas.
- Entregas frecuentes y continuas al cliente de los módulos terminados.
- Es posible que el sistema incorpore más funcionalidades de las inicialmente identificadas.
- Es posible que durante la ejecución del proyecto se altere el orden en el que se desean recibir los módulos o historias de usuario terminadas.

Valores de Trabajo en Equipo

Los valores que deben ser practicados por todos los miembros involucrados en el desarrollo y que hacen posible que la metodología Scrum tenga éxito son:

- Autonomía del equipo
- Colaboración
- Respeto en el equipo
- Responsabilidad y auto-disciplina
- Foco en la tarea
- Información, transparencia y visibilidad
- Calidad

A continuación se detallan las fases ejecutadas bajo el marco de trabajo Scrum.

8.1 Definición de Roles:

Debido a que el recurso humano se limita a dos Ingenieros para la ejecución de este proyecto, los mismo han tomado diferentes roles para desarrollar a conformidad las actividades planteadas por el marco de trabajo.

Product Owner o Dueño del Producto: Dorian Ardila, es el encargado de ser el representante de todas las personas involucradas en el proyecto ya sea personal interno o externo, actúa como un canal de comunicación, define los objetivos del producto del proyecto, además es el encargado de llevar a cabo el ROI (Retorno de inversión).

Scrum Master: Harold Florez, es el encargado de liderar el equipo con el objetivo de llevarlos a todos encaminados a un solo objetivo siguiendo las reglas y los procesos de la metodología Scrum. Además es el encargado de velar para que la lista de requerimientos se encuentre lista antes de continuar con la siguiente iteración, es el encargado de realizar las reuniones diarias (daily meeting) con el equipo de trabajo con el fin de realizar validaciones acerca del avance e inconvenientes presentados en el desarrollo del proyecto, protege al equipo de interrupciones externas que pueda tener el equipo con el objetivo de afianzar la productividad.

Equipo de Trabajo o Team Scrum: Grupo de personas conformadas por Dorian Ardila, Harold Florez, los cuales son personas con los requisitos establecidos de la metodología 100% comprometidas con el grupo de trabajo, confianza en los demás miembros del grupo, descartando el ego profesional que pueda llevar a la afectación del de la ejecución del proyecto, se hacen responsables durante todo el proyecto del trabajo que realizan Tanto en el desarrollo de cada iteración como en la calidad del mismo, cada miembro del equipo se auto asigna tareas y se realiza un estimado de tiempo y esfuerzo para ejecutarlas.

8.2 Personas y roles del proyecto Snies

Persona	Contacto	Rol Desempeñado
Harold Enoc Flórez Perez	Hflorezp2010@hotmail.com	Scrum Manager
Dorian Norberto Ardila Puentes	dardilap@gmail.com	Product Owner

Una vez definidos los roles y los alcances de cada integrante del equipo de trabajo se proceden a realizar el cronograma de reuniones semanales para definir los Product Backlog.

Product Backlog: Es el conjunto de requisitos denominados en el proceso como historias de usuario, descritos en un lenguaje sin tecnicismo en donde se prioriza el ROI (retorno de la inversión) considerando su costo beneficio, durante la ejecución del proyecto se van realizando los ajustes a los requisitos y prioridades en intervalos de tiempo regulares. (Engineers, 2013)

Figura 3. Historias de Usuario

Identificador (ID) de la Historia	Enunciado de la Historia				Criterios de Aceptación			
	Rol	Característica / Funcionalidad	Razón / Resultado	Número (#) de Escenario	Creación de Usuarios	Contexto	Evento	Resultado / Comportamiento esperado
1	Como un (Cliente)	Necesito crear usuarios y roles en el sistema	Con la finalidad de permitirle a los profesores Ingresar datos de los proyectos y diferenciarlos del usuario administrador del sistema	1	Permitir el acceso	En caso que se encuentren creados en el sistema	cuando se realice el proceso de autenticación	Acceso al Sistema
				2	Denegar el acceso	En caso que no se encuentren creados en el sistema	cuando se realice el proceso de autenticación	Denegación Acceso al Sistema
				3	Tipo de Usuario	Cargar las opciones correspondientes al perfil del usuario creado según sus rol	Se cree el usuario y se asigne el rol	El sistema permite crear el usuario y asignarle el rol de administrador o usuario del sistema

En la figura número 3 se describe las historias de usuario junto con los criterios de aceptación y los resultados esperados.

8.3 Sprint

En la metodología Scrum el Sprint es el período en el cual se lleva a cabo el una iteración o trabajo, para la ejecución del Sprint se recomienda que el tiempo estimado sea constante y definido previamente por el equipo de trabajo. La duración del sprint en particular puede alcanzar 2 ó 3 semanas dependiendo del proceso de ejecución (celeridad del equipo) y dificultades que puedan encontrarse en el desarrollo de las iteraciones los cuales deben ser catalogados como un riesgo.

- Reunión de inicio de sprint: Reunión para determinar las funcionalidades o historias de usuario que se van a incluir en el próximo incremento.

Se realizaron reuniones previas al comienzo de la ejecución de cada sprint, definiendo previamente el objetivo, tiempo estimado y el alcance de cada iteración.

Responsabilidades del gestor de producto

- Asistencia a la reunión
- Exposición y explicación de las historias que necesita para la próxima iteración y posibles restricciones de fechas que pudiera tener.

Responsabilidades del Scrum Manager

- Moderación de la reunión

Responsabilidades del equipo técnico

- Confección de la pila del sprint.
- Auto-asignación del trabajo.

8.4 Cuadro de reuniones

- Reunión técnica los lunes: Reuniones diarias con el equipo, contaba con presencia del Coordinador del proyecto o Scrum Manager de duración máxima de 10 minutos.

Debido a las circunstancias en la cuales se desarrollaba el proyecto, no se compartía la misma sede física de trabajo, debido a que los integrantes del proyecto se encuentran laborando en diferentes organizaciones, no se realizaban reuniones diarias tal como lo sugiere el marco de trabajo Scrum, por el contrario se realizaban reuniones de trabajo todos los lunes en un espacio de tiempo corto no más de 30 minutos con el objetivo de validar el estado de los avances o inconvenientes presentados durante la ejecución del Sprint.

Responsabilidades del Scrum Manager

- Supervisión de la reunión y anotación de las necesidades o impedimentos que pueda detectar el equipo.
- Gestión para la solución de las necesidades o impedimentos detectados por el equipo.

Responsabilidades del equipo técnico

- Comunicación individual del trabajo realizado el día anterior y el previsto para día actual.
- Actualización individual del trabajo pendiente.

- Notificación de necesidades o impedimentos previstos u ocurridos para realizar las tareas asignadas.

8.5 Cuadro Reuniones Técnicas

- Reunión de cierre de sprint y entrega: Reunión para probar y entregar el incremento al gestor del producto.

Reunión realizada con el objetivo de recibir el trabajo que fue ejecutado y no por los miembros del equipo. El trabajo completado debe ser entregado al cliente en modo de “demo”, se realiza un acta de entrega en la cual conste el cumplimiento.

En caso que este trabajo no se encuentre completo, no se puede presentar al cliente, se expone los motivos por los cuales presentó retraso o percance que pueda haber alterado la ejecución del mismo, esto en un ambiente laboral puede acarrear a sanciones o multas según lo definido contractualmente.

Características:

- Prácticas: sobre el producto terminado, no sobre simulaciones o imágenes).
- De tiempo acotado máximo de 2 horas.

Responsabilidades del gestor de producto

- Asistencia a la reunión.
- Recepción del producto o presentación de reparos.

Responsabilidades del Scrum Manager

- Moderación de la reunión

Responsabilidades del equipo técnico

- Presentación del incremento.

8.6 Pila de producto (Sprint Backlog)

Es el equivalente a los requisitos del sistema o del usuario. Se realiza una lista de tareas las cuales realiza el equipo de trabajo como un plan para cumplir con los objetivos o requisitos expresados por el cliente, en esta lista podemos ordenar adecuadamente en orden descendente los requisitos que tengan mayor prioridad según lo expresado por el cliente.

