

**SISTEMA DE INFORMACIÓN WEB PARA EL REPOSITORIO DE LOS DOCUMENTOS
GENERADOS EN EL PROCESO DE AUTOEVALUACIÓN Y EL PLAN DE MEJORAMIENTO DE
LOS PROGRAMAS**

**DIEGO FERNANDO CAICEDO MOSQUERA
CÓDIGO NO. 160002004**

**GERMAN AUGUSTO CESPEDES YELA
CÓDIGO NO. 160002108**

**UNIVERSIDAD DE LOS LLANOS
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
VILLAVICENCIO
2016**

**SISTEMA DE INFORMACIÓN WEB PARA EL REPOSITORIO DE LOS DOCUMENTOS
GENERADOS EN EL PROCESO DE AUTOEVALUACIÓN Y EL PLAN DE MEJORAMIENTO DE
LOS PROGRAMAS**

**DIEGO FERNANDO CAICEDO MOSQUERA
CÓDIGO NO. 160002004**

**GERMAN AUGUSTO CESPEDES YELA
CÓDIGO NO. 160002108**

TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE INGENIERO DE SISTEMAS

**DIRECTOR(A)
MÓNICA SILVA QUINCENO
INGENIERA ELECTRÓNICA**

**UNIVERSIDAD DE LOS LLANOS
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
VILLAVICENCIO
2016**

Notas de aceptación:

Firma del Jurado

Firma del Jurado

Villavicencio, 31 de marzo de 2016

DEDICATORIA

A todo aquel ser que ha influenciado en mí y ha hecho que la razón de existir siempre este en constante cambio, que ha logrado que tenga la motivación e inspiración para sentarme a escribir estas últimas líneas al culminar este proyecto que tanto aplace por no haberle encontrado un sentido, pero que aún con esa objeción pudimos vencer el aburrimiento y el desespero de hacer lo que menos se gusta y darle un giro a la vida en los pensamientos para encontrar nuevas expectativas.

Diego Fernando Caicedo

Este logro se lo dedico a mi amada familia, padre, madre y hermano que me han brindado ese apoyo incondicional a diario, para la mujer que siempre ha estado conmigo en mis triunfos y en mis derrotas, colocando ese valor sentimental tan fuerte e importante en mi vida, a mis amigos y compañeros que me brindaron su apoyo moral y compañerismo, para que, así, pudiera sobrepasar cada obstáculo que se ha presentado en mi carrera profesional, muchas gracias.

GERMÁN AUGUSTO CÉSPEDES YELA

AGRADECIMIENTOS

A los mencionados anteriormente (sin nombre).

Diego Fernando Caicedo

A los mencionados anteriormente.

Germán Augusto Céspedes Yela

CONTENIDO

INTRODUCCIÓN.....	12
1. DESCRIPCIÓN DEL PROBLEMA.....	13
2. OBJETIVOS.....	14
2.1 OBJETIVO GENERAL.....	14
2.2 OBJETIVOS ESPECÍFICOS.....	14
3. JUSTIFICACIÓN.....	15
4. CONTEXTO.....	16
5. MARCO TEORICO.....	17
5.1 INGENIERÍA DE SOFTWARE.....	17
5.2 SCRUM.....	17
5.3 PROGRAMACIÓN EXTREMA XP.....	18
5.4 UML.....	18
5.5 START UML.....	18
5.6 SISTEMAS DE INFORMACIÓN.....	19
5.7 BASES DE DATOS.....	19
5.8 POSTGRESQL.....	19
5.9 SERVIDOR WEB.....	20
5.10 HTML5.....	20
5.11 HOJA DE ESTILOS EN CASCADA (CSS).....	20
5.12 MATERIALIZE.....	21
5.13 MATERIAL DESIGN.....	21
5.14 JAVASCRIPT.....	21
5.15 JQUERY.....	21
5.16 AJAX.....	22
5.17 JSON.....	22
5.18 PYTHON.....	23
5.19 DJANGO.....	23
6. METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	24
7. DESARROLLO DEL PROYECTO.....	25
7.1 LISTA DE HISTORIAS DE USUARIO.....	25
7.1.1 Usuarios de la aplicación.....	26

7.2	DIAGRAMA DE CASOS DE USO.....	27
7.3	MOCKUPS.....	28
7.3.1	Inicio de sesión.....	28
7.3.2	Proyectos de acreditación	29
7.3.3	Factores	29
7.3.4	Factor	30
7.3.5	Evaluar factores	30
7.3.6	Equipo de trabajo	31
7.3.7	Gestionar tareas.....	31
7.3.8	Tareas programadas	32
7.3.9	Historial de acciones	32
7.3.10	Registrar usuario	33
7.4	DIAGRAMA DE CLASES.....	34
7.5	MODELO ENTIDAD RELACIÓN	35
7.6	SPRINT BACKLOG	36
7.6.1	Iteración 1	36
7.6.2	Iteración 2	37
7.6.3	Iteración 3	38
7.6.4	Iteración 4	39
8.	RESULTADOS.....	40
9.	CONCLUSIONES.....	41
10	REFERENCIAS BIBLIOGRÁFICAS	43
	BIBLIOGRAFÍA.....	44

LISTA DE TABLAS

Tabla 1 Factores Lineamientos CNA.....	13
Tabla 2 Lista de historias de usuario.....	25
Tabla 3 Iteración 1 – Crear proyecto inicial.....	36
Tabla 4 Iteración 1 – Diseñar web.....	36
Tabla 5 Iteración 1 – Crear plantillas HTML.....	36
Tabla 6 Iteración 1 – Configurar el manejador de URL's.....	36
Tabla 7 Iteración 1 – Configurar el modulo de contactos.....	36
Tabla 8 Iteración 1 – Iniciar sesión.....	37
Tabla 9 Iteración 1 – Diseñar modelo para proyectos.....	37
Tabla 10 Iteración 1 – Crear plantillas para proyectos.....	37
Tabla 11 Iteración 1 – Crear proyectos.....	37
Tabla 12 Iteración 1 – Eliminar proyectos.....	37
Tabla 13 Iteración 2 – Crear directorios.....	37
Tabla 14 Iteración 2 – Descripción tarea.....	37
Tabla 15 Iteración 2 – Subir archivos.....	38
Tabla 16 Iteración 3 – Eliminar archivos.....	38
Tabla 17 Iteración 3 – Crear modelo para la evaluación de factores.....	38
Tabla 18 Iteración 3 – Hacer el formulario para la evaluación.....	38
Tabla 19 Iteración 3 – Proceso de evaluación de un proyecto.....	38
Tabla 20 Iteración 4 – Modelo equipo de trabajo.....	39
Tabla 21 Iteración 4 – Codificar formulario y plantilla de la tarea.....	39
Tabla 22 Iteración 4 – Ingresar equipo.....	39
Tabla 23 Iteración 4 – Modelo de tareas.....	39
Tabla 24 Iteración 4 – Formulario y plantilla de tareas.....	39
Tabla 25 Iteración 4 – Ingresar tareas.....	39

LISTA DE FIGURAS

Figura 1 Comunicación entre servidor y cliente	20
Figura 2 Metodología de desarrollo	24
Figura 3 Arquitectura de servicios.....	26
Figura 4 Diagrama de casos de uso	27
Figura 5 Inicio de sesión	28
Figura 6 Proyectos de acreditación.....	29
Figura 7 Factores.....	29
Figura 8 Factor.....	30
Figura 9 Evaluar factores.....	30
Figura 10 Equipo de Trabajo.....	31
Figura 11 Gestionar Tareas.....	31
Figura 12 Tareas Programadas.....	32
Figura 13 Historial de Acciones.....	32
Figura 14 Historial de Acciones.....	33
Figura 15 Diagrama de Clases	34
Figura 16 Modelo entidad relación.....	35

RESUMEN

Los programas de pregrado de las instituciones de educación superior afrontan procesos necesarios para adquirir la acreditación de alta calidad, un reconocimiento por parte del Ministerio de Educación Nacional, que es otorgado a través del Consejo Nacional de Acreditación- CNA, con el fin de promover y reconocer la dinámica de mejoramiento continuo de los programas, para lograr este objetivo los programas llevan a cabo de manera periódica el proceso de autoevaluación, el cual tiene como base los lineamientos de acreditación de los programas de pregrado del CNA (2013), en los cuales se involucran factores, características, y aspectos a evaluar que deben ser tenidos en cuenta por los programas.

El proceso de autoevaluación involucra a toda la comunidad académica (estudiantes, profesores, directivos y administrativos) de los programas con el fin de generar un compendio de documentos descriptivos los cuales se basan en los lineamientos de acreditación que posteriormente servirán como insumo para dar soporte al proceso realizado; dicha información se debe analizar con el fin de que se logre generar el respectivo plan de mejoramiento que permita generar acciones para el fortalecimiento de los programas, las cuales se desarrollaran año a año a través del plan de acción del programa, permitiendo de esta manera mejorar los aspectos evaluados en los cuales se evidenciaron en el proceso de autoevaluación, en pro de conseguir la alta calidad en el programa; pero otra de las utilidades que se le dan a estos documentos es la de retroalimentar los procesos futuros para lograr obtener ventaja de la experiencia ya adquirida en los procesos realizados durante diferentes periodos.

Una labor importante para consolidar esta información plasmada en los diferentes documentos es conseguir que siempre estén a disposición estos en la comunidad académica, que se logre consolidar una arquitectura organizacional de todos los documentos ya sean para proyectos anteriores como futuros, además de tener una historia de los diferentes procesos de acreditación de los programas de las instituciones, información valiosa para la comunidad.

