UCAB Universidad Católica ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

Aplicación web para la gestión del personal docente por parte de las unidades académicas de la Universidad Católica Andrés Bello

Este Jurado; una vez realizado el examen del presente trabajo ha evaluado su contenido con el resultado: Diecinueve (19) puntos.

J U R A D O E X A M I N A D O R

Firma:	Firma:	Firma:
Nombre: Valeria León	Nombre: Henry Martínez	Nombre: Gloria Tarrío V.

REALIZADO POR	**Gómez Omaña, Eduardo Elías**
TUTOR EMPRESARIAL	**Ing. Benal Vera, Roquefelix**
TUTOR ACADÉMICO	**Ing. Tarrío Villaverde, Gloria**
FECHA	**Diciembre 2016**

UCAB Universidad Católica ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

Aplicación web para la gestión del personal docente por parte de las unidades académicas de la Universidad Católica Andrés Bello

TRABAJO INSTRUMENTAL DE GRADO
Presentado ante la
UNIVERSIDAD CATÓLICA ANDRÉS BELLO
Como parte de los requisitos para optar al título de

INGENIERO EN INFORMÁTICA

REALIZADO POR	**Gómez Omaña, Eduardo Elías**
TUTOR EMPRESARIAL	**Ing. Benal Vera, Roquefelix**
TUTOR ACADÉMICO	**Ing. Tarrío Villaverde, Gloria**
FECHA	**Diciembre 2016**

TABLA DE CONTENIDO

TABLA DE CONTENIDO 2

AGRADECIMIENTOS 5

SINOPSIS 6

CAPÍTULO I – NECESIDADES DE LA EMPRESA 7

I.1 Necesidades de la Empresa 7
I.2 Solución Propuesta 8
I.3 Objetivo General 10
I.4 Objetivos Específicos 10
I.4.1 Aporte Tecnológico 10
I.4.2 Aporte Funcional 10
I.5 Alcance y Limitaciones 11
I.5.1 Alcance 11
I.5.1.2 Aporte Tecnológico 15
I.5.1.3 Aporte Funcional 16
I.5.2 Limitaciones 17
I.6 Justificación 17

CAPÍTULO II – MARCO TEÓRICO 18

II.1 Universidad Católica Andrés Bello 18
II.2 Aplicación Web 19
II.2.1 El Cliente 20
II.2.2 El servidor 21
II.3 Patrón Modelo Vista Controlador (MVC) 21
II.3.1 Componentes del Patrón Modelo Vista Controlador (MVC) 22
II.4 Servicio Web 23
II.4.1 Ventajas de los servicios web 24
II.5 HTML 24
II.6 CSS 25
II.7 Framework 25
II.8 Javascript 25
II.9 Jquery 26
II.10 JSON (JavaScript Object Notation) 26
II.11 Bootstrap 27
II.12 ASP.NET MVC4 27
II.13 Entity Framework 28
II.14 Base de Datos Relacionales 28
II.15 Lenel Systems International 29
II.16 Sistema Académico Banner 29

II.17 Sistema de administración de personal - Picasso: 30
II.18 Sistema Administrador de Base de Datos 31
II.19 SQL Server 31
II.20 LDAP 31
II.20.1 Ventajas de LDAP 32

CAPÍTULO III – MARCO METODOLÓGICO 33
III.1 SCRUM 33
III.1.1 Características de Scrum 34
III.2 Proceso de Scrum 34
III.3 Rol~~es~~ de Scrum 36
III.4 Artefacto~~s~~ de Scrum 37
III.5 Justificación de la metodología 38
III.6 Sprint Backlog 39

CAPÍTULO IV - DESARROLLO 42
IV.1 Sprint 0 42
IV.1.1 Inicio 42
IV.1.2 Product Backlog 43
IV.1.3 Creación del Modelo Entidad Relación 44
IV.2 Sprint 1 46
IV.2.1 Rediseñar el proceso de control de asistencia de los profesores. 46
IV.2.2 Diseñar workflow para el proceso de solicitud de remoción de inasistencia. 47
IV.3 Sprint 2 48
IV.3.1 Definir la estructura del API de servicios web que será utilizada por la aplicación 48
IV.3.2 Integración de la aplicación con API de servicios web. 51
IV.4 Sprint 3 53
IV.4.1 Diseño de las interfaces de usuarios 53
IV.4.2 Login de Usuario 54
IV.4.3 Desarrollo de servicios web para el módulo de perfil 55
IV.4.4 Desarrollo de módulo de Perfil 55
IV.5 Sprint 4 56
IV.5.1 Desarrollo de servicios web para el módulo de datos curriculares 56
IV.5.2 Desarrollo de módulo de datos curriculares 56
IV.6 Sprint 5 58
IV.6.1 Servicios web para módulo de inasistencias 58
IV.6.2 Desarrollo de módulo de inasistencias 59
IV.7 Sprint 6 64
IV.7.1 Rediseñar el proceso de búsqueda de profesores para las escuelas. 64
IV.7.2 Desarrollo de servicios web para el módulo de ofertas. 65
IV.7.3 Desarrollo de módulo de ofertas. 65
IV.8 Sprint 7 72
IV.8.1 Desarrollo de módulo de administración 72

CAPITULO V - RESULTADOS 73

V.1 Resultados según Objetivos 73
V.2 Aporte Tecnológico 75
V.3 Aporte Funcional 76

CAPÍTULO VI - CONCLUSIONES Y RECOMENDACIONES 78
VI.1 Conclusiones 78
VI.2 Recomendaciones 80

BIBLIOGRAFÍA 81
Referencias Electrónicas 81

APÉNDICE 83
Apéndice A: Figuras 83
A1. Pantalla de resultados de búsqueda de profesor. 83
A2. Pantalla de agregar datos curriculares. 83
A3. Pantalla módulo de inasistencias. 84
A4. Pantalla "Mis Inasistencias". 84
A5. Pantalla "Crear Oferta". 85
A6. Pantalla "Mis Ofertas". 85
A7. Pantalla "Administración". 86
A8. Bases de datos SQL Server 86

AGRADECIMIENTOS

El principal agradecimiento, a quien me ha guiado en este camino, el que está conmigo en todo momento, Dios.

A mi madre, padre y hermana, por haberme forjado como la persona que soy en la actualidad; muchos de los logros se los debo a ustedes, en los que incluyo este.

A mi familia, que ha dedicado tiempo para enseñarme nuevas cosas, para brindarme aportes invaluables que servirán para toda mi vida.

A la universidad, porque las oportunidades que me ha brindado son incomparables, incluyendo a los profesores y compañeros, por todos conocimientos y el apoyo que me han otorgado.

A los amigos que trato como hermanos, pues en realidad lo son, solo que de diferente sangre, por sus valiosas ayudas y aportes a este proyecto y por todos los buenos momentos que hemos pasado.

Gracias.