Figura 9. Sprint 6

SPRINT		INICIO	DURACIÓN								
6		9-may-13	5								
Tareas pendientes											
Total Horas Estimadas		42									
Total Horas Ejecutadas		46									
				9-may	10-may	11-may	12-may	13-may			
				9	7	10	11	9			
PILA DEL SPRINT											
Backlog	Tarea	Tipo	Estad	Responsabi	Estimacion inicial	ESFUERZO				Esfuerzo Ejecutado x Tarea	
B-31	Creación, consulta, modificación y eliminación de registros de las listas desplegables que se encontrarán en los diferentes formularios del sistema.	Analisis	Terminada	Harold y Donian	18	4	2	5	3	4	36
		Prototipo		Harold y Donian							
		Codificación		Harold y Donian							
		Pruebas		Harold y Donian							
B-32	Permitir asociar un tipo de usuario en el momento de la creación o edición de los mismos	Analisis	Terminada	Harold y Donian	7	2	1	3	2	1	16
		Prototipo		Harold y Donian							
		Codificación		Harold y Donian							
		Pruebas		Harold y Donian							
B-33	Permitir almacenar un registro sin que todos los campos obligatorios estén diligenciados correctamente en nuevos Centros de Investigación	Analisis	Terminada	Harold y Donian	5	1	2	1	1	1	10
		Prototipo		Harold y Donian							
		Codificación		Harold y Donian							
		Pruebas		Harold y Donian							
B-34	Permitir almacenar un registro sin que todos los campos obligatorios estén diligenciados correctamente en nuevos registros de información	Analisis	Terminada	Harold y Donian	4	1	1	1	1	1	8
		Prototipo		Harold y Donian							
		Codificación		Harold y Donian							
		Pruebas		Harold y Donian							
B-35	Permitir almacenar un registro sin que todos los campos obligatorios estén diligenciados correctamente en nuevos proyectos de investigación	Analisis	Terminada	Harold y Donian	5	2	3	1	1	1	11
		Prototipo		Harold y Donian							
		Codificación		Harold y Donian							
		Pruebas		Harold y Donian							
B-36	Permitir almacenar un registro sin que todos los campos obligatorios estén diligenciados correctamente en nuevos Grupos de Investigación	Analisis	Terminada	Harold y Donian	3	1	1	1	1	1	7
		Prototipo		Harold y Donian							
		Codificación		Harold y Donian							
		Pruebas		Harold y Donian							

Figura 10. Sprint 7

SPRINT		INICIO	DURACIÓN								
7		14-may-13	5								
Tareas pendientes											
Total Horas Estimadas		18									
Total Horas Ejecutadas		20									
				14-may	15-may	16-may	17-may	18-may			
				3	5	5	3	4			
PILA DEL SPRINT											
Backlog	Tarea	Tipo	Estad	Responsabi	Estimacion inicial	ESFUERZO				Esfuerzo Ejecutado x Tarea	
B-37	No permitir almacenar un registro sin que todos los campos obligatorios estén diligenciados correctamente en nuevas Redes de Cooperación	Analisis	Terminada	Harold y Donian	5	1	1	2	1	1	10
		Prototipo		Harold y Donian							
		Codificación		Harold y Donian							
		Pruebas		Harold y Donian							
B-38	No permitir almacenar un registro sin que todos los campos obligatorios estén diligenciados correctamente en nuevos Productos de Investigación	Analisis	Terminada	Harold y Donian	4	1	1	1	1	1	8
		Prototipo		Harold y Donian							
		Codificación		Harold y Donian							
		Pruebas		Harold y Donian							
B-39	No permitir almacenar un registro sin que todos los campos obligatorios estén diligenciados correctamente en nuevos Investigadores Externos	Analisis	Terminada	Harold y Donian	5	1	2	2	1	1	11
		Prototipo		Harold y Donian							
		Codificación		Harold y Donian							
		Pruebas		Harold y Donian							
B-40	No permitir eliminar un maestro, pues esta eliminación corresponde solamente a los valores que componen cada uno de ellos	Analisis	Terminada	Harold y Donian	4	1	1	1	1	2	9
		Prototipo		Harold y Donian							
		Codificación		Harold y Donian							
		Pruebas		Harold y Donian							

Figura 11. Sprint 8

SPRINT		INICIO	DURACIÓN								
8		19-may-13	5								
Tareas pendientes											
Total Horas Estimadas		26									
Total Horas Ejecutadas		26									
				19-may	20-may	21-may	22-may	23-may			
				3	4	5	6	8			
PILA DEL SPRINT											
Backlog	Tarea	Tipo	Estad	Responsabi	Estimacion inicial	ESFUERZO				Esfuerzo Ejecutado x Tarea	
B-41	Habilitar las opciones Agregar Fuente Nacional, Agregar Fuente Internacional y Agregar Gasto a los usuarios tipo Administrador	Analisis	Terminada	Harold y Donian	6	1	1	1	1	2	12
		Prototipo		Harold y Donian							
		Codificación		Harold y Donian							
		Pruebas		Harold y Donian							
B-42	Habilitar las opciones Agregar Entidad y Agregar IES a los usuarios tipo Administrador	Analisis	Terminada	Harold y Donian	4		2	2			8
		Prototipo		Harold y Donian							
		Codificación		Harold y Donian							
		Pruebas		Harold y Donian							
B-43	Los registros de investigación están relacionados a proyectos de investigación previamente ingresados, para fines de eliminación, y no permitir la misma acción.	Analisis	Terminada	Harold y Donian	12	1	2	2	2	5	24
		Prototipo		Harold y Donian							
		Codificación		Harold y Donian							
		Pruebas		Harold y Donian							
B-44	Permitir eliminar valores de las listas desplegables, pero dichas listas desplegables (campos) no se deberán eliminar del sistema.	Analisis	Terminada	Harold y Donian	4	1	1	1	1	1	8
		Prototipo		Harold y Donian							
		Codificación		Harold y Donian							
		Pruebas		Harold y Donian							

Figura 12. Sprint 9

SPRINT		INICIO	DURACIÓN									
9		24-may-13	4									
Tareas pendientes												
Total Horas Estimadas												
Total Horas Ejecutadas												
					V	S	D	L				
					24-may	25-may	26-may	27-may				
					11	13	12	12				
				Horas de trabajo ejecutadas								
PILA DEL SPRINT												
Backlog	Tarea	Tipo	Estado	Responsabl	Estimacion inicial	ESFUERZO				Esfuerzo Ejecutado x Tarea		
B-45	Permitir el acceso a los usuarios que se encuentren creados en la base de datos	Analisis	Terminada	Harold y Dorian	13	4	3	3	3	26		
		Prototipo		Harold y Dorian								
		Codificación		Harold y Dorian								
		Pruebas		Harold y Dorian								
B-46	Denegar el acceso al sistema a los usuarios que no se encuentren en la base de datos de la aplicación	Analisis	Terminada	Harold y Dorian	10	2	2	4	2	20		
		Prototipo		Harold y Dorian								
		Codificación		Harold y Dorian								
		Pruebas		Harold y Dorian								
B-47	Bloquear un usuario en caso de que se ingresen claves erradas en tres ocasiones para el mismo	Analisis	Terminada	Harold y Dorian	14	3	5	2	4	28		
		Prototipo		Harold y Dorian								
		Codificación		Harold y Dorian								
		Pruebas		Harold y Dorian								
B-48	Mostrar mensajes de alerta en caso de que se presenten eventos Usuario o contraseña incorrectas	Analisis	Terminada	Harold y Dorian	12	2	3	3	3	23		
		Prototipo		Harold y Dorian								
		Codificación		Harold y Dorian								
		Pruebas		Harold y Dorian								