ABSTRACT

The undergraduate programs of higher education institutions facing processes needed to acquire high quality accreditation, a recognition by the Ministry of Education, which is awarded by the National Council of Acreditación- CNA, in order to promote and recognize the dynamics of continuous improvement of the programs, to achieve this objective, the programs perform periodically a self-assessment process, which is based on the guidelines of accreditation of undergraduate programs of CNA (2013), where factors, features, and aspects to be evaluated to be taken into account by the programs involved.

The self-assessment process involves the entire academic community (students, teachers, managers and administrators) of the programs with the aim of generating a compendium of descriptive documents which are based on accreditation guidelines that subsequently serve as an input to support the process conducted, such information must be analyzed in order to be able to generate the corresponding improvement plan that will generate improvement actions for strengthening programs, which develop year after year through the program action plan, thereby enabling to improve the aspects evaluated in which shortcomings were evident in the self-assessment process, towards achieving high quality in the program; but other utilities that are given to these documents is to provide feedback for future processes to achieve take advantage of the experience already gained in the processes performed during different periods.

An important task to consolidate this information reflected in the various documents is to always be available to those in the academic community, to achieve to consolidate organizational architecture of all documents whether to previous projects and future, besides having a history of the different processes of accreditation of programs of institutions wich is valuable information for the community.

INTRODUCCIÓN

La Facultad de Ciencias Básicas e Ingenierías FCBI de la Universidad de los Llanos mediante sus diferentes programas de pregrado han trabajado arduamente, con el objetivo de que el Ministerio de Educación a través del Consejo Nacional de Acreditación CNA les otorgue la acreditación de alta calidad a sus programas, para el año 2014 se ha culminado con éxito este proceso en el programa de Ingeniería de Sistemas y ha conseguido la acreditación de alta calidad por un periodo de cuatro años, el programa tiene la misión de trabajar por la renovación de la acreditación y por ende quiere conseguir organizar la documentación ya generada para que sirva como retroalimentación para procesos venideros y además de que la Facultad pueda replicar esta misma información a sus otros programas.

No se ha logrado encontrar evidencia pública de procesos sistematizados en acreditación y autoevaluación por parte de otras universidades con acreditación de alta calidad por esto se dispuso implementar una herramienta que permita almacenar y administrar los diferentes documentos generados en los procesos de acreditación, autoevaluación y plan de acción, buscando que toda la información relacionada con el tema esté a disposición oportunamente, para aquellos integrantes de la comunidad académica que así lo requieran y que cada uno de los procesos tenga una estructura de archivos estándar y organizada que formen una arquitectura de trabajo.

Para lograr conseguir implementar este sistema de información se utilizó una combinación de dos metodologías de desarrollo ágiles SCRUM y XP, la SCRUM hace énfasis a la filosofía a utilizar y la programación extrema se utilizó para las actividades que esta realiza, para el desarrollo de las diferentes actividades en cada sprint se tuvo muy en cuenta también el Framework web de tercera generación Django que utiliza Modelo Vista Plantilla MTV. También se utilizaron dos frameworks importantes del lado del cliente como lo son jQuery y Materialize basado en material Design.

1. DESCRIPCIÓN DEL PROBLEMA

El Programa de Ingeniería de Sistemas ha logrado después de un arduo trabajo de toda su comunidad académica (estudiantes, profesores, directivos y administrativos), que el Ministerio de Educación a través del Consejo Nacional de Acreditación – CNA –otorgó la Acreditación de Alta Calidad por un periodo de cuatro años, en este proceso se generó un conjunto de documentos descriptivos basados en los lineamientos de acreditación de programas de pregrado que el CNA brinda para dar orientación a los programas y desarrollar los correspondientes procesos; dichos factores de los lineamientos que fueron nuevamente propuestos en el año 2013 se pueden observar-en la Tabla 1.

Cuando el proceso de acreditación culminó con éxito obteniéndose la Acreditación de alta Calidad del mismo, el Programa de Ingeniería de Sistemas pensando en los próximos cuatro años, tiempo en el cual se vencerá esta acreditación y se deberá realizar nuevamente el proceso para lograr la renovación de esta acreditación evidenció que esta documentación desarrollada en el transcurso de los últimos tres o cuatro años por diferentes miembros de la comunidad académica del programa no se encuentra a disposición y de forma actualizada en un sitio central donde lo puedan consultar los docentes, administrativos, estudiantes y directivos interesados en conocer los procesos anteriores, esta falta de buen manejo de información le hace perder valor a los procesos de autoevaluación del programa, que han contribuido de forma benéfica en el logro de la acreditación, valioso para la Facultad de Ciencias Básicas e Ingenierías y sus programas.

Otra actividad que afrontan los programas de la Facultad de Ciencias Básicas e Ingenierías es la elaboración del Plan de Mejoramiento, el cual se realiza teniendo en cuenta las falencias encontradas en cada uno de los factores evaluados durante el proceso, este plan se hace contemplando un periodo de aproximadamente tres años (generalmente hasta que ocurre un nuevo proceso de autoevaluación), y es el punto de partida para la elaboración de los planes de acción anuales del programa, la importancia de tener presente este documento radica en que cada vez que se comienza con el proceso no se tienen a disposición los anteriores documentos que fueron hechos, esto produce inconformidad con los docentes por no poder reutilizar esta documentación, y tener presente el punto de partida de elaboración del nuevo plan.

FACTORES
Misión, proyecto institucional y de programa
Estudiante
Profesores
Procesos Académicos
Visibilidad Nacional e Internacional
Investigación, Innovación y Creación Artística y Cultural
Bienestar Institucional
Organización, Administración y Gestión
Impacto de los egresados en el medio
Recursos Físicos y Financieros
10 FACTORES
40 CARACTERÍSTICAS
243 ASPECTOS A EVALUA

Tabla 1 Factores Lineamientos CNA

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Implementar un Sistema de Información Web para almacenar y administrar los documentos generados en los diferentes planes de mejoramiento y los procesos de acreditación basados en los lineamientos para programas de pregrado del CNA del año 2013.

2.2 OBJETIVOS ESPECÍFICOS

- Analizar los procedimientos basados en los lineamientos del CNA del 2013 que se llevaron a cabo por el programa en el desarrollo del proceso de acreditación de alta calidad y los procedimientos del plan de mejoramiento.
- Diseñar un modelo de datos basado en los lineamientos y planes de mejoramiento que me permita administrar toda la documentación.
- Crear un historial de cada uno de los procesos de acreditación y los planes de mejoramiento.
- Centralizar la documentación para tener acceso a ella de manera oportuna y eficiente.

3. JUSTIFICACIÓN

Toda institución de educación superior después de adquirir su registro calificado en cada uno de los programas, trabaja con el fin de mejorar su calidad y obtener buenos resultados en los procesos de autoevaluación posteriores a la obtención del registro, y por qué no pensar en presentar los resultados de dicha autoevaluación al CNA con el fin de obtener la acreditación de alta calidad. Al iniciar estos procesos el equipo de trabajo del programa encargado de elaborar los respectivos informes de autoevaluación, evidencia que resulta tedioso hacerlos debido a la falta de conocimiento sobre cómo abordar el proceso, siendo este tan complejo, delicado e importante, razón por la cual se considera que independientemente de la respuesta obtenida por parte del CNA (programa Acreditado o no acreditado) se debe tener presente la forma como se llevó a cabo el proceso, la información recopilada, el informe final producto de las falencias encontradas al realizar la autoevaluación del programa.

Los diferentes programas de la Universidad de los Llanos realizan un proceso llamado plan de mejoramiento donde se consolidan las acciones necesarias para corregir las diferentes falencias encontradas que se generan como consecuencia de los diferentes procesos de autoevaluación y de los cuales no ha quedado un registro de estos planes.

Lo ideal sería tener un Sistema de Información Web donde se puedan almacenar todos los documentos actualizados y que estos puedan ser consultados de manera oportuna, lo que sería dado a conocer como sistema de información según la definición del autor James A. Senn sería: "Es el proceso de examinar la situación de una empresa con el propósito de mejorarla con métodos y procedimientos más adecuados"¹. Estos documentos deberán estar organizados bajo una serie de parámetros que serán analizados en el respectivo proceso de análisis los cuales podrán ser fecha de creación, creador, proceso, ítems de los lineamientos, entre otros. Organizar la documentación ayuda en la mejora continua porque me permite reutilizar el conocimiento y los documentos adquiridos en los antiguos procesos de autoevaluación.

¹ Senn, J. A. (1992). *Análisis y Diseño de Sistemas de Información*. México : McGraw Hill.

4. CONTEXTO

El proceso de acreditación de un programa de pregrado debe cumplir con los lineamientos que exige el CNA por el cual sería muy importante que fuese apoyado por alguna herramienta de software.

Se estuvo indagando en Internet en busca de un software para apoyar el proceso de acreditación universitaria y hemos encontrado uno que se llama **Software de Acreditación Universitaria**² hecho y comercializado por la empresa **Scripta Software Ltda.**, el cual apoya los procesos de Registro Calificado, Autoevaluación, Estándares de Calidad y Acreditación Universitaria de acuerdo a lo establecido en el Sistema Nacional de Acreditación. La empresa está ubicada en la ciudad de Bogotá ha hecho software para la gestión documental, el POT, negocios fiduciarios, servicio al cliente, entre otros productos, para empresas como Findeter, Citi Bank, Coca-Cola, Servientrega, Banco Santander, Revista Semana, entre otras empresas.