SINOPSIS

El presente Trabajo Instrumental de Grado titulado "Aplicación web para la gestión del personal docente por parte de las unidades académicas de la Universidad Católica Andrés Bello", se desarrolló bajo los lineamientos de la Dirección de Tecnología de Información (DTI) y tiene como fin integrar la información académica y personal de los profesores que cumplen funciones dentro de la universidad, optimizar el sistema que controla la información sobre las inasistencias de los profesores y además la implementación de un nuevo sistema que permite a las escuelas crear ofertas de trabajo para alguna materia, a dichas materias los profesores pueden postularse y la escuela seleccione el profesor de su preferencia para el cargo.

Como parte de la administración de la aplicación, se cuenta con un módulo que permite manejar los roles y permisos que poseen los distintos usuarios dentro del sistema.

Se utilizó SCRUM, ágil y flexible, con desarrollo iterativo e incremental, el uso de bloques de trabajo (sprint) fue fundamental para concretar los objetivos planificados.

Finalmente, el entorno de desarrollo de la aplicación fue Visual Studio 2012, utilizando el framework Modelo Vista Controlador 4 (MVC 4), sobre los lenguajes de programación: C#, HTML, Jquery, CSS, SQL.

CAPÍTULO I – NECESIDADES DE LA EMPRESA

I.1 Necesidades de la Empresa

La Universidad Católica Andrés Bello (UCAB) desea optimizar e integrar la información personal y académica de los profesores que cumplen funciones dentro de la universidad. Actualmente cada escuela puede acceder al sistema BANNER 8 y hacer diferentes consultas para obtener la información.

Como parte de esta integración, la universidad requiere de un sistema de búsqueda, por distintos parámetros, donde se pueda acceder de manera más fácil y rápida a toda la información relacionada a los profesores.

La universidad también requiere de un sistema donde cada escuela tenga la oportunidad de ofrecer empleos a los profesores. Este sistema permitirá la creación de ofertas para profesores sobre alguna materia. La escuela debe generar una oferta, donde se detalle toda la información de la materia, tales como: los horarios, el contenido programático, jefe de cátedra, etc.

Los profesores podrán consultar las ofertas creadas por las escuelas, filtrar estas consultas por distintos parámetros y postularse como candidato en la materia.

La universidad también requiere una optimización en el sistema de asistencias, donde las escuelas puedan cargar inasistencias de manera más ágil y eficiente. Además, permitir a los profesores visualizar sus inasistencias y solicitar remociones de inasistencias en caso de encontrar alguna discrepancia. Estas

solicitudes de remoción serán aprobadas o rechazadas por las escuelas según los datos que se mostrarán al momento de tomar la decisión.

I.2 Solución Propuesta

Se propone para la solución del problema antes mencionado, diseñar, desarrollar e implementar una aplicación web que permita la consulta de datos, creación de ofertas de trabajo, postulaciones y gestión de asistencias para los profesores dentro de la Universidad Católica Andrés Bello (UCAB).

La aplicación web permitirá a las escuelas:

- Consultar datos personales, datos curriculares e historial académico de los profesores que forman, o formaron parte, de la UCAB.
- Cargar inasistencia a los profesores.
- Aprobar o rechazar solicitudes de remoción de inasistencias.
- Administrar ofertas académicas.

La aplicación web permitirá a los profesores:

- Consultar datos personales, datos curriculares e historial académico.
- Postularse a ofertas académicas.
- Revisar sus inasistencias.
- Solicitar remociones de inasistencias.

Adicionalmente, la aplicación contará con un módulo administrativo que permitirá:

- Según el rol del usuario, habilitar los distintos módulos de la aplicación.
- Agregar o eliminar excepciones, lo cual permitirá decidir a cuales módulos de la aplicación tendrá acceso un usuario en particular.

Para la elaboración de dicha aplicación será necesario el levantamiento y definición de todos los procesos, roles, responsabilidades e interacción entre unidades involucradas.

El sistema constará con un total de seis módulos para cumplir con los requerimientos, los cuales son:

- Módulo de autenticación (LDAP).
- Módulo administrativo.
- Módulo de gestión de datos personales, académicos y curriculares de profesores.
- Módulo de gestión de ofertas académicas.
- Módulo de gestión de asistencia.

Esta aplicación web está dirigida a aumentar la interacción de los profesores y el personal administrativo con las herramientas tecnológicas que brinda la universidad, agilizando las actividades diarias dentro y fuera del campus.

I.3 Objetivo General

Desarrollar una aplicación web que permita la gestión docente por parte de las unidades académicas dentro de Universidad Católica Andrés Bello (UCAB).

I.4 Objetivos Específicos

1. Desarrollar el módulo de autenticación (LDAP).
2. Desarrollar el módulo administrativo.
3. Desarrollar el módulo de gestión de datos personales, académicos y curriculares de profesores.
4. Desarrollar el módulo de gestión de ofertas académicas.
5. Desarrollar el módulo de control de asistencias.

I.4.1 Aporte Tecnológico

6. Integración con el sistema administrativo-académico Banner 8.
7. Integración con el sistema de control de acceso Lenel.
8. Integración con el sistema de gestión de personal Picasso.

I.4.2 Aporte Funcional

9. Rediseñar el proceso de control de asistencia de los profesores.
10. Rediseñar el proceso de búsqueda de profesores para las escuelas.

I.5 Alcance y Limitaciones

I.5.1 Alcance

- Desarrollar el módulo de autenticación (LDAP).

Desarrollar un módulo que permita el inicio de sesión a la aplicación utilizando sus credenciales UCAB, conformadas por un usuario único y contraseña. La autenticación estará basada en el servidor LDAP de la Universidad Católica Andrés Bello. La sesión permanecerá iniciada por un tiempo prestablecido, el cual puede ser configurable.

- Desarrollar el módulo administrativo.

Desarrollar un módulo que permita manejar roles y excepciones de los usuarios. Los roles de cada usuario dependerán del cargo que posean dentro de la universidad. Basados en estos roles los usuarios tendrán permisos sobre los distintos módulos que conforman la aplicación. Un usuario administrador podrá agregar o eliminar permisos a otro usuario sobre los módulos, que serán manejadas como excepciones y son editables. Los permisos otorgados por los roles no serán editables.

- Desarrollar el módulo de gestión de datos personales, académicos y curriculares de profesores.

  Para lograr este objetivo se implementarán las siguientes funcionalidades:

Mi Perfil: El usuario podrá ver sus datos personales, curriculares y académicos.

Datos Personales: El usuario podrá consultar los datos básicos, datos de contacto y las dependencias actuales dentro de la UCAB.

Datos Curriculares: El usuario podrá agregar títulos, eliminar títulos pendientes o rechazados, visualizar el motivo por el cual fue rechazado un título. Podrá realizar las mismas acciones para "Cursos y Certificaciones".

Historial Académico: Se mostrará una tabla con toda la información de los cursos dictados por el usuario dentro de la UCAB.

Consultar Perfil: El usuario podrá realizar una búsqueda mediante los parámetros: cédula, nombre, apellido o correo electrónico de la UCAB para consultar los "Datos Personales", "Datos Curriculares" e "Historial Académico" de un profesor en particular.