Figura 13. Sprint 10

SPRINT		INICIO	DURACIÓN									
10		28-may-13	7									
Tareas pendientes												
Total Horas Estimadas												
Total Horas Ejecutadas												
					M	X	J	V	S	D	L	
					28-may	29-may	30-may	31-may	1-jun	2-jun	3-jun	
					1	4	4	3	6	2	2	
				Horas de trabajo ejecutadas								
PILA DEL SPRINT												
Backlog	Tarea	Tipo	Estado	Responsabl	Estimacion inicial	ESFUERZO				Esfuerzo Ejecutado x Tarea		
B-49	Permitir generar el reporte de proyectos de investigación, con la información suministrada a través de los módulos anteriores	Analisis	Terminada	Harold y Dorian	4	1	1	1	1	2	10	
		Prototipo		Harold y Dorian								
		Codificación		Harold y Dorian								
		Pruebas		Harold y Dorian								
B-50	El sistema debe generar el reporte en formato excel .xls.	Analisis	Terminada	Harold y Dorian	4	1	1	1	1	2	9	
		Prototipo		Harold y Dorian								
		Codificación		Harold y Dorian								
		Pruebas		Harold y Dorian								
B-51	Documentación Proyecto	Analisis	Terminada	Harold y Dorian	10	2	2	1	2	2	21	
		Prototipo		Harold y Dorian								
		Asesorias		Harold y Dorian								
				Harold y Dorian								

8.7 Requisitos No Funcionales

Requisitos de Software

- Lenguaje donde se desarrolló el sistema de información fue el framework de Kumbia PHP
- La aplicación es desarrollada sobre el motor de Base de datos SQL Server, pero hay que tener en cuenta que KumbiaPHP es compatible con motores de base de datos como MySQL, PostgreSQL y Oracle.
- La solución debe correr en los exploradores: Chrome, Internet Explorer, Mozilla Firefox.
- La solución es compatible con otros sistemas operativos debido a que es multiplataforma como Windows y Unix con un servidor web y PHP5 instalado.

Requisitos de Hardware

Procesador:

- 1 Procesador Intel Xeon Procesadores 3.0 Ghz.
- 1 Procesador AMD Opteron 2200 Series.

Memoria Cache:

4 MB

8.9 LISTADO DE ACTIVIDADES

A continuación se describe el listado de actividades a ejecutar en el proyecto.

Definición del cronograma de trabajo

- Definir un cronograma, con el fin de delimitar el esfuerzo para cada una de las etapas o fases del desarrollo de la aplicación.
- Definición de las actividades que componen el proyecto.
- Validación de los tiempos requeridos para cada una de estas actividades.
- Asignación de las actividades a los integrantes del equipo de trabajo.

Diseñar la interfaz gráfica de la aplicación

- Validación con el usuario de preferencia de color o plantillas prediseñadas estándar para ser utilizada como fondo, teniendo en cuenta que no debe ser recargado, fuerte y que contraste.
- Diseñar la interfaz gráfica de la aplicación, de manera que cumpla con los requisitos mínimos de usabilidad web.
- Determinación de tamaño y tipo de letra, de acuerdo al estándar manejado por la universidad.
- Diseño del menú de navegación del usuario en la aplicación, y disposición de los botones y opciones que el usuario podrá utilizar en la aplicación.

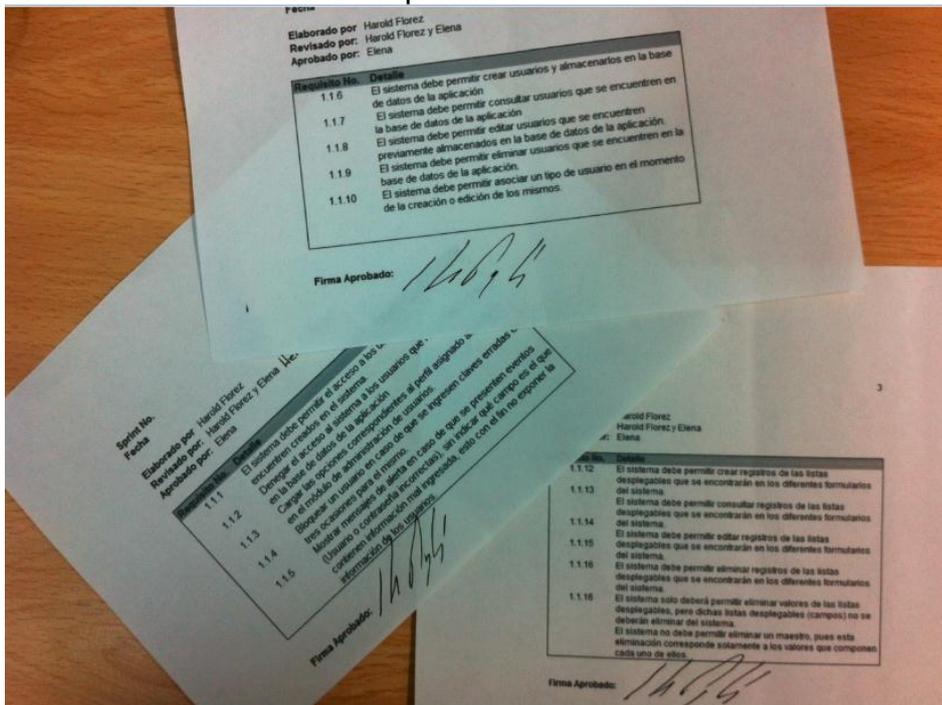
Aprobaciones por parte del usuario para realizar el paso a producción

- Planeación y ejecución de los casos prueba funcionales, a completitud y consistencia de datos en la aplicación.
- Corrección de los bugs encontrados en la ejecución de las pruebas, y nueva ejecución de las pruebas.
- Reuniones de prueba de los casos diseñados anteriormente, con el fin de tener las aprobaciones correspondientes del mismo y realizar la puesta en producción de la aplicación.

Priorización de los requisitos basados en las historias de usuarios.

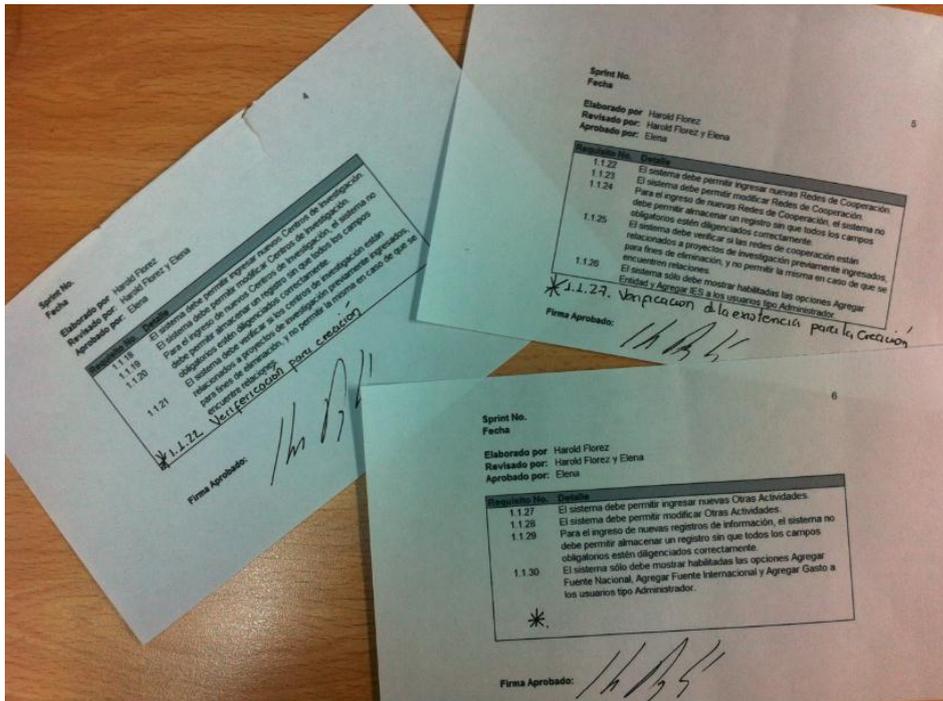
Al momento de realizar la priorización de los requisitos, debemos tener en cuenta que los contextos, escenarios, la tecnología, el equipo pueden tender a cambiar, es por esto que siempre debemos priorizar, no solo en el principio del proyecto sino en cada iteración. Ver anexo 1