No se logró encontrar un software realizado por la comunidad académica que supla las necesidades de alguna institución de educación superior en marco de la acreditación universitaria.

También se consultaron software para la creación de repositorios y se logró encontrar un tema mucho más abordado a nivel mundial en las comunidades académicas y de software libre, algunos que encontramos son los siguientes: **DSpace**³ es un sistema de gestión de repositorios digitales, fue concebido para la creación y gestión de repositorios institucionales y para las necesidades del archivo digital, gestiona objetos electrónicos, almacenamiento digital y publicación, soporta gran variedad de datos: libros, tesis, fotos, filmes, entre otros. **Eprints**⁴ es un software de generación de repositorios genéricos desarrollado por la universidad de Southampton, está diseñado para crear repositorios basados en la web altamente configurables, se suele utilizar como archivo abierto de artículos de investigación, pero se puede archivar cualquier cosa que este en algún formato digital.

² Software, S. (s.f.). *Scripta Software*. Recuperado el 17 de 11 de 2014

³ DSpace. (s.f.). *DSpace*. Recuperado el 14 de 11 de 2014, de <http://www.dspace.org/>

⁴ Eprints. (s.f.). *Eprints*. Recuperado el 17 de 11 de 2014, de <http://www.eprints.org/software/>

5. MARCO TEORICO

5.1 INGENIERÍA DE SOFTWARE

Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, y el estudio de estos enfoques, es decir, la aplicación de la ingeniería al software. Integra matemáticas, ciencias de la computación y prácticas cuyos orígenes se encuentran en la ingeniería.

Una definición de la IEEE:

- La ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación, y mantenimiento del software⁵.

5.2 SCRUM⁶

Scrum es el nombre con el que se denomina a los marcos de desarrollo ágiles caracterizados por:

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos autoorganizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o de cascada.

SCRUM es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en Scrum son el *ScrumMaster*, que procura facilitar la aplicación de scrum y gestionar cambios, el *ProductOwner*, que representa a los *stakeholders* (interesados externos o internos), y el *Team* que ejecuta el desarrollo y demás elementos relacionados con el. Durante cada *sprint*, un periodo entre una y cuatro semanas (la magnitud es definida por el equipo y debe ser lo más corta posible), el equipo crea un incremento de software *potencialmente entregable* (utilizable). El conjunto de características que forma parte de cada sprint viene del *Product Backlog*, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar (PBI, Product Backlog Item). Los elementos del *Product Backlog* que forman parte del sprint se determinan durante la reunión de *Sprint Planning*. Durante esta reunión, el *Product Owner* identifica los elementos del *Product Backlog* que quiere ver completados y los hace del conocimiento del equipo.

⁵ IEEE Standard Glossary of Software Engineering Terminology, IEEE std 610.12-1990, 1990. ISBN 155937067X.

⁶ Agile Project Management with Scrum, Ken Schwaber, Microsoft Press, January 2004, 163pp, ISBN 0-7356-1993-X

5.3 PROGRAMACIÓN EXTREMA XP

Es una metodología de desarrollo de la ingeniería de software formulada por Kent Beck, autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change* (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de la XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

5.4 UML

UML, es un lenguaje visual orientado al modelado de sistemas, UML sirve para definir un problema que afecta la organización y plantear una solución de diseño, modelando los procesos involucrados, ayuda al mejoramiento del nivel de comunicación formal dentro de un proyecto, el cual permite representar visualmente las reglas de creación, estructura y comportamiento de un grupo relacionado de objetos y procesos.

Asimismo permite visualizar de manera eficiente la complejidad de un sistema o una organización en un reducido número de diagramas; es decir, abordar la complejidad con una documentación minimalista y para desarrollar procesos/productos con una mayor fiabilidad y calidad; siendo el impacto de las decisiones más visibles.

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado de Rational o RUP), pero no especifica en sí mismo qué metodología o proceso a usar.

5.5 START UML

StarUml es un proyecto de software libre que intenta reemplazar las herramientas de UML comerciales tales como Rational Rose. Se ha utilizado para desarrollar el diagrama de clases y los diagramas de casos de uso en la etapa de análisis del proyecto. Debido a su interfaz intuitiva, se trata de una herramienta fácil de manejar para el usuario⁷.

⁷ Start Uml, recuperado el 29 de agosto de 2015, <http://staruml.io/>

5.6 SISTEMAS DE INFORMACIÓN

Es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su uso posterior, generados para cubrir una necesidad o un objetivo⁸. Dichos elementos formarán parte de alguna de las siguientes categorías:

- Personas;
- Datos;
- Actividades o técnicas de trabajo;
- Recursos materiales en general (generalmente recursos informáticos y de comunicación, aunque no necesariamente).

Todos estos elementos interactúan para procesar los datos (incluidos los procesos manuales y automáticos) y dan lugar a información más elaborada, que se distribuye de la manera más adecuada posible en una determinada organización, en función de sus objetivos.

5.7 BASES DE DATOS⁹

Se le llama base de datos a los bancos de información que contienen datos relativos a diversas temáticas y categorizados de distinta manera, pero que comparten entre sí algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjunto.

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, y por ende se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos.

Existen programas denominados sistemas gestores de bases de datos, abreviado DBMS, que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos DBMS, así como su utilización y administración, se estudian dentro del ámbito de la informática.

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas; También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

5.8 POSTGRESQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por

⁸ Angell, I.O. and Smithson S. (1991) Information Systems Management: Opportunities and Risks

⁹ Ley de Protección de Datos, Consultado el 20 de noviembre de 2015, de <http://www.infoleg.gov.ar/infolegInternet/anexos/60000-64999/64790/norma.htm>

organizaciones comerciales. Dicha comunidad es denominada el PGDG (*PostgreSQL Global Development Group*)¹⁰.

5.9 SERVIDOR WEB

Se entiende como un ordenador en el que se ejecuta el protocolo TCP/IP, que escucha las peticiones de los usuarios y las atiende o las restringe según se dé el caso. Estas peticiones y respuestas se efectúan mediante un protocolo de intercambio llamado HTTP (HiperText Transfer Protocol) que se encuentra implícito dentro del TCP/IP.

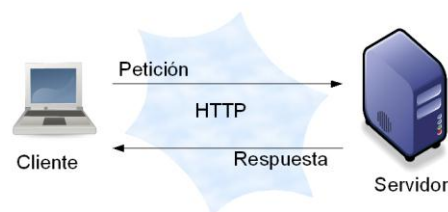


Figura 1 Comunicación entre servidor y cliente

Fuente: <http://www.wp-monkeys.com/php-comunicacion-entre-servidor-y-cliente.html>, consultado 20 de noviembre de 2015

5.10 HTML5

HTML5 es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: una «clásica», HTML (text/html), conocida como *HTML5*, y una variante XHTML conocida como sintaxis *XHTML5* que deberá servirse con sintaxis XML (application/xhtml+xml). Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo. La versión definitiva de la quinta revisión del estándar se publicó en octubre de 2014.

5.11 HOJA DE ESTILOS EN CASCADA (CSS)

Hoja de estilo en cascada o CSS (siglas en inglés de *cascading style sheets*) es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El World Wide Web Consortium (W3C) es el encargado de formular la especificación de las hojas de estilos que servirán de estándar para los agentes de usuario o navegadores. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

La información de estilo puede ser definida en un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo «style».

¹⁰ PostgreSQL Global Development Group. Consultado el 20 de noviembre de 2015, de <http://www.postgresql.org/about/licence/>

5.12 MATERIALIZE

Un framework web front-end moderno y responsivo basado en Material Design¹¹.

5.13 MATERIAL DESIGN

Creado y diseñado por Google, Material Design es un lenguaje de diseño que combina los principios clásicos del diseño exitoso junto con la innovación y la tecnología. El objetivo de Google consiste en desarrollar un sistema de diseño que permite una experiencia de usuario unificada a través de todos sus productos en cualquier plataforma.

5.14 JAVASCRIPT

JavaScript (abreviado comúnmente "JS") es un lenguaje de programación interpretado, dialecto del estándar ECMAScript¹². Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side Javascript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo. JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

5.15 JQUERY

jQuery es una biblioteca de Javascript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizada.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y Licencia Pública General GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

¹¹ Materialize, consultado el 20 de noviembre de 2015, de <http://materializecss.com/>

¹² ECMAScript, Language Specification, consultado en <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>

5.16 AJAX

AJAX, acrónimo de *Asynchronous JavaScript And XML* (Javascript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (*Rich Internet Applications*). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones¹³.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página, aunque existe la posibilidad de configurar las peticiones como síncronas de tal forma que la interactividad de la página se detiene hasta la espera de la respuesta por parte del servidor. Javascript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante *XMLHttpRequest*, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML. Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

5.17 JSON

JSON, acrónimo de *JavaScript Object Notation*, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de Javascript que no requiere el uso de XML. La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX. Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos en este contexto es que es mucho más sencillo escribir un analizador sintáctico (parser) de JSON. En JavaScript, un texto JSON se puede analizar fácilmente usando la función `eval()`, lo cual ha sido fundamental para que JSON haya sido aceptado por parte de la comunidad de desarrolladores AJAX, debido a la ubicuidad de JavaScript en casi cualquier navegador web. En la práctica, los argumentos a favor de la facilidad de desarrollo de analizadores o del rendimiento de los mismos son poco relevantes, debido a las cuestiones de seguridad que plantea el uso de `eval()` y el auge del procesamiento nativo de XML incorporado en los navegadores modernos. Por esa razón, JSON se emplea habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia (de aquí su uso por Yahoo, Google, etc, que atienden a millones de usuarios) cuando la fuente de datos es explícitamente de fiar y donde no es importante el no disponer de procesamiento XSLT para manipular los datos en el cliente.