Datos Curriculares por Aprobar: El usuario podrá aprobar o rechazar solicitudes curriculares pendientes. Todas las decisiones estarán almacenadas y podrán ser reversadas.

- Desarrollar el módulo de gestión de ofertas académicas.

Para lograr este objetivo se implementarán las siguientes funcionalidades:

Crear Oferta: El usuario podrá crear ofertas académicas completando los siguientes campos:

Nombre de la materia

Semestre / año, horas a la semana

Disponibilidad del profesor

Contenido programático

Requisitos

Mis Ofertas: El usuario podrá consultar, editar y eliminar las ofertas creadas en la escuela a la que pertenece. El usuario también podrá consultar los datos y la cantidad de profesores que se hayan postulado a alguna oferta en específico.

Buscar Ofertas: El usuario podrá consultar las ofertas creadas y podrá ordenarlas por:

"Más postulaciones".

"Más recientes"

"Por escuelas".

El usuario puede postularse o eliminar su postulación de alguna oferta específica.

- Desarrollar el módulo de control de asistencias.

Para lograr este objetivo se implementarán las siguientes funcionalidades:

Cargar Inasistencias: El usuario podrá consultar las distintas secciones que estén asignadas a un profesor y cargar la cantidad de horas que tenga de inasistencia.

Mis Inasistencias: El usuario podrá visualizar sus inasistencias y en caso de ser necesario, solicitar una remoción de inasistencia, ya sea por olvido de firma o porque las horas de clases han sido recuperadas. Estas solicitudes de remoción serán enviadas a la escuela para ser revisadas y ser aprobadas y rechazadas según sea el caso.

Mis Aprobaciones: El usuario podrá rechazar o aprobar las distintas solicitudes de remoción que sean enviadas a la escuela a la que pertenece. Se mostrará al usuario toda la información sobre el día de la inasistencia para facilitar el proceso de toma de decisión. Todas las decisiones estarán almacenadas y podrán ser reversadas.

I.5.1.2 Aporte Tecnológico

Se diseñará e implementará una API que contendrá los distintos servicios web que permitirán la interacción con los sistemas: Picasso, Lenel y Banner 8, desde ésta y otras aplicaciones.

Esta aplicación será realizada en Microsoft Visual Studio 2015, bajo el lenguaje C# y el framework .NET

La comunicación será realizada mediante el protocolo HTTP y utilizando el estilo de arquitectura REST (Representational State Transfer).

El uso de la arquitectura REST cumplirá con las siguientes premisas:

Cliente/Servidor: Los servicios web son cliente/servidor y definen una comunicación entre ambos separando completamente las responsabilidades entre ambas partes.

Sin estado: Los servicios web no mantienen estado asociado al cliente, cada petición que se realiza a ellos es completamente independiente de la siguiente.

Servicios Uniformes: Todos los servicios REST compartirán una forma de invocación y métodos uniforme utilizando los métodos GET,POST,PUT ,DELETE.

I.5.1.3 Aporte Funcional

- Rediseñar el proceso de control de asistencia de los profesores.

Actualmente las inasistencias se cargan a través de una aplicación de escritorio que requiere: la localidad del campus, período, NRC y fecha para cargar la inasistencia. La nueva aplicación web reducirá el trabajo ya que sólo requerirá la cédula o nombre del profesor y cargará los NRC del período actual y las fechas de las clases a las cuales se les puede agregar una inasistencia por parte del profesor.

Actualmente un profesor debe dirigirse a la escuela para conocer sus inasistencias, ya sean por olvido de firma o por recuperaciones que no hayan sido agregadas. Si el profesor desea una remoción de inasistencia, se levanta un reporte y se utiliza la aplicación de escritorio que solicitará el período, NRC, y fecha de la inasistencia para poder realizar la remoción.

El nuevo proceso permitirá al profesor revisar sus inasistencias y al mismo tiempo solicitar una remoción sin dirigirse a la escuela.

El nuevo proceso también permite a las escuelas consultar estas solicitudes de remoción y tomar la decisión sobre aprobar o rechazar la solicitud, la cual el profesor podrá revisar en su lista de inasistencias.

- Rediseñar el proceso de búsqueda de profesores para las escuelas.

Actualmente las escuelas para solicitar profesores y publicar las vacantes en las distintas escuelas, recurren al envío de correos masivos a los distintos profesores.

Utilizando el nuevo proceso las escuelas podrán generar ofertas de trabajo para las distintas materias. Estas ofertas podrán ser listadas por los profesores y éstos postularse a distintas ofertas.

La escuela tomará la decisión entre los distintos postulados y contactará el mejor candidato para dictar la materia. La universidad actualmente no cuenta con un sistema similar a este.

I.5.2 Limitaciones

- Los usuarios finales serán personal de la UCAB.
- Funcionará únicamente en navegadores web.
- Datos de profesores pueden no estar actualizados, ya que se depende de bases de datos externas.

I.6 Justificación

La implantación de la aplicación web facilitará y optimizará el proceso de consulta de información de profesores. Optimizará la carga de inasistencias para el personal de escuela y facilitará a los profesores visualizar sus inasistencias. Permitirá a las escuelas encontrar los mejores profesores para dictar las materias y a los profesores conseguir nuevas oportunidades de trabajo.

CAPÍTULO II – MARCO TEÓRICO

Este capítulo presenta los conceptos teóricos y las tecnologías relevantes para el desarrollo de la solución.

II.1 Universidad Católica Andrés Bello

Es una universidad privada con sede en Caracas, Venezuela. Miembro de la Asociación de Universidades Confiadas a la Compañía de Jesús en América Latina (AUSJAL), es dirigida por padres jesuitas. Tiene su sede principal en Caracas y núcleos en Los Teques, Puerto Ordaz, Coro.

Está conformada por un total de cinco escuelas:

- Facultada de Ciencias Económicas y Sociales.
- Facultad de Derecho.
- Facultad de Humanidades y Educación.
- Facultad de Ingeniería.
- Facultad de Teología.

La universidad cuenta con una amplia lista de profesores, altamente capacitados, a dedicación y a tiempo convencional. Los profesores a dedicación se ubican en la sede principal de la Escuela. Entre sus funciones, se cuenta la asesoría a los estudiantes en las materias de su área de conocimiento, la cual se cumple en el marco del programa de profesor asesor.

Se considera inasistencia no justificada de profesor a la falta parcial o total a las horas programadas para dictar alguna clase.

Estas inasistencias pueden ser removidas si el profesor presenta alguna justificación explicando el motivo por el cual no asistió a las horas de clase o si realiza alguna recuperación, previamente planificada, de las horas no dictadas.

II.2 Aplicación Web

"Una aplicación web es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador) como el servidor (web) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones" (Luján Mora, 2002). Es decir, es un software que se ejecuta en un navegador web, que puede ser codificado en diferentes lenguajes de programación soportados por los navegadores web y su ventaja es que no depende de la instalación de ningún software como tal, sólo se necesita de un navegador web para su ejecución.