Figura 16. Historia de Usuario Aprobado



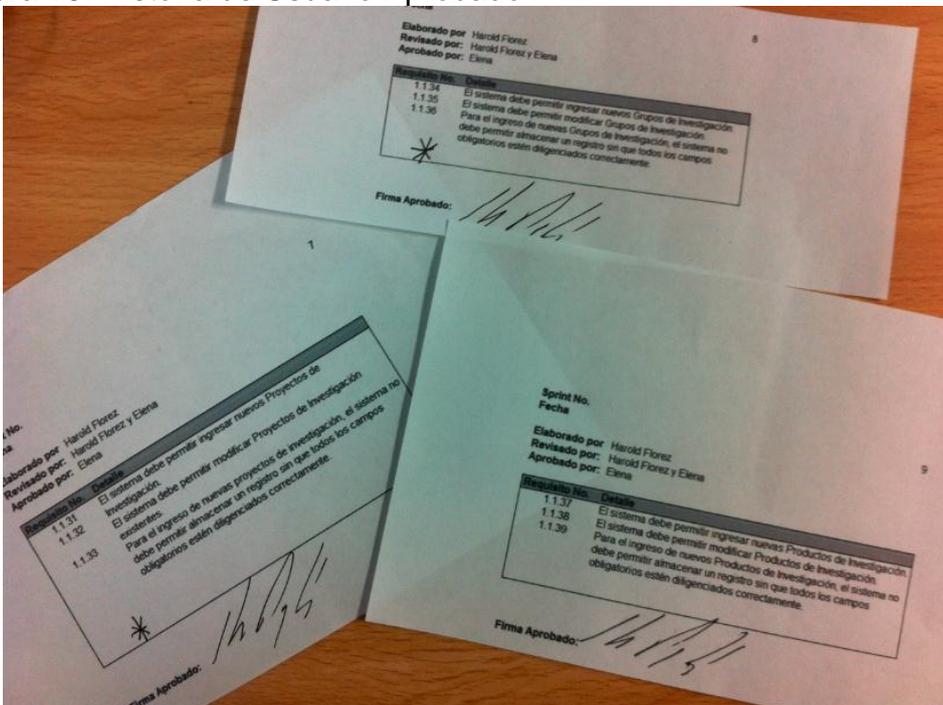
En la figura número 16 observamos los sprints, una vez elaboradas son revisadas y aprobadas por el cliente.

Figura 17. Historia de Usuario Aprobado



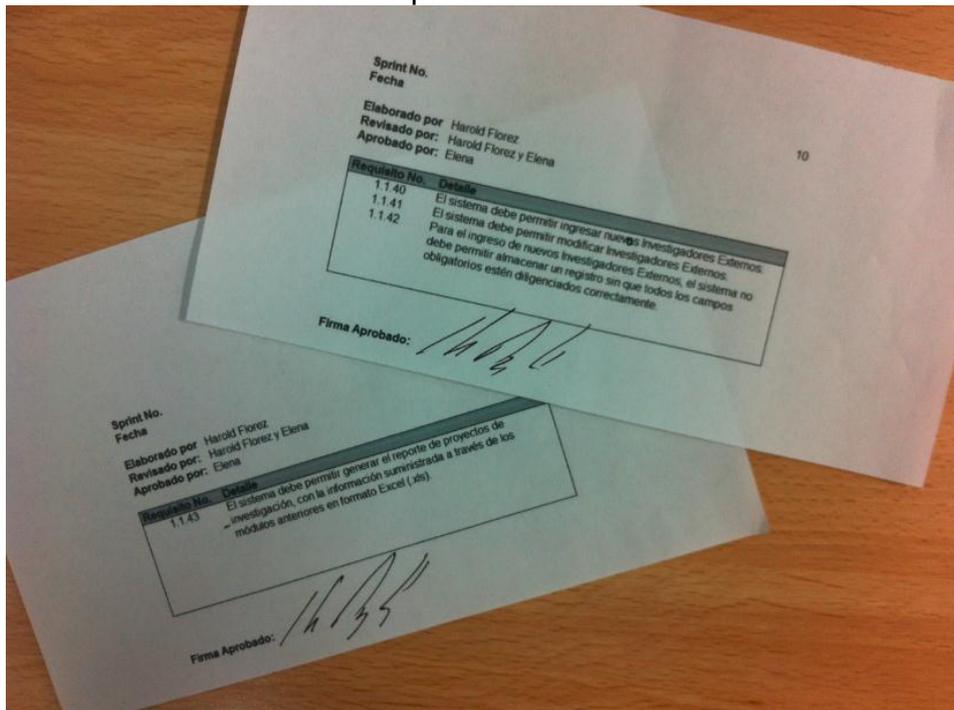
En la figura número 17 observamos los sprints, una vez elaboradas son revisadas y aprobadas por el cliente.

Figura 18. Historia de Usuario Aprobado



En la figura número 18 observamos los sprints, una vez elaboradas son revisadas y aprobadas por el cliente.

Figura 19. Historia de Usuario Aprobado



En la figura número 19 observamos los sprints, una vez elaboradas son revisadas y aprobadas por el cliente.

8.10 DETALLE DE LA SOLUCIÓN

A continuación se realiza la descripción del modelo de la solución del diseño y arquitectura del proyecto Scrum.

8.10.1 Diseño y Arquitectura de la solución

La solución presenta un diseño por capas, donde cada una de ellas realiza la abstracción y el encapsulamiento de la funcionalidad necesaria para que cada uno de los componentes del sistema pueda inter operar de forma autónoma y no dependiente de una tecnología o lógica particular, haciendo que cada uno de los componentes del sistema puedan ser rápidamente intervenidos, actualizados, reemplazados, según los requerimientos operativos del negocio o según las necesidades de nueva funcionalidad lo vayan demandando.

8.10.2 Arquitectura N-Capas

La arquitectura de software bajo un modelo n-capas, plantea la abstracción de funcionalidad para cada capa según las interacciones con los diferentes componentes del sistema y los requerimientos funcionales y no funcionales que esta deba cumplir.

Para el diseño se planteó una Arquitectura a 3 Capas que permita realizar la abstracción de funcionalidad de Acceso a Datos, Lógica de Negocios y Presentación.

A continuación se describe la funcionalidad de cada una de las capas utilizadas en la arquitectura de la solución:

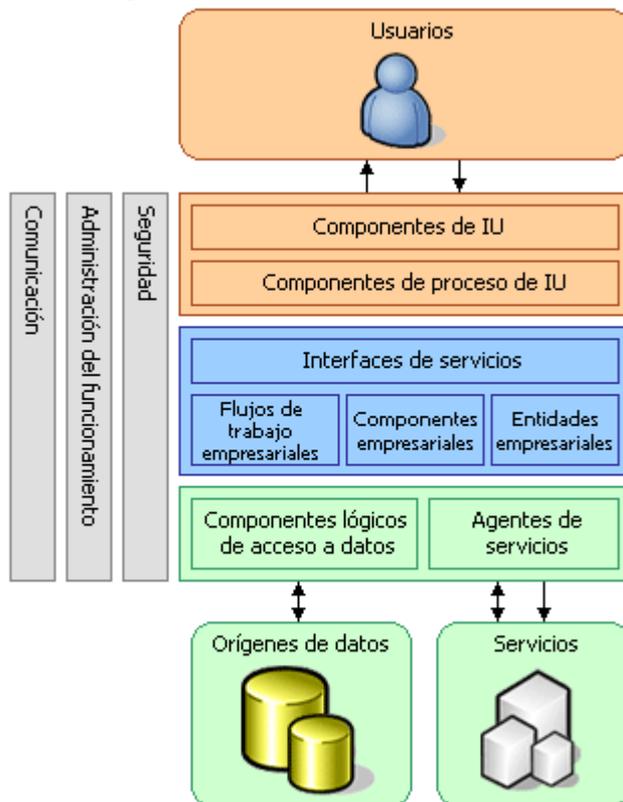
8.10.2.1 Capa de Acceso a Datos: Se encarga de realizar las conexiones directamente a la Bases de Datos SQL Server y gestionar todas las consultas directamente con la base de datos, para luego subir los datos obtenidos a la capa de lógica de negocio. Esta capa abstrae toda la funcionalidad de comunicación con la base de datos y encapsula todos los procedimientos necesarios para las Transacciones de tipo DDL.