¹³ *Ajax: A New Approach to Web Applications*, consultado en <http://adaptivepath.org/ideas/ajax-new-approach-web-applications/>

5.18 PYTHON

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible¹⁴. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma. Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License, que es compatible con la Licencia Pública General GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores.

5.19 DJANGO

Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como Modelo–vista–controlador. Fue desarrollado en origen para gestionar varias páginas orientadas a noticias de la World Company de Lawrence, Kansas, y fue liberada al público bajo una licencia BSD en julio de 2005; el framework fue nombrado en alusión al guitarrista de jazz gitano Django Reinhardt.

En junio del 2008 fue anunciado que la recién formada Django Software Foundation se haría cargo de Django en el futuro.

La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio No te repitas (DRY, del inglés *Don't Repeat Yourself*). Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos.

Los orígenes de Django en la administración de páginas de noticias son evidentes en su diseño, ya que proporciona una serie de características que facilitan el desarrollo rápido de páginas orientadas a contenidos. Por ejemplo, en lugar de requerir que los desarrolladores escriban controladores y vistas para las áreas de administración de la página, Django proporciona una aplicación incorporada para administrar los contenidos, que puede incluirse como parte de cualquier página hecha con Django y que puede administrar varias páginas hechas con Django a partir de una misma instalación; la aplicación administrativa permite la creación, actualización y eliminación de objetos de contenido, llevando un registro de todas las acciones realizadas sobre cada uno, y proporciona una interfaz para administrar los usuarios y los grupos de usuarios (incluyendo una asignación detallada de permisos). O3DIWT

¹⁴ The Making of Python, consultado el 20 de noviembre de 2015 <http://www.artima.com/intv/pythonP.html>

6. METODOLOGÍA DE DESARROLLO DE SOFTWARE

Se utilizó para el desarrollo de este proyecto metodologías ágiles, como lo son Scrum y Programación Extrema. Se utilizó Scrum ya que ofrece un conjunto de buenas prácticas para trabajar colaborativamente en equipo y una forma exitosa de realizar incrementos funcionales de negocio llamados Sprint, pero esta metodología carece de una definición de etapas para conseguir el producto de software por eso se complementa con la Programación Extrema que define ciertas etapas y características, pero se aclara que su filosofía de trabajo y la programación no serán utilizadas. También se debe tener en cuenta en la definición de las actividades que se utilizará una arquitectura Modelo Vista Controlador MVC.

1. Realizar Historias de Usuario
2. Comenzar desarrollo del Product BackLog
3. Planificar Sprint y Realizar Sprint BackLog
4. Diseñar Diagrama de Casos de Uso
5. Realizar Tareas de Ingeniería
6. Realizar Mockups
7. Diseñar Diagrama de Clases
8. Diseñar Modelo Entidad Relación
9. Codificar los Modelos
10. Codificar las Vistas
11. Codificar las Plantillas
12. Codificar el manejador de URL's
13. Refactorizar el código
14. Diseñar y aplicar Pruebas Funcionales
15. Realizar Integración
16. Revisión y Retrospectiva del Sprint

Se realizo el siguiente gráfico para una mejor interpretación de la metodología desarrollada con cada una de sus partes y de la forma que hace la iteración cada sprint.

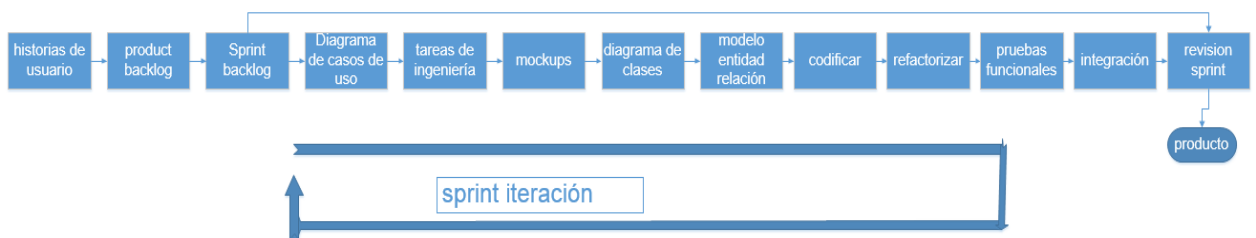


Figura 2 Metodología de desarrollo

7. DESARROLLO DEL PROYECTO

A continuación se mostrará de manera global algunos de los elementos utilizados para el desarrollo de este proyecto pero para ampliar toda la información al respecto podrá dirigirse al Manual Técnico donde se especifica con mayor detalle cada de los técnicas utilizadas en la metodología aplicada en este desarrollo.

7.1 LISTA DE HISTORIAS DE USUARIO

Se abordo a los conocedores del proceso de autoevaluación para la acreditación del programa de Ingeniería de Sistemas para recolectar los requerimientos adecuados para dar solución a esta problemática de la mejor manera y se logro al final de este proceso recolectar una serie de historias de usuario que listamos a continuación de forma abreviada.

No. Historia	Nombre	Prioridad	Iteración	Usuario
1	Crear sitio web principal	Alta	1	
2	Crear proyectos de Acreditación	Alta	1	Líder de proyecto
3	Eliminar proyectos de Acreditación	Alta	1	Líder de proyecto
4	Crear directorios	Alta	2	Colaborador y Líder de proyecto
5	Descargar archivos	Alta	2	Colaborador y Líder de proyecto
6	Subir archivos	Alta	2	Colaborador y Líder de proyecto
7	Eliminar archivos	Alta	3	Colaborador y Líder de proyecto
8	Evaluar factores	Alta	3	Líder de proyecto
9	Gestionar equipo de trabajo	Alta	4	Líder de proyecto
10	Gestionar tareas	Alta	4	Líder de proyecto

Tabla 2 Lista de historias de usuario

En el manual técnico se encuentran cada historia de usuario con mayor detalle.

7.1.1 Usuarios de la aplicación

Usuario	Descripción
Director o lider de proyecto	Para cada proyecto nuevo de acreditación en los diferentes programas se asigna un lider para que de orientación de principio a fin.
Colaborador	En los proyectos de acreditación se organizan grupos para elaborar los diferentes factores y a cada grupo se van ingresando los colaboradores los cuales van a trabajar en estos procesos.

En la Figura 3 se muestra el alcance que tiene cada usuario con respecto a las funcionalidades o servicios que ofrece la aplicación para entender mejor las posibilidades que tiene cada uno de estos.



Figura 3 Arquitectura de servicios

A continuación se muestran los diseños que se realizaron para el desarrollo de la aplicación como son: diagrama de casos de uso, algunos prototipos de interfaces y funcionalidades, diagrama de clases y modelo entidad relación.