El protocolo HTTP forma parte de la familia de protocolos de comunicaciones TCP/IP, que son los empleados en Internet.

En la Figura 1 se puede observar la estructura básica de una aplicación web.

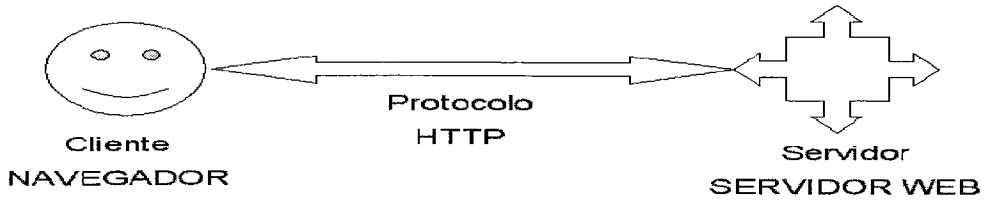


Figura 1: Estructura básica de una aplicación web.

Fuente: Luján Mora, 2002

II.2.1 El Cliente

La manera más sencilla de describir un cliente web es: "un programa con el que interacciona el usuario para solicitar a un servidor web el envío de los recursos que desea obtener mediante HTTP" (Luján Mora, 2002). La parte cliente de las aplicaciones web suele estar formada por el código HTML que forma la página web más algo de código ejecutable realizado en lenguaje de script del navegador (JavaScript o VBScript) o mediante pequeños programas realizados en Java. También se suelen emplear plug-ins que permiten visualizar otros contenidos multimedia.

II.2.2 El servidor

El servidor web es un programa que siempre está a la espera de solicitudes por parte de un cliente web utilizando una conexión mediante el protocolo HTTP. "En los sistemas Unix suele ser un "demonio" y en los sistemas Microsoft Windows un servicio" (Luján Mora, 2002).

II.3 Patrón Modelo Vista Controlador (MVC)

Para definir un patrón se puede partir de la definición:

"Un patrón describe un problema que ocurre una y otra vez en nuestro entorno, así como la solución a ese problema de tal modo que se puede aplicar esta solución un millón de veces, sin hacer lo mismo dos veces" (Alexander, 1977).

Modelo Vista Controlador (MVC) es un patrón o estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo (Alicante, 2012).

II.3.1 Componentes del Patrón Modelo Vista Controlador (MVC)

El Modelo contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia. Es responsable de acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento. Además define las reglas de negocio (la funcionalidad del sistema) (Alicante, 2012).

La Vista, o interfaz de usuario, compone la información que se envía al cliente y los mecanismos de interacción con este. Es responsable de recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.), además contiene reglas de gestión de eventos. Estas acciones pueden suponer peticiones al modelo o a las vistas (Alicante, 2012).

El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestiona el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno. Es responsable de recibir datos del modelo y los muestra al usuario (Alicante, 2012).

En la Figura 2 se puede observar el flujo que sigue el control generalmente.

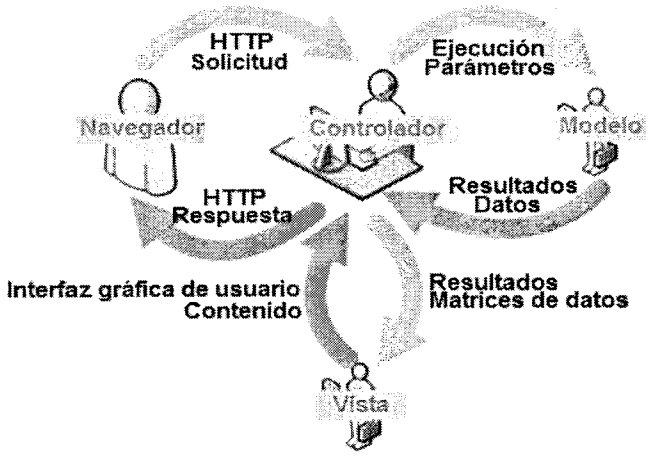


Figura 2: Flujo de estructura MVC.

Adaptado de: **Alicante, 2012**

II.4 Servicio Web

"Son un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web" (w3c, 2013). Todos los servicios web vienen acompañados de una descripción donde se plasman las operaciones que realizan los mismos, es decir, cómo se deben hacer las peticiones y cómo obtener las respuestas.

II.4.1 Ventajas de los servicios web

- Aumenta la interoperabilidad entre programas independientemente de la plataforma en donde están.
- Aumenta la interoperabilidad entre servicios y programas de diferentes compañías y ubicados en diferentes lugares geográficos.
- Fomentan los estándares y protocolos basados en texto, haciendo más fácil acceder y entender su contenido y funcionamiento.
- Al emplear HTTP, pueden utilizar un sistema firewall sin cambiar las reglas del filtrado.

II.5 HTML

Definiéndolo de la manera más sencilla, HTML es el lenguaje con el que se programan la mayoría de páginas web. "El HTML, acrónimo inglés de HyperText Markup Language, es el código que posibilita la creación y edición de documentos web. El código HTML utiliza el código ASCII puro que puede teclearse en cualquier editor básico para posteriormente ser interpretado secuencialmente por un objeto cliente denominado navegador (browser) que visualiza el resultado en pantalla" (Creative Commons, 2013).

II.6 CSS

"Las Hojas de Estilo en Cascada CSS (Cascade Style Sheet) son una serie de instrucciones que nos permiten dar formato aprovechando las limitaciones que nos presenta el HTML y sus etiquetas de formato al momento de dar formato a una página" (Creative Commons, 2013).

II.7 Framework

"Genéricamente, un framework es un conjunto de herramientas, librerías, convenciones y buenas prácticas que pretenden encapsular las tareas repetitivas en módulos genéricos fácilmente reutilizables" (LibrosWeb, 2014).

"Los frameworks se utilizan en el ámbito de la programación de aplicaciones desde hace décadas. Recientemente han comenzado a utilizarse para programar y diseñar aplicaciones web, por lo que ya existen decenas de frameworks para CSS" (LibrosWeb, 2014).

II.8 Javascript

JavaScript es un lenguaje de programación ampliamente utilizado en el mundo del desarrollo web por ser muy versátil y potente, tanto para la realización de pequeñas tareas como para la gestión de complejas aplicaciones.

"JavaScript es un lenguaje interpretado que, al contrario de las aplicaciones normales, que son ejecutadas por el sistema operativo, es ejecutado por el navegador que se utiliza para ver las páginas. Eso hace que se pueda desarrollar

aplicaciones de muy diverso tipo, desde generadores de HTML, comprobadores de formularios, etc..., hasta programas que gestionen las capas de una página. Pueden desarrollarse incluso aplicaciones que permitan poder tener capas en una página como si fueran ventanas, y dar la sensación de estar trabajando con una aplicación con interfaz de ventanas". (Atribución-CompartirIgual, 2011).