8.10.2.2 Capa de Lógica de Negocio: Se encarga de proveer y abstraer toda la funcionalidad necesaria para dar cumplimiento a los requisitos funcionales de la solución. En esta capa se programa toda la lógica de negocio, como validaciones y procesamiento de datos hasta la obtención de información. Se comunica directamente con su nivel inferior la Capa de Datos quien genera las entradas al procesamiento y con su nivel superior la capa de presentación en donde se proporcionan las salidas del procesamiento ("Información").

8.10.2.3 Capa de Presentación: Proporciona la Interfaz gráfica que permite a los usuarios del sistema interactuar con los datos-información contenida en los repositorios de datos. Se comunica única y exclusivamente con la capa de lógica de negocio.

8.10.3 Diagrama de la Arquitectura

Figura 20.



En la figura 20 observamos el diagrama de arquitectura de aplicaciones .NET capas de componentes de servicios.

Diagrama de Referencia: Arquitectura de aplicaciones de .NET: Diseño de aplicaciones y servicios

<http://msdn.microsoft.com/es-es/library/ms954595.aspx>

8.11 CASOS DE USO

8.11.1 Requisitos Funcionales Acceso al Sistema

REQ 1.1.1 El sistema debe permitir el acceso a los usuarios que se encuentren creados en el sistema.

REQ 1.1.2 Denegar el acceso al sistema a los usuarios que no se encuentren en la base de datos de la aplicación

REQ 1.1.3 Cargar las opciones correspondientes al perfil asignado al usuario en el módulo de administración de usuarios.

REQ 1.1.4 Bloquear un usuario en caso de que se ingresen claves erradas en tres ocasiones para el mismo.

REQ 1.1.5 Mostrar mensajes de alerta en caso de que se presenten eventos (Usuario o contraseña incorrectas), sin indicar qué campo es el que contienen información mal ingresada, esto con el fin no exponer la información de los usuarios.

8.11.1.1 Casos de Uso

CU001 Acceso al sistema

AUTOR	Harold Flórez Pérez – Dorian Ardila Puentes	REVISOR	Carlos Castro
CASO DE USO: CU001 – Acceso al sistema			
Actores:	<ul style="list-style-type: none">• Sistema• Usuario		
Propósito:	Permitir a los usuarios creados el ingreso al sistema, garantizando el correcto logueo y la carga de las opciones correspondientes según el perfil asignado al usuario.		
Descripción:	El usuario podrá logrearse en el sistema al seleccionar la opción de inicio de sesión, y luego del logueo exitoso, el sistema le presentará las opciones correspondientes a su tipo de usuario. El logueo, se realizará mediante el ingreso de usuario y contraseña otorgado para cada usuario.		
Caso de Uso relacionado			
Precondiciones	<ul style="list-style-type: none">- El usuario deberá estar creado en la aplicación y debe tener perfil o tipo de usuario asignado.- El usuario deberá contar con un usuario y contraseña para ingresar a la aplicación.		
Requisitos Relacionados	REQ 1.1.1, REQ 1.1.2, REQ 1.1.3, REQ 1.1.4, REQ 1.1.5		
Escenarios			

<p>Flujo Básico</p>	<ol style="list-style-type: none"> 1. El usuario selecciona la opción Iniciar Sesión en la pantalla principal de la aplicación. 2. El sistema presenta en pantalla los campos usuario y contraseña junto con la opción <i>Iniciar Sesión</i>. 3. El usuario ingresa los datos solicitados en pantalla y selecciona la opción <i>Iniciar Sesión</i>. 4. El sistema efectúa las validaciones pertinentes a completitud y consistencia de datos, y valida la cantidad de veces que se han intentado loguear con el usuario ingresado. 5. El sistema encuentra que la cantidad de veces que se han intentado loguear con el usuario no supera tres veces. 6. El sistema encuentra que los datos ingresados son correctos, y que el tipo de usuario que está tratando de loguearse es Administrador. 7. El sistema muestra la pantalla principal compuesta por un menú en la parte izquierda, con las opciones Financiación y Gastos, Investigación, Entidades Territoriales, Cerrar Sesión, en el centro de la pantalla dividido en varias secciones, se encuentran los enlaces directos de la siguiente forma: Proyecto de Investigación: les, NBC, Objetivo Socioeconomico, Tipo Proyecto Investigación, Proyecto de Investigación, Redes de Cooperación, Grupos de Investigación, Centros de Investigación, Productos de Investigación, Tipo Productos de Investigación, Entidades Integrantes de Red, Otras Actividades, Reportes de Sistemas. Investigadores: Investigadores, Nivel Educativo, Tipo Participación Proyecto. Fuentes de Financiación y Tipos de Gastos: Fuente de Financiación Nacional, Fuentes de Financiación Internacional y Tipos de Gastos. Entidades Territoriales: País, Departamentos, Municipios, Sector Entidad. Administración de Usuarios: Administrar Usuarios, Administrar Permisos, Cambiar Mis Datos de usuario, Roles de Usuarios; y adicionalmente las opciones Inicio y Cerrar Sesión. Fin de flujo.
<p>Flujo Alternativo 5</p>	<ol style="list-style-type: none"> 1. El sistema encuentra que la cantidad de veces que se han intentado loguear con el usuario supera tres veces, bloquea el acceso para dicho usuario y muestra en pantalla el siguiente mensaje "Ha superado el número de intentos fallidos permitidos para un usuario, por favor comuníquese con el administrador del sistema para realizar el proceso de desbloqueo del usuario". Fin de flujo.

Flujo Alterno 6	<p>a.</p> <ol style="list-style-type: none"> 1. El sistema encuentra que los datos no son correctos, y muestra un mensaje en color rojo indicando "Usuario/Clave incorrectos". El flujo continúa en el paso 3 del flujo principal. <p>b.</p> <ol style="list-style-type: none"> 1. El sistema encuentra que los datos ingresados son correctos y que el tipo de usuario que está tratando de loguearse es Docente. 2. El sistema muestra en pantalla en el menú las opciones "Investigación" y "Cerrar Sesión". Fin de Flujo.
------------------------	--

8.11.2 Administrar de Permisos CU002 CU 002

AUTOR	Harold Flórez Pérez – Dorian Ardila	REVISOR	Carlos Castro
CASO DE USO: CU002 – Administrar Permisos			
Actores:	<ul style="list-style-type: none"> • Sistema • Administrador 		
Propósito:	Permitir al usuario Administrador del sistema administrar los permisos por perfil creado en el sistema.		
Descripción:	El administrador podrá marcar o desmarcar opciones del sistema para permitir o denegar el acceso a los perfiles seleccionados anteriormente.		
Caso de Uso relacionado			
Precondiciones	<ul style="list-style-type: none"> - El usuario deberá estar logueado en la aplicación. - El usuario deberá estar creado en la aplicación y debe tener perfil o tipo de usuario asignado. - El perfil debe estar creado en el sistema. 		
Requisitos Relacionados			
Escenarios			

<p>Flujo Básico</p>	<ol style="list-style-type: none"> 1. El usuario selecciona la opción Administrar Permisos del menú principal de la aplicación, además de las opciones Inicio, “Opciones”+Nombre de la opción y Cerrar Sesión. 2. El sistema presenta una lista desplegable con todos los perfiles creados en el sistema, junto con la opción <i>Consultar</i>. 3. El usuario selecciona algún perfil de los disponibles en la lista desplegable y hace clic en la opción Consultar. 4. El sistema presenta en pantalla de forma paginada el listado de las funcionalidades del sistema, y al frente de cada uno de los registros la opción Permiso (Y Si – N No) 5. El usuario selecciona la opción (Y Si – N No) de alguno de los registros, 6. El sistema almacena el permiso otorgado en la base de datos. Fin de Flujo.
<p>Flujo Alternativo 2</p>	<ol style="list-style-type: none"> a. <ol style="list-style-type: none"> 1. EL usuario selecciona la opción Inicio 2. El sistema muestra la pantalla inicial de la aplicación. Fin de Flujo. b. <ol style="list-style-type: none"> 1. El usuario selecciona la opción Opciones + Nombre de la opción. 2. El sistema muestra la opción Agregar, Listar todas, Cambiar Permisos de usuario, Modificara, Eliminar. Fin de Flujo. c. <ol style="list-style-type: none"> 1. El usuario selecciona la opción Cerrar Sesión. 2. El sistema presenta la pantalla del loguin de la aplicación (ver CU Acceso al sistema)

8.11.3 Administración de Maestros

8.11.3.1 Requisitos Funcionales Administración de Maestros

REQ 1.1.6 El sistema debe validar en el momento de una eliminación de registros de un maestro, si dicho maestro tiene más de dos registros, es decir, para la eliminación de valores, los maestros no podrán tener menos de dos valores.