7.2 DIAGRAMA DE CASOS DE USO

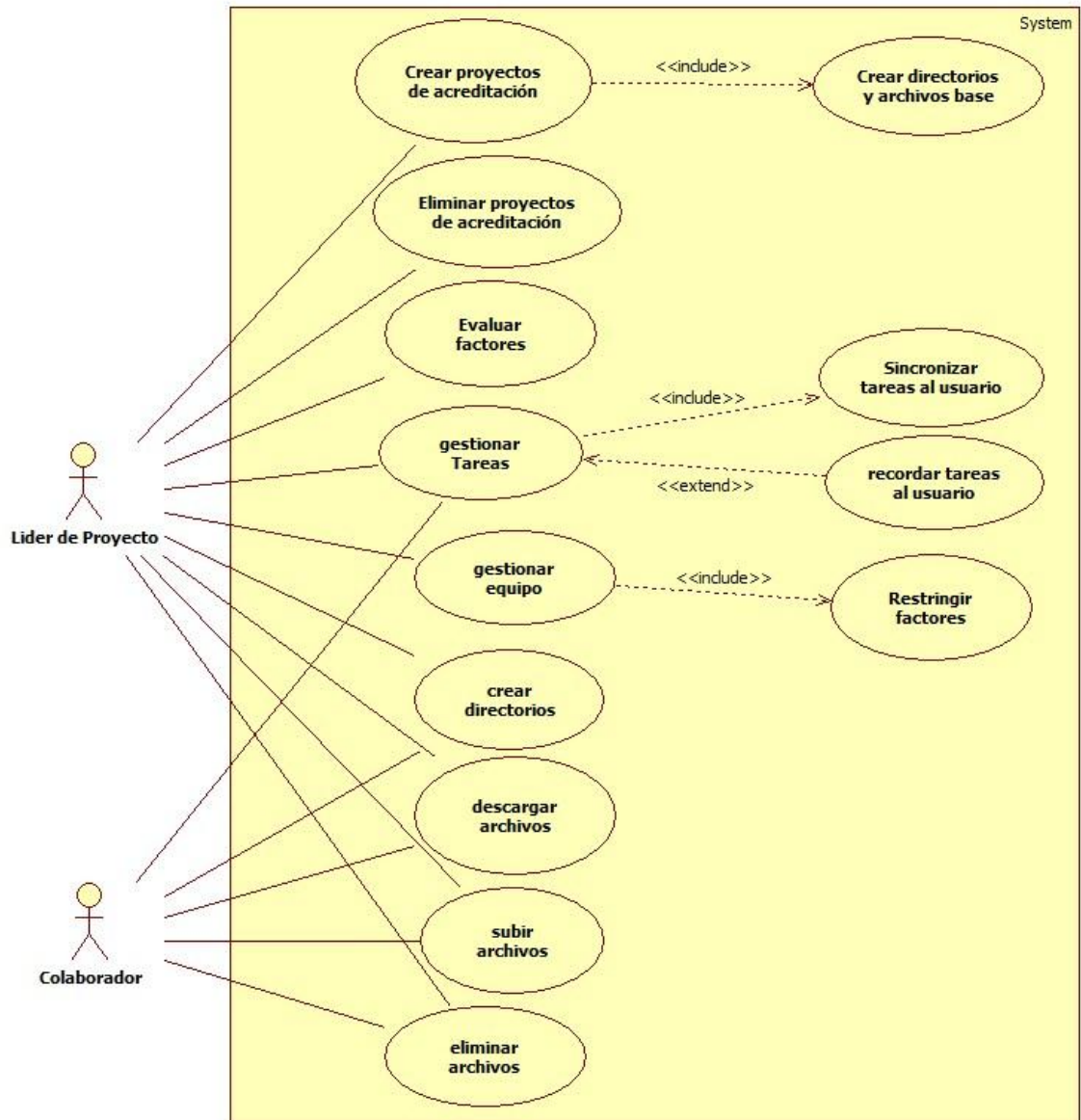
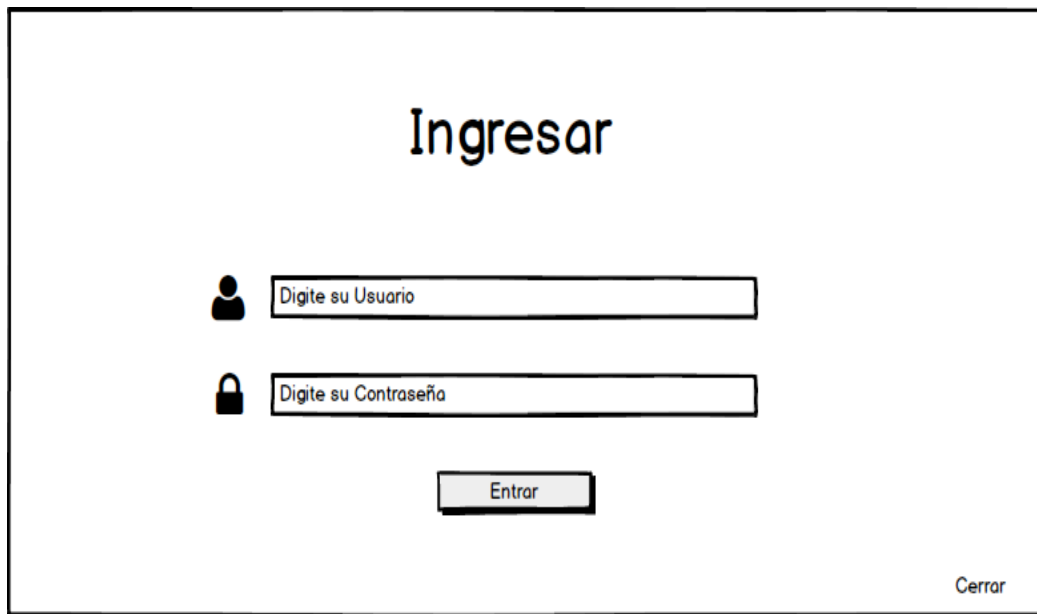


Figura 4 Diagrama de casos de uso

7.3MOCKUPS

Se elaboraron algunos prototipos de interfaces gráficas para tener una mejor percepción del sistema a realizar, fue de gran ayuda en la recolección de los requerimientos puesto que se pudo interpretar y aclarar la finalidad del sistema.

7.3.1 Inicio de sesión



The mockup shows a login interface with the following elements:

- Header: "Ingresar" (Login)
- User input field: Labeled "Digite su Usuario" (Enter your Username) with a user icon.
- Password input field: Labeled "Digite su Contraseña" (Enter your Password) with a lock icon.
- Login button: Labeled "Entrar" (Login).
- Close button: Labeled "Cerrar" (Close) in the bottom right corner.