II.9 Jquery

"jQuery es una biblioteca JavaScript rápido, pequeño y rico en funciones." (JQuery, 2014). Entre sus funciones están recorrido HTML y manipulación de documentos, manejo de eventos, animación, mucho más simple con una API fácil de usar que funciona a través de una multitud de navegadores. Con una combinación de versatilidad y extensibilidad.

II.10 JSON (JavaScript Object Notation)

"Es un formato para el intercambio de datos, básicamente JSON describe los datos con una sintaxis dedicada que se usa para identificar y gestionar los datos." (Json, 2012). JSON nació como una alternativa XML, el fácil uso en JavaScript ha generado un gran número de seguidores de esta alternativa. Una de las mayores ventajas que tiene el uso de JSON es que puede ser leído por cualquier lenguaje de programación. Por lo tanto, puede ser usado para el intercambio de información entre distintas tecnologías.

II.11 Bootstrap

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

"Los diseños creados con Bootstrap son simples, limpios e intuitivos, esto les da agilidad a la hora de cargar y al adaptarse a otros dispositivos. El Framework trae varios elementos con estilos predefinidos fáciles de configurar: Botones, Menús desplegables, Formularios incluyendo todos sus elementos e integración jQuery para ofrecer ventanas y tooltips dinámicos." (Lessin, 2013).

II.12 ASP.NET MVC4

ASP.NET MVC 4 es un framework para crear aplicaciones web escalable basada en estándares, utilizando patrones de diseño bien establecidos y el poder de ASP.NET y .NET Framework.

El modelo arquitectónico Modelo-Vista-Controlador (MVC) separa una aplicación en tres componentes principales: el modelo, la vista y el controlador. El marco de ASP.NET MVC proporciona una alternativa al modelo de formularios Web Forms de ASP.NET para crear aplicaciones web. "El marco de ASP.NET MVC es un marco de presentación de poca complejidad y fácil de comprobar que (como las

aplicaciones basadas en formularios Web Forms) se integra con las características de ASP.NET existentes, tales como páginas maestras y la autenticación basada en pertenencia. El marco de MVC se define en el ensamblado System.Web.Mvc." (Microsoft, msdn.microsoft.com, 2014).

II.13 Entity Framework

Entity Framework (EF) es un asignador objeto-relacional que permite a los desarrolladores de .NET trabajar con datos relacionales usando objetos específicos del dominio. "Elimina la necesidad de la mayor parte del código de acceso a datos que los desarrolladores suelen tener que escribir." (Microsoft, msdn.microsoft.com, 2014).

Sobre esto se puede inferir que entity framework es el encargado de conectar a los sistemas creados con ASP.NET con las bases de datos proporcionando el acceso a datos.

II.14 Base de Datos Relacionales

Una base de datos se puede definir como un conjunto de información que pertenece al mismo contexto, que se encuentra agrupada o almacenada para su uso posterior. "Una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta." (Jackson, 2013).

II.15 Lenel Systems International

Es un sistema de control de acceso, integrado con un sistema de monitoreo de alarmas y circuito cerrado de video. Incorpora software orientado a objetos y una avanzada base de datos estructurada cliente/servidor.

La información de los usuarios es almacenada en la base de datos, luego es descargada a los controladores inteligentes del sistema Lenel, o codificados en las tarjetas inteligentes.

Este sistema permite llevar un registro de los movimientos (entradas y salidas) de los usuarios dentro del recinto.

II.16 Sistema Académico Banner

Banner es una herramienta tecnológica de gestión académica que facilita la administración y gestión académica que se realiza en la Universidad.

Entre sus principales características están:

- Mantiene registros detallados, consistentes y precisos sobre docentes y alumnos.
- Permite el acceso a la información sobre la asistencia de alumnos y docentes a partir de sus datos.
- Facilita el proceso de publicación de calificaciones finales y parciales.
- Permite la construcción de los horarios de semestres y trimestres futuros.

II.17 Sistema de administración de personal - Picasso:

Sistema informático que forma una interfaz entre la gestión de recursos humanos y la tecnología de la información.

Sus principales funciones son:

- Mantener centralizada la información del personal en el departamento de recursos humanos.
- Manejar múltiples tipos de nóminas con diferentes formas de pago y contratos colectivos, diferenciando cada una de ellas en cuanto a su normativa.
- Definición de los cálculos de vacaciones, utilidades, prestaciones, intereses, liquidaciones directamente por el usuario, permitiéndole a éste cambiar la forma de cálculo de alguno de los conceptos anteriormente mencionados, de manera sencilla y en el momento que lo necesite.
- Proveer a los usuarios del sistema de varios mecanismos de seguridad para el control de acceso a la data almacenada en las bases de datos.
- Proveer información histórica de gastos incurridos.

II.18 Sistema Administrador de Base de Datos

Los Sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones. Está compuesto de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

II.19 SQL Server

SQL Server es un sistema administrador de Base de Datos Relacional, Cliente–Servidor, que permite una mayor escalabilidad de explorar objetos de Base de Datos y la integración de secuencias de los comandos en la base de Datos OLTP y OLAP. "Puede utilizar SQL Server para realizar procesamiento de transacciones, almacenar y analizar datos, y generar nuevas aplicaciones de base de datos." (Jackson, 2013).

II.20 LDAP

Protocolo ligero de acceso a directorios (en inglés, Lightweight Directory Access Protocol, LDAP) es un conjunto de protocolos abiertos usados para acceder información guardada centralmente a través de la red. LDAP organiza la información en un modo jerárquico usando directorios. Estos directorios pueden almacenar una gran variedad de información y se pueden incluso usar de forma similar al Servicio de información de red, permitiendo que cualquiera pueda acceder a su cuenta desde cualquier máquina en la red acreditada con LDAP.

"LDAP es un sistema cliente/servidor. El servidor puede usar una variedad de bases de datos para guardar un directorio, cada uno optimizado para operaciones de lectura rápidas y en gran volumen. Cuando una aplicación cliente LDAP se conecta a un servidor LDAP puede, o bien consultar un directorio, o intentar modificarlo. En el evento de una consulta, el servidor, puede contestarla localmente o puede dirigir la consulta a un servidor LDAP que tenga la respuesta.

Si la aplicación cliente está intentando modificar información en un directorio LDAP, el servidor verifica que el usuario tiene permiso para efectuar el cambio y después añade o actualiza la información." (Hat, 2005).

II.20.1 Ventajas de LDAP

- La mayor ventaja de LDAP es que se puede consolidar información para toda una organización dentro de un repositorio central.

- LDAP también soporta un número de bases de datos back-end en las que se guardan directorios.

CAPÍTULO III – MARCO METODOLÓGICO

Este capítulo tiene como objetivo describir la metodología implementada en el desarrollo de la aplicación web para la gestión del personal docente por parte de las unidades académicas de la Universidad Católica Andrés Bello (UCAB), dicha metodología es SCRUM, la misma será explicada a continuación:

III.1 SCRUM

"Scrum es una metodología de desarrollo ágil, definiendo la forma de abordar un proceso de desarrollo de software de forma ágil y liviana a través de la descripción de un conjunto de roles, componentes y organización de la actividad diaria" (Vass Digital, 2012).