REQ 1.1.7 El sistema debe permitir crear registros de las listas desplegables que se encontrarán en los diferentes formularios del sistema.

REQ 1.1.8 El sistema debe permitir consultar registros de las listas desplegables que se encontrarán en los diferentes formularios del sistema.

REQ 1.1.9 El sistema debe permitir editar registros de las listas desplegables que se encontrarán en los diferentes formularios del sistema.

REQ 1.1.10 El sistema debe permitir eliminar registros de las listas desplegables que se encontrarán en los diferentes formularios del sistema.

REQ 1.1.11 El sistema solo deberá permitir eliminar valores de las listas desplegables, pero dichas listas desplegables (campos) no se deberán eliminar del sistema.

REQ 1.1.12 El sistema no debe permitir eliminar un maestro, pues esta eliminación corresponde solamente a los valores que componen cada uno de ellos.

8.11.3.2 Casos de Uso Administrar maestros

AUTOR	Harold Flórez Pérez – Dorian Ardila	REVISOR	Carlos Castro
CASO DE USO: CU003 – Administrar Maestros			
Actores:	<ul style="list-style-type: none"> • Sistema • Administrador 		
Propósito:	Permitir al usuario Administrador del sistema administrar los maestros (valores de las lista desplegables) que estarán disponibles en los diferentes formularios de la aplicación.		
Descripción:	El administrador podrá insertar, modificar, eliminar y buscar valores de las listas desplegables que se presentan en los diferentes formularios del sistema. Se permite listar los registros almacenados anteriormente por maestro para posteriormente realizar las modificaciones de un elemento seleccionado por medio de doble clic.		
Caso de Uso relacionado			
Precondiciones	<ul style="list-style-type: none"> - El usuario deberá estar logueado en la aplicación. - El usuario deberá estar creado en la aplicación y debe tener perfil o tipo de usuario asignado. 		
Requisitos Relacionados	REQ 1.1.6, REQ 1.1.7, REQ 1.1.8, REQ 1.1.9, REQ 1.1.10, REQ 1.1.11, REQ 1.1.12		
Escenarios			

Flujo Básico	<ol style="list-style-type: none">7. El usuario selecciona algún maestro de los disponibles en el menú presentado de la página principal.8. El sistema presenta en pantalla las opciones <i>Agregar, Listar Todos, Modificar y Eliminar</i>.9. El usuario selecciona la opción <i>Modificar</i>.10. El sistema lista todos los registros del maestro y muestra en la parte superior de la pantalla el mensaje que dice “Doble Click en la Fila para modificar”.11. El usuario selecciona por medio de doble clic un registro.12. El sistema muestra los campos correspondientes al maestro y la opción <i>Modificar Datos</i>.13. El usuario modifica los datos y selecciona la opción <i>Modificar Datos o Guardar</i>.14. El sistema efectúa las validaciones pertinentes a completitud y consistencia de datos.15. El sistema almacena los datos ingresados o modificados en la base de datos de la aplicación y presenta al usuario el resultado de la operación en pantalla “Se ha modificado el registro con éxito”. Fin de flujo.
---------------------	---

<p>Flujo Alterno 3</p>	<p>a.</p> <ol style="list-style-type: none"> 1. El usuario selecciona la opción <i>Agregar</i>. 2. El sistema presenta los campos del maestro en blanco para ser diligenciados junto con la opción Guardar. El flujo continúa en el paso 7 del flujo básico. <p>b.</p> <ol style="list-style-type: none"> 1. El usuario selecciona la opción <i>Listar Todos</i>. 2. El sistema muestra de manera paginada todos los registros correspondientes al maestro seleccionado, además de mostrar debajo del listado las opciones Inicio, Opciones y Cerrar sesión. 3. El usuario selecciona la opción "<i>Opciones</i>" + nombre del maestro. El flujo continúa en el paso 2 del flujo básico. <p>c.</p> <ol style="list-style-type: none"> 1. El usuario selecciona la opción <i>Eliminar</i>. 2. El sistema muestra de manera paginada todos los registros correspondientes al maestro seleccionado. 3. El usuario selecciona por medio de doble clic un registro 4. El sistema muestra los campos del registro y la opción Eliminar. 5. El usuario selecciona la opción Eliminar. 6. El sistema valida y encuentra que la lista desplegable tiene más de dos registros. 7. El sistema pide confirmación para eliminar el valor de la lista desplegable, presentándole el siguiente mensaje: "<i>Esta seguro que desea Eliminar el registro de la aplicación?</i>" 8. El usuario confirma la eliminación del registro. 9. El sistema realiza la eliminación del usuario en la base de datos de la aplicación. El flujo continúa en el paso 2 del flujo básico.
<p>Flujo alternativo 3.c.2</p>	<ol style="list-style-type: none"> 1. El sistema encuentra que la lista desplegable tiene dos registros, y muestra un mensaje indicando que no es posible eliminar estos registros de la aplicación. El flujo continúa en el paso 2 del flujo básico.
<p>Flujo Alterno 9</p>	<ol style="list-style-type: none"> 1. El sistema encuentra que los datos no están ingresados correctamente, le informa al usuario que los ingrese nuevamente, y muestra al paso del mouse por cada uno de los campos, una sugerencia para ingreso de información. El Flujo continúa en el paso 7 del Flujo básico.

8.11.4 Generación de Reporte

8.11.4.1 Requisitos Funcionales Generación de Reporte

REQ 1.1.13 El sistema debe permitir generar el reporte de proyectos de investigación, con la información suministrada a través de los módulos anteriores en formato Excel (.xls).

8.11.4.2 Casos de Uso

CU003 Generar Reporte

AUTOR	Harold Flórez Pérez – Dorian Ardila Puentes	REVISOR	Carlos Castro
CASO DE USO: CU011 – Generar Reporte			
Actores:	<ul style="list-style-type: none">• Sistema• Administrador		
Propósito:	Permitir al usuario Administrador del sistema generar el reporte de los proyectos de investigación de la Universidad de San Buenaventura.		
Descripción:	El administrador podrá generar el reporte con los campos que detallados en el requisito de información INF018 .		
Caso de Uso relacionado			
Precondiciones	- El usuario deberá estar creado en la aplicación y debe tener perfil o tipo de usuario asignado.		
Requisitos Relacionados	REQ 1.1.43		
Escenarios			
Flujo Básico	<ol style="list-style-type: none">1. El usuario selecciona del menú presentado en la página principal la opción Reportes.2. El sistema presenta en pantalla las opciones <i>Generar Reporte</i> y <i>Salir</i>.3. El usuario selecciona la opción <i>Generar Reporte</i>.4. El sistema abre una ventana donde indica las opciones <i>Abrir</i>, <i>Guardar</i>, <i>Cancelar</i>.5. El usuario selecciona la opción <i>Abrir</i>.6. El sistema muestra en una ventana de Excel el reporte generado con los campos citados en el requisito de información INF018, junto con las opciones propias de Excel. Fin del Flujo.		

Flujo Alterno 3	a. 1. El usuario selecciona la opción <i>Salir</i> . 2. El sistema presenta la pantalla del menú principal. Fin del flujo.
Flujo Alterno 5	a. 1. El usuario selecciona la opción <i>Guardar</i> . 2. El sistema abre la ventana del explorador con las posibilidades de rutas de almacenamiento del equipo y las opciones propias del navegador. Fin del flujo. b. 1. El usuario selecciona la opción <i>Cancelar</i> 2. El sistema cierra la ventana y muestra nuevamente la pantalla de la generación de reportes. El flujo continúa en el paso 2 del flujo básico.