Figura 5 Inicio de sesión

7.3.2 Proyectos de acreditación

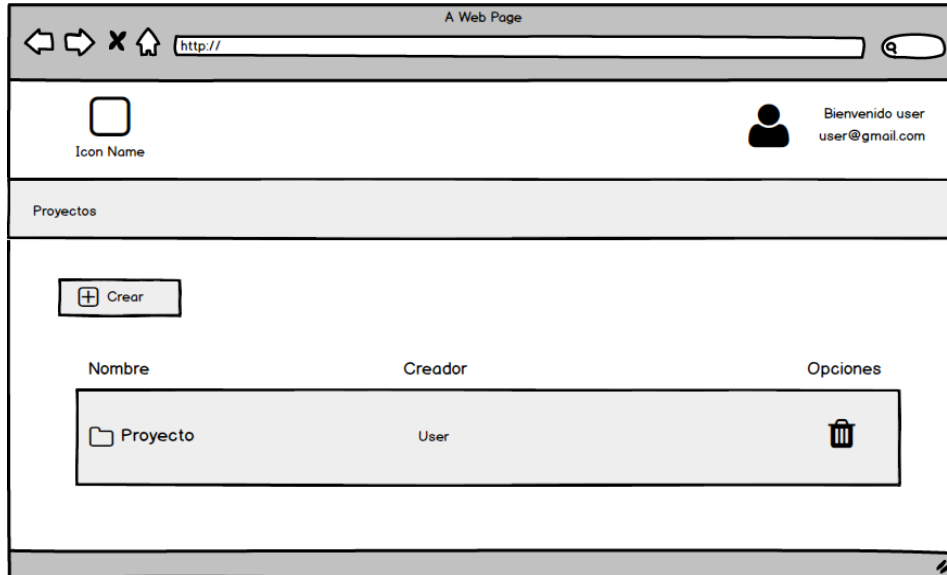


Figura 6 Proyectos de acreditación

7.3.3 Factores

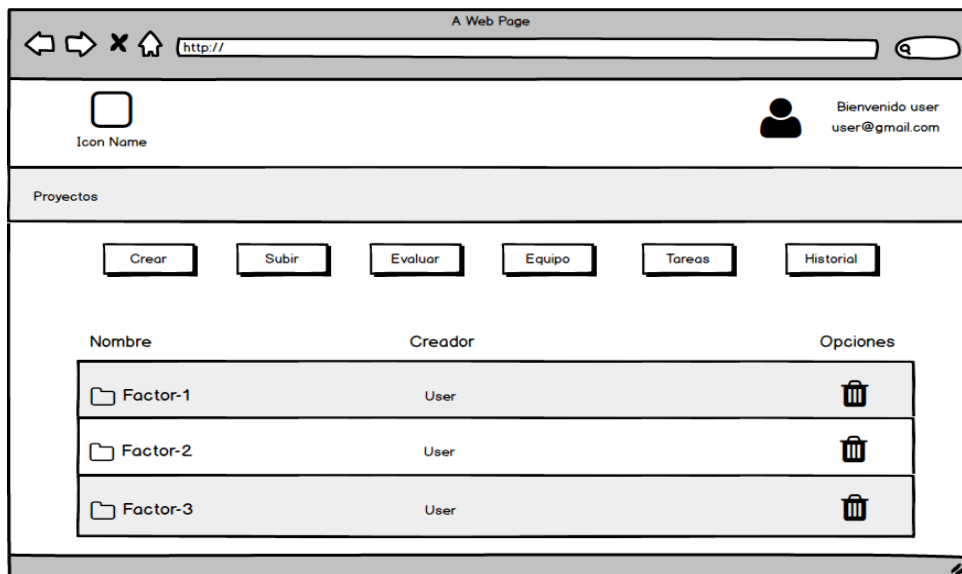


Figura 7 Factores

7.3.4 Factor

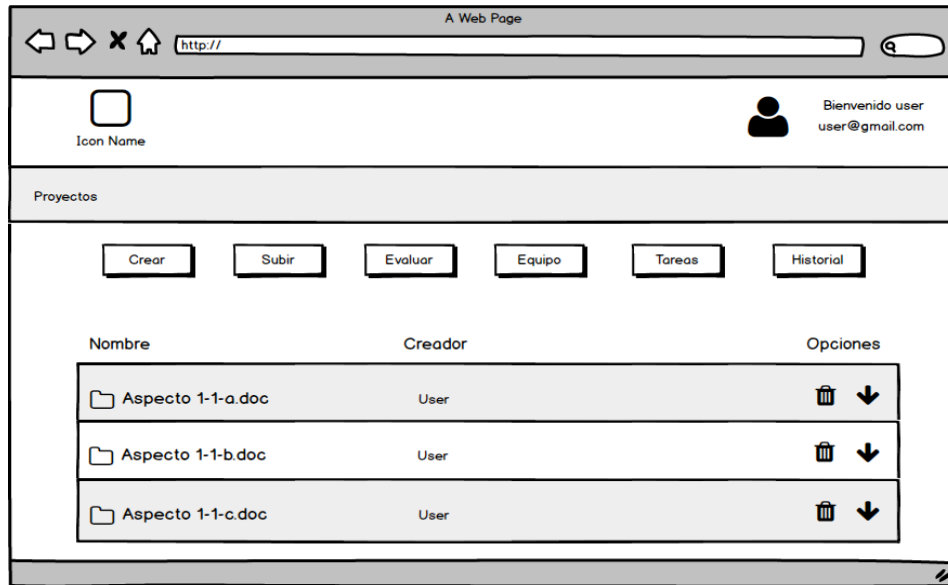


Figura 8 Factor

7.3.5 Evaluar factores

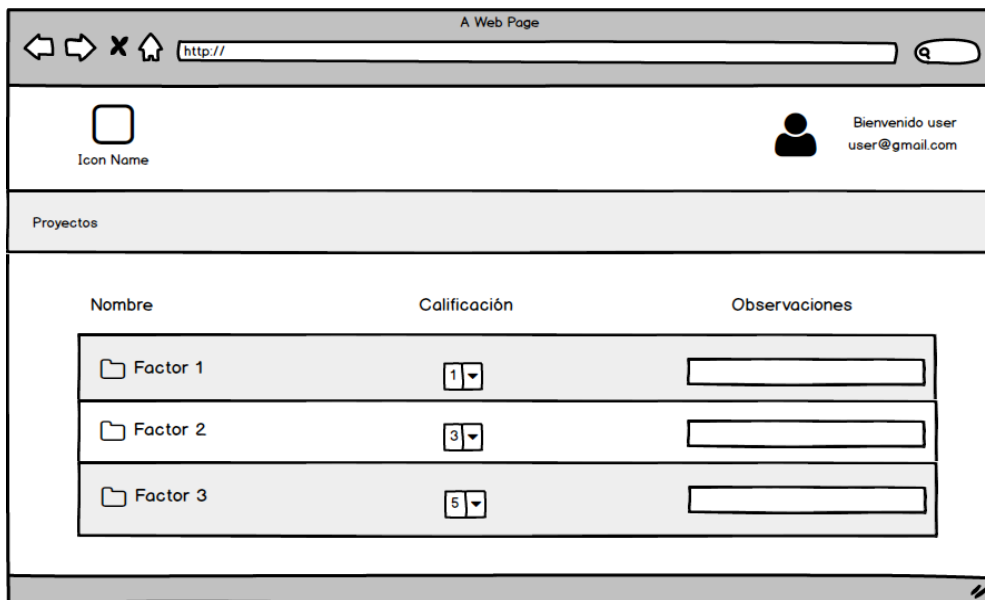


Figura 9 Evaluar factores

7.3.6 Equipo de trabajo

A Web Page

http://

Icon Name

Bienvenido user
user@gmail.com

Proyectos

Equipo de Trabajo

Usuario
User

Factores
Factor 1

Guardar Cancelar

Figura 10 Equipo de Trabajo

7.3.7 Gestionar tareas

A Web Page

http://

Icon Name

Bienvenido user
user@gmail.com

Proyectos

Gestión de Tareas

Usuarios
User

Tarea

Fecha Inicial

Fecha Final

Guardar Cancelar

Figura 11 Gestionar Tareas

7.3.8 Tareas programadas

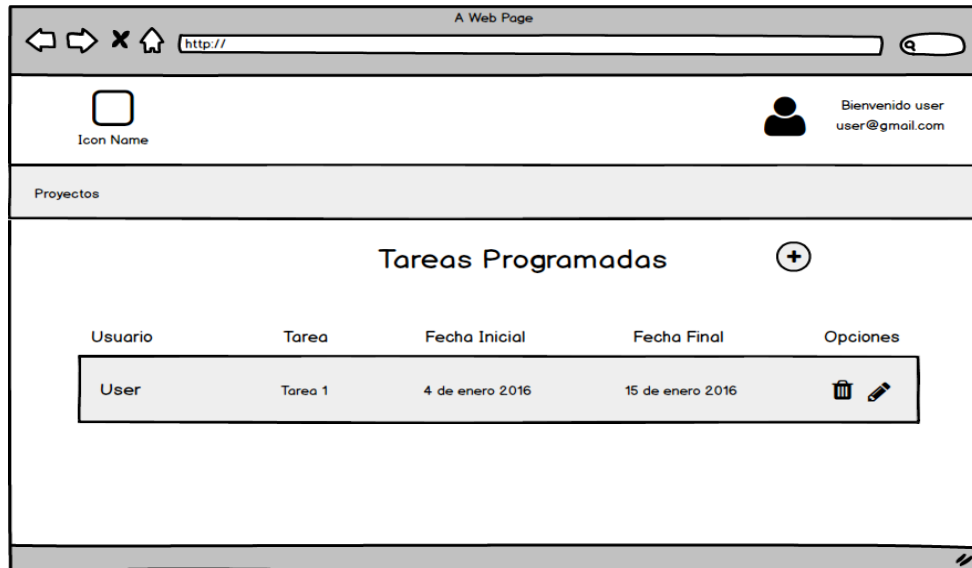


Figura 12 Tareas Programadas

7.3.9 Historial de acciones

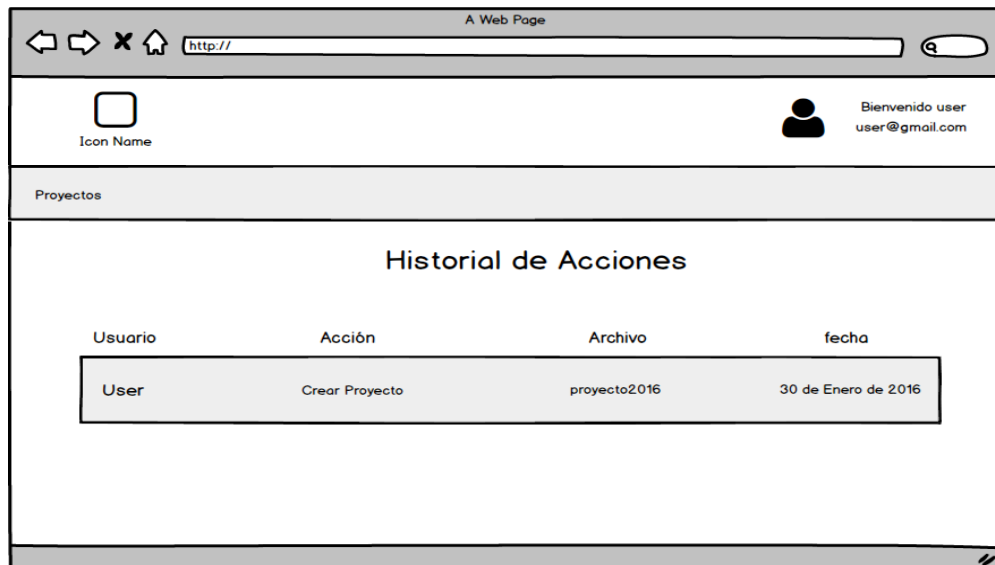


Figura 13 Historial de Acciones

7.3.10 Registrar usuario

The image shows a web browser window titled "A Web Page". The address bar contains "http://". The page header includes a search icon, a user profile icon, and the text "Bienvenido user user@gmail.com". Below the header is a section labeled "Proyectos". The main content area contains a registration form with the following fields and labels:

- Nombre :
- Apellido :
- Clave :
- Nueva Clave :
- Repetir Nueva Clave :

At the bottom of the form are two buttons: "Guardar" and "Cancelar".