Scrum es una metodología en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum, el proceso de un proyecto se ejecuta en bloques cortos y fijos (iteraciones de un mes natural y hasta dos semanas, si así se necesita). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

III.1.1 Características de Scrum

- Es una metodología ágil.
- Es liviano, iterativo e incremental.
- Se realizan entregas funcionales.
- Simplicidad.
- Motivación y responsabilidad.
- Adaptativo y flexible.
- Mitigación de riesgos.

III.2 Proceso de Scrum

El desarrollo se realiza de forma iterativa e incremental. Cada iteración, denominada Sprint, tiene una duración pre establecida de entre una y cuatro semanas, obteniendo como resultado una versión del software con nuevos requerimientos listos para ser usados.

"En cada nuevo Sprint, se va ajustando la funcionalidad ya construida y se añaden nuevos requerimientos priorizando siempre aquellos que aporten mayor valor de negocio" (Softeng, 2015).

- **Product Backlog:** Conjunto de requisitos denominados historias descritos en un lenguaje no técnico y priorizados por valor de negocio, o lo que es lo mismo, por retorno de inversión considerando su beneficio y coste. "Los requisitos y prioridades se revisan y ajustan durante el curso del proyecto a intervalos regulares" (Softeng, 2015).

- **Planificación del Sprint**: Reunión durante la cual el Product Owner presenta las historias del backlog por orden de prioridad. "El equipo determina la cantidad de tiempo que puede comprometerse a completar en ese sprint, para en una segunda parte de la reunión, decidir y organizar cómo lo va a conseguir" (Softeng, 2015).

- **Sprint**: Iteración de duración prefijada durante la cual el equipo trabaja para convertir las historias del Product Backlog a las que se ha comprometido, en una nueva versión del software totalmente operativo (Softeng, 2015).

- **Scrum Diario**: "Reunión diaria de máximo 15 minutos en la cual el equipo se sincroniza para trabajar de forma coordinada" (Softeng, 2015). Cada miembro comenta que hizo el día anterior, que hará hoy y si hay impedimentos.

- **Revisión del Sprint**: Lista de las tareas necesarias para llevar a cabo las historias del sprint.

- **Demo y Retrospectiva del Sprint**: "Reunión que se celebra al final del sprint y en la que el equipo presenta las historias conseguidas mediante una demostración del producto. Posteriormente, en la retrospectiva, el equipo analiza qué se hizo bien, qué procesos serían mejorables y discuten acerca de cómo perfeccionarlos" (Softeng, 2015).

III.3 Roles de Scrum

- **Product Owner**: Es la persona que habla por el cliente, el mismo asegura que se cumpla las expectativas. Es el responsable del proyecto. Identifica los elementos del Product Backlog.

- **Líder (Scrum Master)**: Persona que lidera al equipo guiándolo para que cumpla las reglas y procesos de la metodología. Gestiona la reducción de impedimentos del proyecto y trabaja con el Product Owner para maximizar el ROI. Se conoce como ROI (RETURN ON INVESTMENT) o retorno de la inversión, a la razón financiera que compara el beneficio o la utilidad obtenida en relación a la inversión realizada.

- **Dueño del producto:** Representante de los accionistas y clientes que usan el software. Se focaliza en la parte de negocio y el responsable del ROI del proyecto (entregar un valor superior al dinero invertido). "Traslada la visión del proyecto al equipo, formaliza las prestaciones en historias a incorporar en el Product Backlog y las re-prioriza de forma regular" (Softeng, 2015).

- **Equipo de desarrollo:** Grupo de profesionales con los conocimientos técnicos necesarios y que desarrollan el proyecto de manera conjunta llevando a las que se comprometen al inicio de cada sprint.

III.4 Artefactos de Scrum

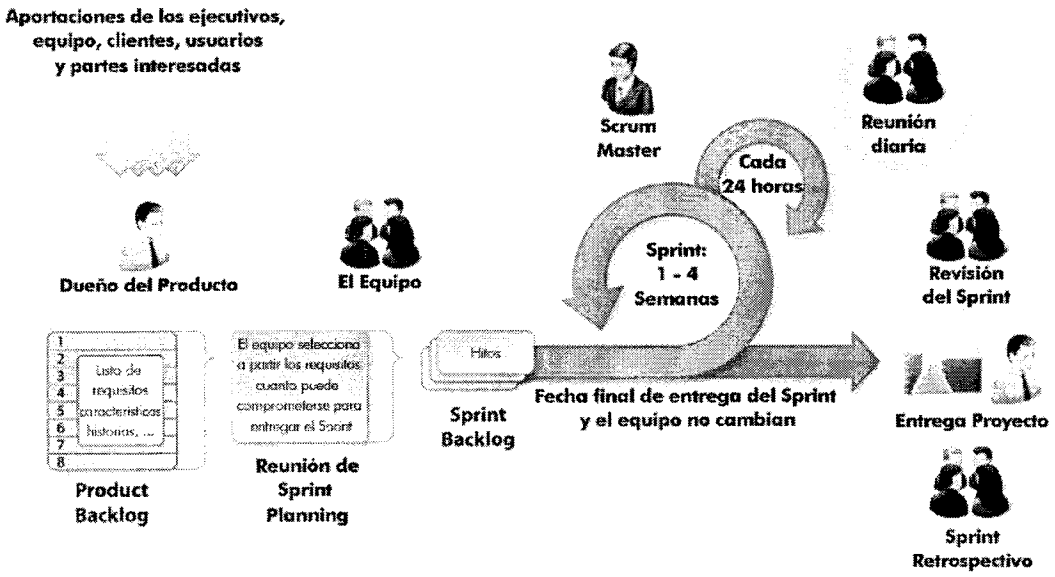
- **Pila de Producto**: Es una lista ordenada de todo lo que podría ser necesario para el producto, y la única fuente de requerimientos para cualquier cambio a realizarse. "Contiene todas las características funcionales, requerimientos, mejoras y correcciones de! producto, y otros atributos como descripción, ordenación y estimación" (Ken Schwaber y Jeff Sutherland, 2013).

- **Pila de Sprint:** Es el conjunto de elementos de la Pila de Producto seleccionados para el Sprint, más un plan para entregar el Incremento y conseguir el objetivo. "Define el trabajo que se llevará a cabo para convertir elementos de la pila en un incremento hecho" (Ken Schwaber y Jeff Sutherland, 2013).

- **Incremento**: Parte del producto desarrollada en un sprint, en condiciones de ser usada. Resultado de cada sprint.

En la Figura 3 se puede observar el esquema de la metodología Scrum.

Figura 3: Reglas de SCRUM



(Isla Visual, 2014)

III.5 Justificación de la metodología

Se escogió como metodología de trabajo Scrum por todos sus beneficios.

Entre ellos, dado que es una metodología ágil y se trabaja colaborativamente en equipo, se agilizan los tiempos y la aceptación de cada entrega, ya que éstas han sido revisadas por los involucrados por parte de la DTI.