8.12 PROTOTIPOS DESARROLLADOS

El aplicativo se desarrolló mediante 3 versiones de prototipos.

La primera versión fue realizada en visual studio .net utilizando entity framework para el modelo de base de datos, ésta versión permitió aclarar los requisitos previos con los usuarios tanto funcionales como no funcionales, a partir de la cual se sugirió utilizar otra plataforma. Esta versión contenía un modelo de acceso a datos y una interfaz web para administración de maestros. Ver anexo (1)

La versión número dos fue realizada en PHP con apoyo del framework Kumbia. Esta versión contenía un modelo de acceso a datos y una interfaz web para administración de maestros.

La versión número tres se cohesionó con el manejo de acceso a base de datos y manejo de maestros, incluye el modelo de negocio y las clases distribuidas por capas anteriormente mencionadas en el modelo de arquitectura diseñada para la solución.

En las figuras de la 21 a la 28 se presenta una interfaz gráfica del modelo del prototipo final.

Figura 21. Módulo de Acceso



Figura 22. Interfaz Página Principal

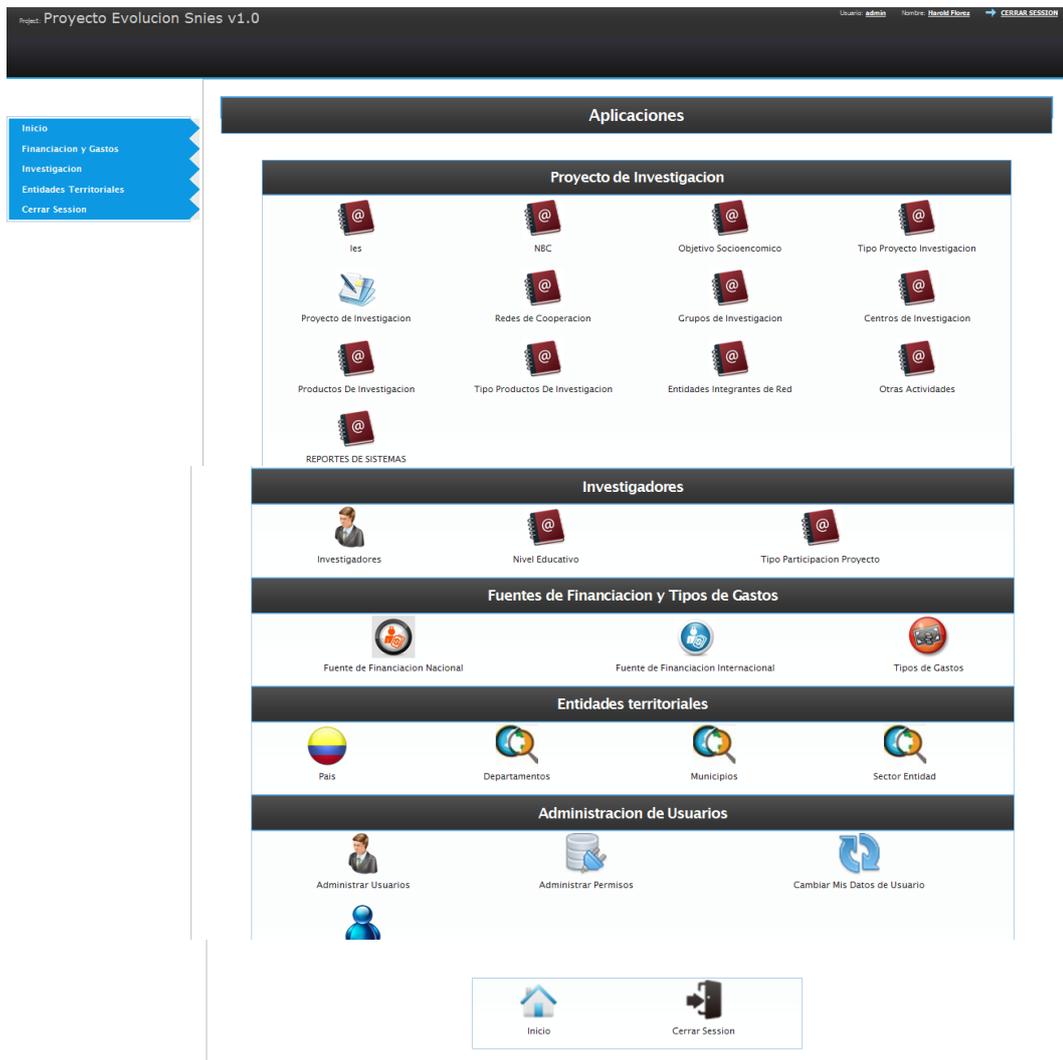


Figura 23. IES

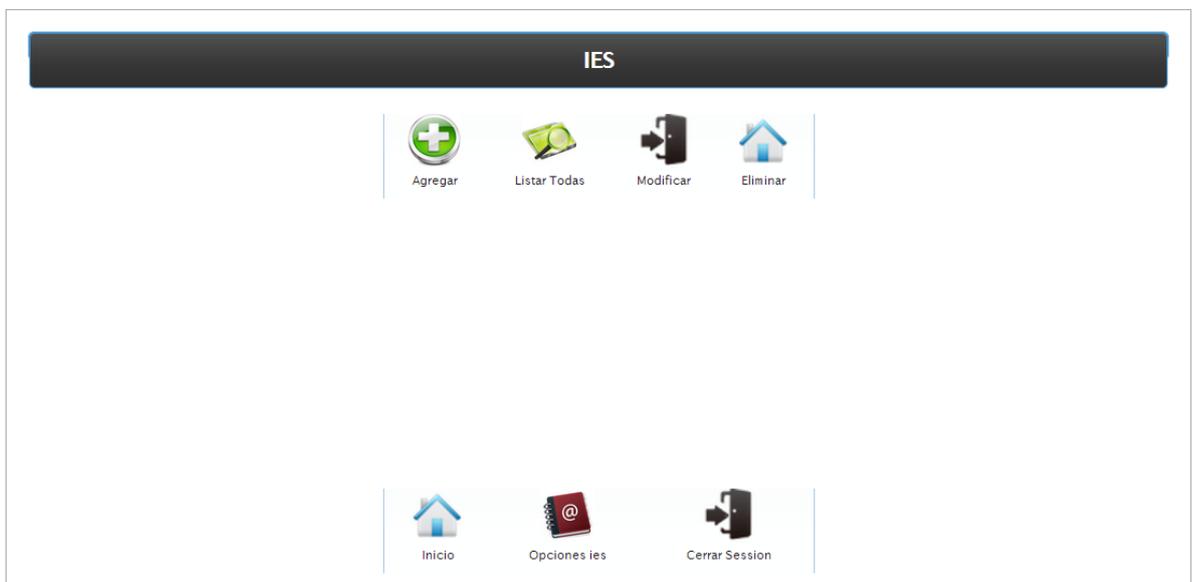


Figura 24. Agregar IES

The screenshot shows a web interface for adding a new IES. At the top, there is a dark blue header with the text 'IES'. Below it is a sub-header 'AGREGAR IES' and the title 'FORMULARIO PARA AGREGAR IES'. The form contains two input fields: 'CODIGO IES:' and 'NOMBRE IES:'. A blue 'Guardar' button with a dropdown arrow is located at the bottom right of the form. Below the form, there is a navigation bar with three icons: a house icon labeled 'Inicio', a red notebook icon labeled 'Opciones ies', and a black door icon labeled 'Cerrar Sesion'.

Figura 25. Validación IES

This screenshot shows the same 'Agregar IES' form as in Figure 24, but with validation error messages displayed below the input fields. The messages are: 'Error: No se pudo Guardar el registro...', 'El campo Cod_IES no puede ser nulo', and 'El campo Nombre_IES no puede ser nulo'. The navigation bar at the bottom remains the same.