Figura 14 Historial de Acciones

7.4 DIAGRAMA DE CLASES

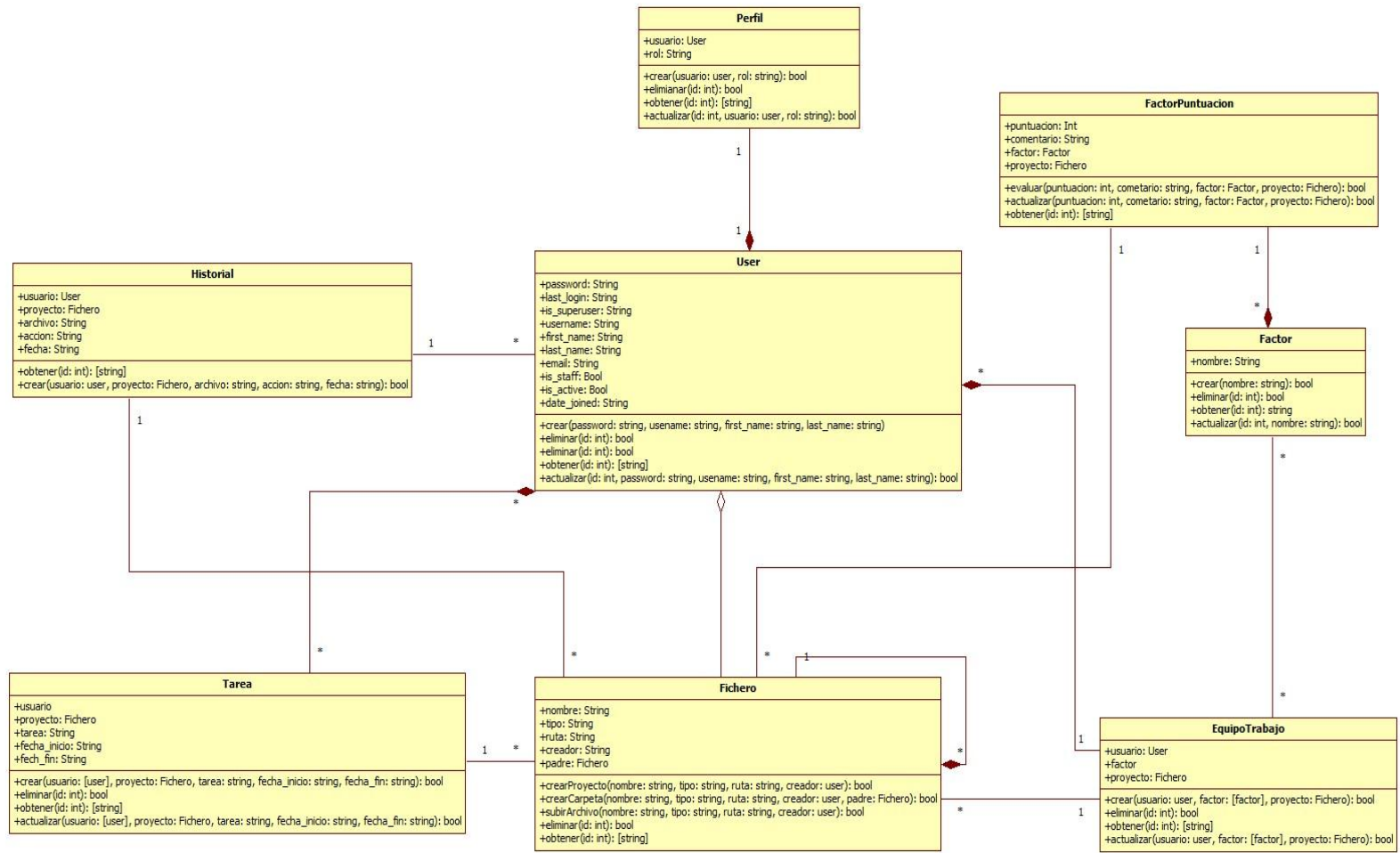


Figura 15 Diagrama de Clases

7.5 MODELO ENTIDAD RELACIÓN

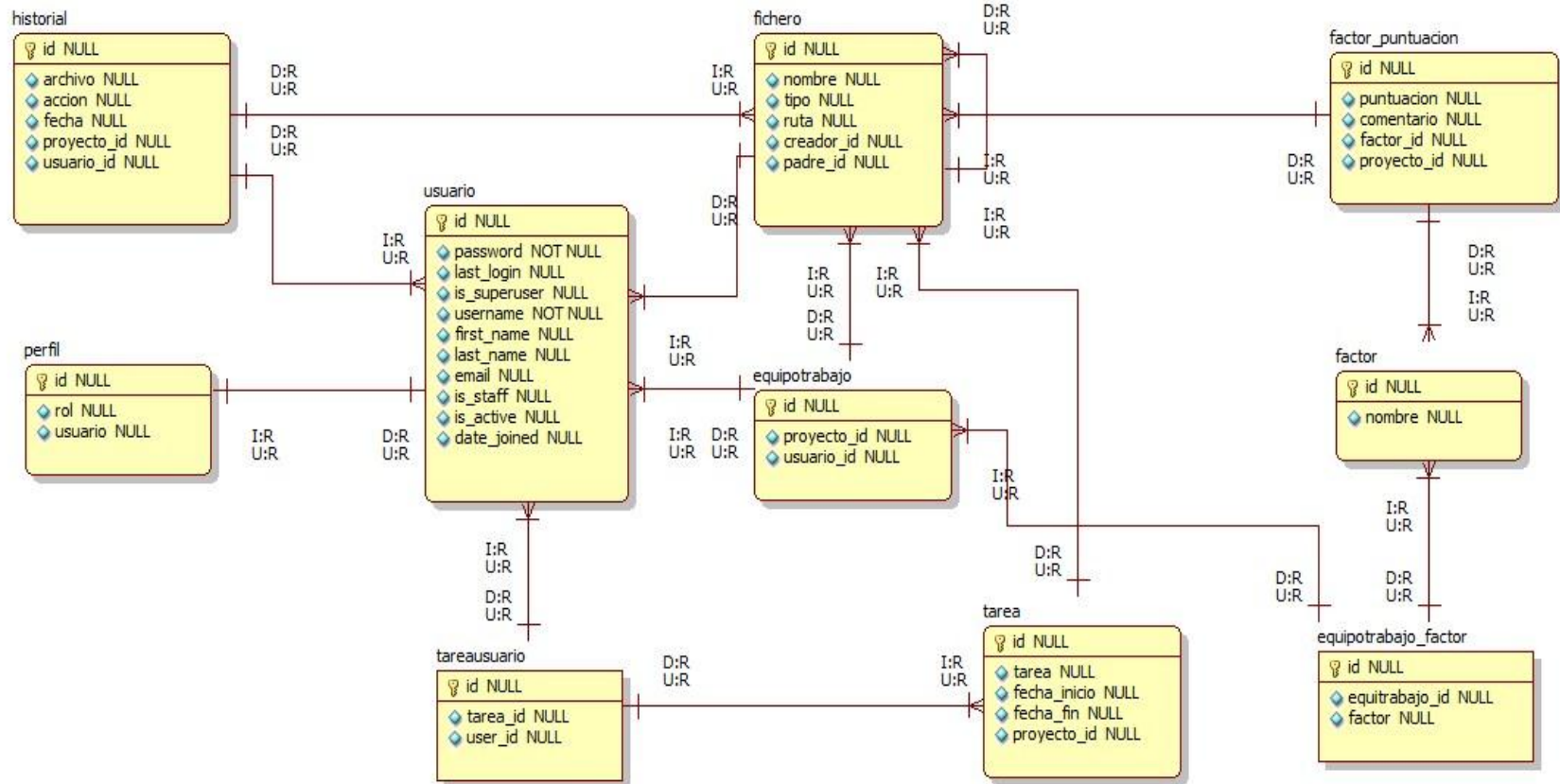


Figura 16 Modelo entidad relación

7.6 SPRINT BACKLOG

Aquí se describe cada una de las tareas que se realizaron en cada iteración del proceso de desarrollo de la aplicación.

7.6.1 Iteración 1

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
1	1	Crear proyecto inicial	Desarrollo	1

Tabla 3 Iteración 1 – Crear proyecto inicial

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
1	2	Diseñar web	Diseño	4

Tabla 4 Iteración 1 – Diseñar web

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
1	3	Crear plantillas HTML	Desarrollo	4

Tabla 5 Iteración 1 – Crear plantillas HTML

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
1	4	Configurar el manejador de URL's	Desarrollo	1

Tabla 6 Iteración 1 – Configurar el manejador de URL's

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
1	5	Configurar el módulo de contactos	Desarrollo	3

Tabla 7 Iteración 1 – Configurar el modulo de contactos

No.	No.	Descripción Tarea	Tipo Tarea	Duración
-----	-----	-------------------	------------	----------

Historia	Tarea			(días)
1	6	Iniciar Sesión	Desarrollo	2

Tabla 8 Iteración 1 – Iniciar sesión

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
2	7	Diseñar modelo para proyectos	Desarrollo	3

Tabla 9 Iteración 1 – Diseñar modelo para proyectos

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
2	8	Crear plantillas para proyectos	Desarrollo	4

Tabla 10 Iteración 1 – Crear plantillas para proyectos

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
2	9	Crear proyectos	Desarrollo	5

Tabla 11 Iteración 1 – Crear proyectos

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
3	10	Eliminar proyectos	Desarrollo	1

Tabla 12 Iteración 1 – Eliminar proyectos

7.6.2 Iteración 2

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
4	11	Crear directorios	Desarrollo	4

Tabla 13 Iteración 2 – Crear directorios

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
5	12	Descargar archivos	Desarrollo	2

Tabla 14 Iteración 2 – Descripción tarea

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
6	13	Subir archivos	Desarrollo	4

Tabla 15 Iteración 2 – Subir archivos

7.6.3 Iteración 3

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
7	14	Eliminar archivos	Desarrollo	2

Tabla 16 Iteración 3 – Eliminar archivos

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
8	15	Crear modelo para la evaluación de factores	Desarrollo	3

Tabla 17 Iteración 3 – Crear modelo para la evaluación de factores

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
8	16	Hacer el formulario para la evaluación	Desarrollo	4

Tabla 18 Iteración 3 – Hacer el formulario para la evaluación

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
8	17	Proceso de evaluación de un proyecto	Desarrollo	5

Tabla 19 Iteración 3 – Proceso de evaluación de un proyecto

7.6.4 Iteración 4

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
9	18	Modelo equipo de trabajo	Desarrollo	3

Tabla 20 Iteración 4 – Modelo equipo de trabajo

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
9	19	Codificar formulario y plantilla de la tarea	Desarrollo	4

Tabla 21 Iteración 4 – Codificar formulario y plantilla de la tarea

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
9	20	Ingresar equipo	Desarrollo	4

Tabla 22 Iteración 4 – Ingresar equipo

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
10	21	Modelo de tareas	Desarrollo	2

Tabla 23 Iteración 4 – Modelo de tareas

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
10	22	Formulario y plantilla de tareas	Desarrollo	3

Tabla 24 Iteración 4 – Formulario y plantilla de tareas

No. Historia	No. Tarea	Descripción Tarea	Tipo Tarea	Duración (días)
10	23	Ingresar tarea	Desarrollo	4

Tabla 25 Iteración 4 – Ingresar tareas

8. RESULTADOS

Al finalizar el desarrollo de este proyecto se obtuvieron los siguientes resultados:

1. El módulo para el almacenaje e ingreso de la documentación.
En el desarrollo de la aplicación este fue uno de los módulos primordiales puesto que ingresar los documentos de cada uno de los procesos de acreditación y autoevaluación se hace muy importante en estos y por eso dejamos que el usuario tuviese la posibilidad de cargar al servidor cualquier tipo de documentos hasta de 200 MB por archivo organizandolos por medio de un sistema de ficheros.
2. El módulo para la administración de la documentación
Hacer una administración de tipo CRUD para los diferentes archivos de cada uno de los procesos de acreditación y autoevaluación era el objetivo más importante y por eso fue el módulo en el que nos enfocamos para el desarrollo de esta aplicación y lo que se hizo fue un sistema de administración de archivos basado en como lo hacen los sistemas operativos convencionales y un valor agregado fue otorgar permisos para las diferentes acciones por medio de roles de usuario.
3. El módulo para la gestión de usuarios
Para la gestión de usuarios se utilizo la aplicación administrativa que ofrece el framework django haciendo uso del modelo que ya ofrece para administración de usuarios y agregandole un modelo para hacer los roles de cada de uno de ellos y así generando un administrador CRUD de estos dos modelos y obtener las ventajas de utilizar esta aplicación administrativa.
4. Despliegue de la aplicación en un servidor web
El despliegue de la aplicación se realizo utilizando una instancia estándar D1(1 Nucleo, 3.5 GB de memoria) en Microsoft Azure en ejecución con Ubuntu 14.04 LTS y servidor web NGINX.