Las reuniones cortas diarias (Scrum diario), las revisiones y retrospectiva presentando un demo con las nuevas funcionalidades fueron las otras características que formaron parte en la toma de decisión de Scrum como metodología de trabajo.

En el Scrum diario se reúne todo el equipo y se planifica el trabajo, con esta reunión también se trata de encontrar las funcionalidades que estén deficientes y en cómo se pueden mejorar. Al finalizar cada sprint se hace entregan entregas funcionales y se planifican las pautas para continuar el desarrollo.

Estas reuniones son de gran importancia ya que de encontrar alguna deficiencia o alguna duda en los requerimientos, se planificaba una reunión directamente con la DTI y los distintos docentes que al final, serán los usuarios de la aplicación.

Dada la magnitud del proyecto y por la cantidad de docentes que serán los usuarios finales, cada conclusión obtenida de las reuniones es importante para lograr la satisfacción de estos. Scrum es adaptable al cambio, esto quiere decir que aparte de seguir un plan de trabajo, se permite la mejora constante del producto, se pueden realizar modificaciones cuando sean solicitadas y no al momento de la entrega final, siendo una metodología adaptativa. Este aspecto es fundamental, aunque existe una visión clara del alcance del sistema, es muy probable que de las reuniones con los usuarios surjan funcionalidades adicionales que pudieran ser incluidas en el proyecto.

III.6 Sprint Backlog

A continuación se describe la dinámica de trabajo implementada en el desarrollo de la solución. Se listará por cada Sprint el conjunto de actividades realizadas.

En la tabla 1 se puede observar el Sprint Backlog del proyecto.

Tabla 1. Sprint Backlog

Sprint	Actividades	Entregables	Aporte
0	• Levantamiento de información del proceso de gestión de inasistencias de la UCAB. • Levantamiento de información sobre tablas del sistema BANNER que contiene información sobre profesores. • Definición de estructuras del sistema de ofertas y del módulo de administración. • Diseño e implementación de base de datos "Ucabas" e "Intranet Profesores". • Selección de herramientas de desarrollo.	• Product Backlog del proyecto. • Diagrama entidad relación. • Base de datos funcional, en SQL Server.	
1	• Rediseñar el proceso de control de asistencia de los profesores. • Diseñar workflow para el proceso de solicitud de remoción de inasistencia.	• Diagrama de flujo del proceso de control de asistencia de los profesores.	• 9
2	• Definir la estructura del API de servicios web que será usada por la aplicación. • Integración de la aplicación con servicios web.	• Estructura del API de los servicios web. • Ejecución de pruebas de servicios web (Banner).	• 6 • 7 • 8
3	• Diseño de las interfaces de usuario. • Login de Usuario. • Desarrollo de servicios web para módulo de perfil. • Desarrollo del módulo de perfil	• Prototipo de las interfaces de Usuario. • Módulo de autenticación. • Servicios web para módulo de perfil (Banner, Picasso). • Módulo de perfil.	• 1 • 3

4	• Desarrollo de nuevo requerimiento del módulo de perfil. • Desarrollo de servicios web para módulo de datos curriculares. • Desarrollo del módulo de datos curriculares.	• Servicios web para módulo de datos curriculares (Picasso). • Módulo de datos curriculares. • Módulo de perfil con nueva funcionalidad.	• 3
5	• Desarrollo de servicios web integrados con el sistema Lenel. • Desarrollo de módulo de inasistencias	• Servicios web para módulo de inasistencias (Banner, Lenel). • Módulo de inasistencias.	• 5 • 7
6	• Rediseñar el proceso de proceso de búsqueda de profesores para las escuelas. • Desarrollo de servicios web para módulo de ofertas. • Desarrollo de módulo de ofertas.	• Diagrama de flujo del proceso de ofertas. • Servicios web de ofertas.(Banner) • Módulo de ofertas.	• 4 • 10 • 6
7	• Desarrollo de módulo de administración.	• Módulo de administración.	• 2

Fuente: Elaboración propia.

CAPÍTULO IV - DESARROLLO

A continuación, se describen las actividades realizadas por sprint.

IV.1 Sprint 0

IV.1.1 Inicio

El objetivo principal de este sprint es la transferencia y el análisis de la información entre los involucrados del equipo Scrum. Se definen las próximas reuniones, reconocimiento del equipo, roles de cada integrante, entre otros. Adicionalmente, se planificó una reunión con las personas encargadas del manejo de inasistencias para conocer el proceso y tener un punto de arranque sobre el tema. También se planificaron reuniones con distintos directores de escuela y profesores para obtener sugerencias sobre las funcionalidades del proyecto.

La Dirección de Tecnología de Información (DTI) de la UCAB mantiene como estándar desarrollar las aplicaciones utilizando la plataforma Microsoft .NET. Se tomó la decisión de utilizar SQL Server 2012 como manejador de base de datos, y la versión de .NET framework 4.5 está incluida en Visual Studio .NET 2015. Se seleccionó el marco de trabajo ASP.NET MVC 4.0, el cual permite desarrollar usando el patrón de arquitectura modelo vista controlador para la solución.

IV.1.2 Product Backlog

En la tabla 2 se puede observar el Product Backlog del proyecto.

Tabla 2. Product Backlog

PRODUCT BACKLOG
Aplicación web que debe estar disponible desde cualquier navegador
El ingreso a la aplicación web debe realizarse únicamente por personal académico y administrativo de la UCAB registrados en el sistema de LDAP, mediante usuario y contraseña.
El diseño de la aplicación web debe seguir los estándares de las aplicaciones web desarrolladas por la DTI.
El acceso a los distintos servidores desde la aplicación web será a través de un API de servicios web.
La aplicación web permitirá la creación y edición de ofertas de trabajo por parte de las escuelas destinadas a los profesores dentro de la UCAB.
La aplicación web permitirá la búsqueda y consulta de datos de los profesores que han formado parte de la UCAB.
La aplicación web permitirá a los profesores la consulta de sus inasistencias y la solicitud de remoción en caso de encontrar algún error.
La aplicación web contará con un módulo administrativo que regulará su uso para los distintos niveles de usuarios.

Fuente: Elaboración propia.

Para el desarrollo de este proyecto se definieron los roles y sus responsables:

- **Product owner**: Roquefelix Benal (Jefe de la unidad de desarrollo).
- **Scrum master**: Roquefelix Benal (Jefe de la unidad de desarrollo).
- **Scrum team**: Eugenio Carrasco (Coordinador de servicio web).

  Víctor Bereciartu (Responsable de pruebas).

  Eduardo Gómez (Desarrollador).

IV.1.3 Creación del Modelo Entidad Relación

A partir de levantamiento de información surgió la necesidad de la creación de dos modelos Entidad Relación. El primero debe presentar las inasistencias, solicitudes de remoción, ofertas y datos curriculares de profesores.

El segundo modelo presenta los roles y permisos de los distintos usuarios, según su cargo, para la aplicación. Este modelo será utilizado para el manejo de permisos de los usuarios de las distintas aplicaciones desarrolladas en la DTI.