Figura 26. Listado de las IES Registradas

Estado de las IES Registradas

Codigo ies	Nombre ies
1001	Unviersidad de San Buenaventura Medellin

Página 1 de 1 Mostrando 1 - 1 de 1



Inicio



Opciones ies



Cerrar Session

Figura 27. Modificar Listado de IES Registradas

Doble Click en la Fila para modificar

Estado de las IES Registradas

Codigo ies	Nombre ies
1001	Unviersidad de San Buenaventura Medellin

Página 1 de 1 Mostrando 1 - 1 de 1



Inicio

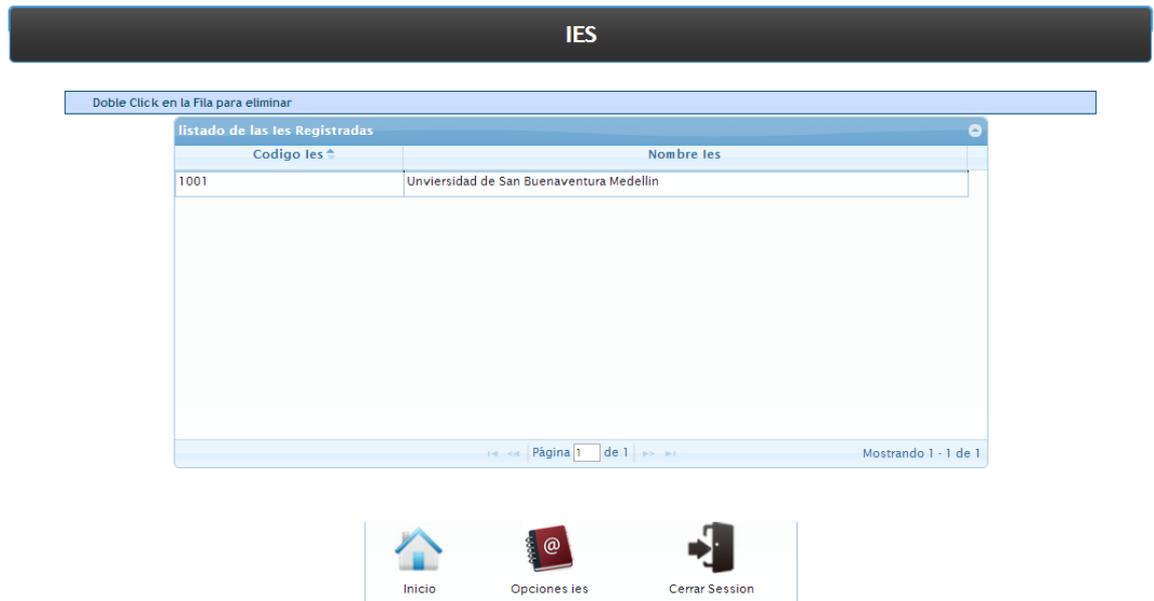


Opciones ies



Cerrar Session

Figura 28. Eliminación Listado IES Registradas



8.13 Pruebas Funcionales

En las pruebas funcionales se realizaron los diseños de aplicación con el objetivo de evaluar el cumplimiento de los requisitos levantados en la etapa de análisis de información, así como la aceptación del usuario final la ingeniera Helena Perez.

En el anexo (1) se detalla el criterio de aceptación para determinar si se cumplen los requisitos según el usuario final. Además se anexa los casos de prueba junto con la evidencia.

8.14 Manual de usuario y manual de instalación

En el anexo numero (2) se detalla el manual de usuario incluyendo el manejo del framework y la base de datos.

En el manual de usuario se detalla el manejo de cada uno de los módulos. Acceso, negocio, modulo maestros y reportes.

9. REFERENCIAS BIBLIOGRÁFICAS

Engineers, S. S. (2013). Proceso y Roles de Scrum. Recuperado el Junio de 2013, de <http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html>

Innova, W. (2013). *Web innova*. Obtenido de Metodologia del sitio Web: <http://www.webinnova.com.co/desarrollo-web-aplicaciones-a-la-medida-php-jsp-mysql-oracle-colombia/metodologia-de-desarrollo.html>

Sutherland, K. S. (10 de 2011). *Guia Scrum*. Obtenido de La Guia de Scrum: <http://www.scrum.org/Scrum-Guides>

Díaz, J. R. (29 de 12 de 2008). *Cómo gestionar proyectos con Scrum*. Obtenido de Técnicas ágiles y CMMI en un proyecto de Banca: <http://www.proyectosagiles.org/tecnicas-agiles-cmmi-2-proyecto-banca>

PSL, W. (2 de 02 de 2012). *PSL*. Obtenido de PSL en las noticias: <http://www.psl.com.co/noticias/psl-en-las-noticias.html>

Medina, L. E. (04 de 2011). *Aplicación de SCRUM para minimizar los riesgos en un proyecto de desarrollo*. Obtenido de EPN.EDU.CO: <http://bibdigital.epn.edu.ec/bitstream/15000/3760/1/CD-3537.pdf>

Adelaida Valbín Landeo, I. V. (2010). Scrum con UML para programadores.

Ambler, S. W. (19 de 08 de 2002). *Agile Modeling*. Recuperado el 04 de 2013, de ¿ Qué son los modelos ágiles?:

<http://www.agilemodeling.com/shared/AMPamphletSpanish.pdf>

Araújo, A. (07 de 2004). *Metodologías Ágiles*. Obtenido de ¿ Testing ágil?: <http://www.slideshare.net/alar758/metologas-giles-testing-gil>

Wikipedia. (10 de 11 de 2012). *Ingeniería de Software*. Recuperado el 04 de 2013, de http://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software

10. GLOSARIO

ADMINISTRADOR DE SISTEMA: es aquella persona que se dedica a mantener y operar un sistema de cómputo o una red.

BASES DE DATOS: es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

CASO DE USO: es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software.

DATO: información en un formato que pueda ser procesado por un ordenador. La información se condensa digitalmente, de modo que un texto, imagen o sonido se pueda representar en la pantalla.

DIAGRAMA DE CLASE: es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

DIAGRAMA DE SECUENCIA: es un tipo de diagrama usado para modelar interacción entre objetos en un sistema según UML.

HARDWARE: componentes físicos y equipos periféricos de una computadora, ("los fierros"). Es toda componente física involucrada en el funcionamiento de equipo informático.

INFRAESTRUCTURA: conjunto de las redes, datos y equipamientos de las telecomunicaciones.

METRICA: es cualquier medida o conjunto de medidas destinadas a conocer o estimar el tamaño u otra característica de un software o un sistema de información, generalmente para realizar comparativas o para la planificación de proyectos de desarrollo.

ROUTER: es un dispositivo para la interconexión de redes informáticas que permite asegurar el enrutamiento de paquetes entre redes o determinar la ruta que debe tomar el paquete de datos.

RACK: es un bastidor destinado a alojar equipamiento electrónico, informático y de comunicaciones.

RED: es el conjunto de ordenadores y dispositivos conectados entre sí por medio de cables y tarjetas especiales.

SERVIDOR: es una computadora que, formando parte de una red, provee servicios a otras computadoras denominadas clientes.

SOFTWARE: programas, procedimientos y reglas para la ejecución de tareas específicas en un sistema de cómputo.

SIMULACION: modelar el comportamiento dinámico de un sistema.

USUARIO: Cualquiera que requiere los servicios de un sistema de cómputo.

SNIES: (Sistema Nacional de Información de la Educación Superior) es un sistema de información que ha sido creado para responder a las necesidades de información de la educación superior en Colombia.

11.CONCLUSIONES

- Es importante validar las necesidades específicas y a detalle del usuario final, esto con el fin de no incurrir posteriormente en reprocesos por cambios no controlados.
- El éxito de un proyecto enmarcado en una metodología ágil radica en el seguimiento periódico a las actividades definidas en cada una de las iteraciones
- A través de la utilización de metodologías ágiles como Scrum para el desarrollo de proyectos, se hace más fácil el control de los riesgos presentes en los mismos
- A través de la sistematización de la información, es posible tener de manera centralizada la gestión de la misma, se optimizan procesos y se asegura la estandarización.