9. CONCLUSIONES

Se ha logrado culminar el desarrollo de este proyecto, el cual para terminarlo con éxito se plasmaron unos objetivos que fueron el camino para este propósito; el equipo analizo los procedimientos basados en los lineamientos que propone el Consejo Nacional de Acreditación CNA y el proceso de acreditación y autoevaluación que llevo a cabo el programa de Ingeniería de Sistemas e Ingeniería Electrónica, basándonos en esto se concluye que no se puede diseñar un modelo de datos puesto que de acuerdo a la experiencia del programa y lo que propone el CNA se contempla hacer mejor una administración de ficheros así se logra implementar el Sistema de Información Web para almacenar y administrar los documentos generados en los diferentes planes de mejoramiento y los procesos de acreditación. Se desarrolla una base de documentos recopilando la información de los procesos anteriores haciendo una estructura de directorios y archivos para empezar proyectos venideros sin tener que volver a armar toda la documentación y conseguir información que usualmente no va a cambiar en la institución.

Finalmente aprovechando la forma de trabajar del Grupo de Autoevaluación del Programa GAP que organiza los equipos para trabajar en los diferentes factores para generar los documentos en el proceso, teniendo en cuenta esto, se le agregaron nuevas funcionalidades a la aplicación haciendo un módulo para crear equipos asignandole a cada equipo los factores a trabajar, también se hizo un módulo para asignar tareas a cada usuario y enviar un email automático mensual para recordar a cada usuario hacer sus tareas del proyecto, estas funcionalidades no estaban contempladas inicialmente pero ayudan a organizar el trabajo del GAP.

10. TRABAJOS FUTUROS

Pensando en posteriores mejoras para la aplicación es importante actualizar la base de documentos que se construyeron puesto que se baso en los lineamientos del año 2013 y en futuros años estos podrían cambiar y así mismo cambiaría toda la estructura de los proyectos de acreditación que se construyan de ahí en adelante. También sería muy importante trabajar en un modulo de reportes para tener a la mano cierta información valiosa como las tareas realizadas por cada integrante, las tareas pendientes por integrante, los equipos que se han conformado, el historial de acciones de cada integrante, entre otros más que se podrían utilizar y que serían de mucha ayuda.

Visualizando las mejoras orientadas hacia las tecnologías utilizadas sería interesante aumentar la interactividad y usabilidad de la aplicación en la parte de administración de documentos haciendo uso mayor de AJAX y lograr así un uso más transparente de esta. Se debe mejorar la seguridad donde se almacenan los documentos que los usuarios están subiendo dentro de la aplicación porque no se implementó ningún tipo de cifrado para proteger estos documentos.

10 REFERENCIAS BIBLIOGRÁFICAS

- 1.Senn, J. A. (1992). *Análisis y Diseño de Sistemas de Información*. México : McGraw Hill.
- 2.Software, S. (s.f.). *Scripta Software*. Recuperado el 17 de 11 de 2014, de <http://scriptasoftware.com/portal/index.php/licenciamiento/software-de-acreditacion>
- 3.DSpace. (s.f.). *DSpace*. Recuperado el 14 de 11 de 2014, de <http://www.dspace.org/> Eprints. (s.f.).
- 4.*Eprints*. Recuperado el 17 de 11 de 2014, de <http://www.eprints.org/software/>
- 5.*IEEE Standard Glossary of Software Engineering Terminology*, IEEE std 610.12-1990, 1990.
- 6.*Agile Project Management with Scrum*, Ken Schwaber, Microsoft Press, January 2004, 163pp.
- 7.Start Uml, recuperado el 29 de agosto de 2015, <http://staruml.io/>
- 8.Angell, I.O. and Smithson S. (1991) *Information Systems Management: Opportunities and Risks*
- 9.Ley de Protección de Datos, consultado el 20 de noviembre de 2015, de <http://www.infoleg.gov.ar/infolegInternet/anexos/60000-64999/64790/norma.htm>
- 10.PostgreSQL Global Development Group. Consultado el 20 de noviembre de 2015, de <http://www.postgresql.org/about/licence/>
- 11.Ian Hickson y David Hyatt, ed. HTML5, consultado el 20 de noviembre de 2015 , de <http://www.w3.org/html/wg/drafts/html/master/introduction.html#html-vs-xhtml>
- 12.Materialize, consultado el 20 de noviembre de 2015, de <http://materializecss.com/>
- 13.ECMA Script, Language Specification, consultado el 20 de noviembre de 2015, de <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>
- 14.Ajax: A New Approach to Web Applications, consultado el 20 de noviembre de 2015 <http://adaptivepath.org/ideas/ajax-new-approach-web-applications/>
- 15.The Making of Python, consultado el 20 de noviembre de 2015 <http://www.artima.com/intv/pythonP.html>

BIBLIOGRAFÍA

BAHIT, Eugenia. Python para principiantes. Ceative Commons no comercial.

BIBEAULT, Bear; KATZ, Yehuda. Jquery 1.4; traducción Manuel Cegarra Polo. Madrid: Anaya Multimedia, 2011. 528 p.

CONSEJO NACIONAL DE ACREDITACIÓN. Lineamientos para la acreditación de programas de pregrado. Bogotá D.C., 2013. 52 p.

FIRTMAN, Maximiliano. Ajax: web 2.0 con jQuery para profesionales. 2 edición. Buenos Aires: Alfaomega, 2010. 321 p.

GARCIA M., Saul. La guía definitiva de Django: Desarrolla aplicaciones web de forma rápida y sencilla. 2015. GNU Free Documentation License. 598 p.

GUIDO, Van Rossum. El tutorial de Python. Python Software Foundation, 2009. 116 p.

INSTITUTO COLOMBIANO DE NORMALIZACIÓN Y CERTIFICACIÓN. Referencias Bibliográficas: contenido, forma y estructura. NTC 5613. Bogotá D.C.: El Instituto, 2008. 33 p.

INSTITUTO COLOMBIANO DE NORMALIZACIÓN Y CERTIFICACIÓN. Documentación: presentación de tesis, trabajos de grado y otros trabajos de investigación. NTC 1486. Bogotá D.C.: El Instituto, 2008. 36 p.

KENDALL, Kenneth E; KENDALL, Julie E. Análisis y diseño de sistemas de información. 8 edición. México: Prentice-Hall, 2011. 600 p.

KNIBERG, Henrik. SCRUM y XP desde las trincheras: cómo hacemos SCRUM. Estados Unidos: InfoQ, 2007. 122 p.

KNOWLTON, Jim. Python; traductor María Jesús Fernández Vélez. Madrid: Ediciones Anaya Multimedia, 2008. 272 p.

LÓPEZ QUIJADO, José. Domine Javascript. 3 edición. México: Alfaomega: Ra-Ma, 2011. 705 p.

MELONI, Julie C. HTML5, CSS3 y Javascript; traductor Sergio Luis Gonzalez Cruz. Madrid: Ediciones Anaya Multimiedia, 2012. 655 p.

PABÓN FERNÁNDEZ, Nohra, et al. Evaluación y acreditación universitaria: base de datos y estado del arte. Santa fé de Bogotá: Universidad Pedagógica Nacional, Fundación Universidad de Bogotá Jorge Tadeo Lozano, 1996. 118 p.

PIATTINI VELTHUIS, Mario G., et al. Tecnología y diseño de bases de datos relacionales. México: Alfaomega: Ra-Ma, 2006. 946 p.

PRESSMAN, Roger S. Ingeniería del Software: Un enfoque práctico. Séptima Edición. México: McGraw-Hill, 2010. 808 p.

RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady. El lenguaje unificado. 2 edición. España: Pearson Educación S.A., 2007. 464 p.

RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady. El lenguaje unificado: manual de referencia. 2 edición. España: Pearson Educación S.A., 2007. 688 p.

SCHULZ, Ralph. Diseño Web con CSS. Barcelona: Alfaomega: Marcombo Ediciones Técnicas, 2009. 304 p.

SILBERSCHATZ, Abraham. Fundamentos de diseño de bases de datos. 5 edición. España: McGraw-Hill, 2007. 600 p.

SOMMERVILLE, Ian. Ingeniería de Software. 9 edición. México: Pearson Educación, 2011. 792 p.

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. PostgreSQL 8.4.22 Documentation. 2014. 2371 p.