En las Figuras 4 y 5 se puede observar los diagramas Entidad de Relacional de las bases de Datos "Ucabas" e "Intranet Profesores" respectivamente.

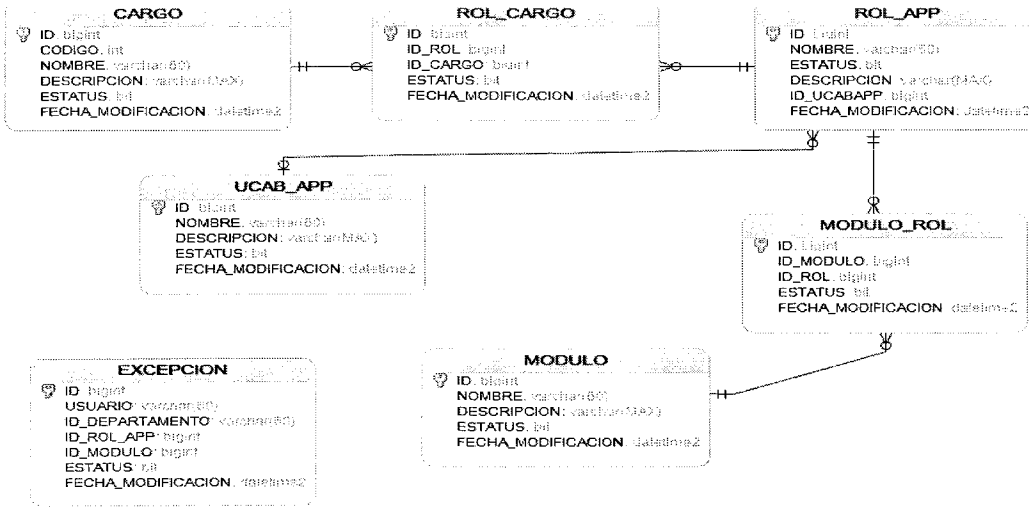


Figura 4. Diagrama Entidad Relación "UCABAS".
Fuente: Elaboración propia.

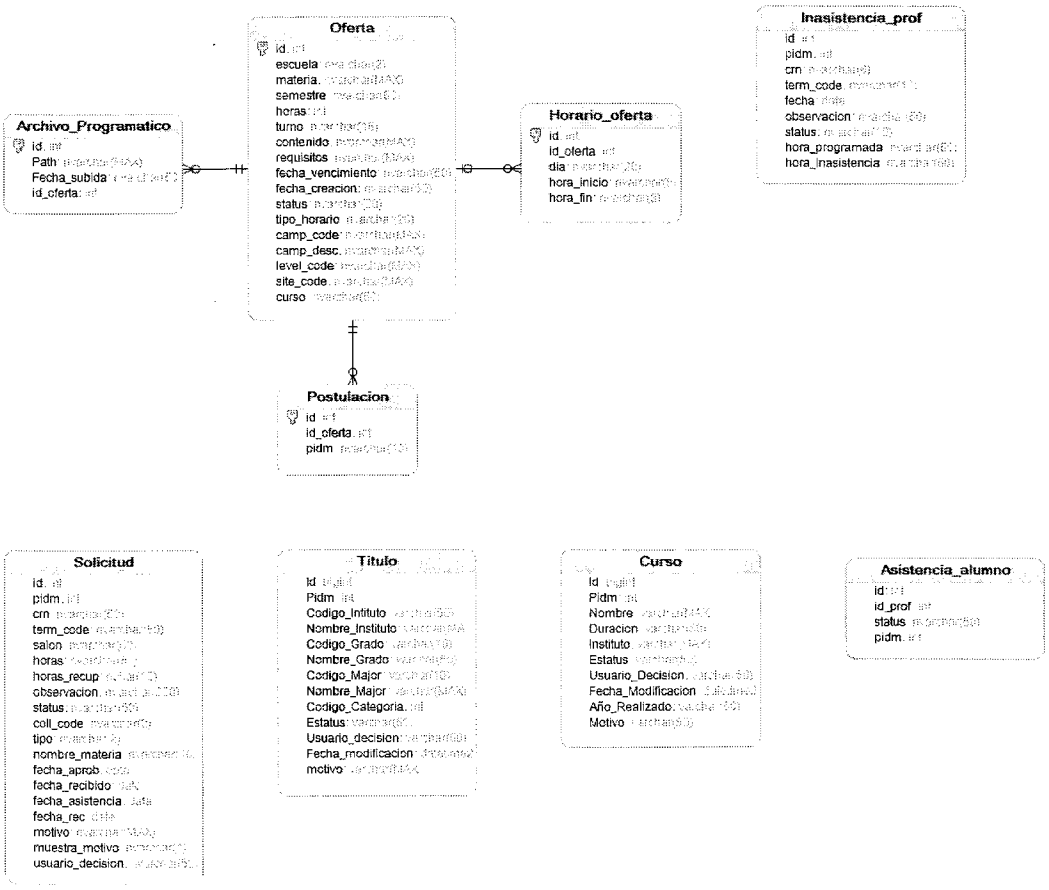


Figura 5. Diagrama Entidad Relación "Intranet_Profesores".
Fuente: Elaboración propia.

Partiendo de los modelos Entidad Relación, se construyeron las bases de datos respetando todas las restricciones de integridad establecidas para lograr la consistencia y seguridad de la información (Ver Apéndice A8 Figura 18-A).

IV.2 Sprint 1

Para realizar el rediseño del proceso de control de asistencia de los profesores, se analizó cómo se realiza el proceso actual. Para ello, se llevó a cabo una serie de reuniones con el personal que maneja dicho trabajo.

IV.2.1 Rediseñar el proceso de control de asistencia de los profesores.

Actualmente el personal encargado debe iniciar sesión en una aplicación de escritorio conectada a Banner, donde selecciona el día, período, NRC e indicador para cargar la inasistencia. Este proceso debe hacerse por cada sección que tenga inasistencia en el día. Por defecto, se asume que si no está registrada una inasistencia para una fecha determinada, quiere decir que el profesor asistió a su cátedra.

Entre las debilidades que se encontraron, se destacan las siguientes:

- El personal debe tener una cuenta registrada en Banner.
- Aplicación poco intuitiva para un usuario final.

- Se debe introducir todos los datos de la inasistencia (Día, Período, NRC, Indicador).
- Se corre el riesgo de cargar una inasistencia que ha sido firmada a través de la aplicación "UCABdroide Profesores".
- Se cargan una a una las inasistencias del día del profesor.
- Existe la posibilidad de que ocurran errores en la carga de inasistencia.

Entre las mejoras que pueden plantearse están:

- El usuario iniciará sesión en la aplicación web con su usuario y clave LDAP.
- Búsqueda del profesor a través de su cédula, nombre o correo UCAB.
- Se puede cargar inasistencias a distintos NRC de un mismo profesor.
- Se reduce el tiempo empleado para la carga de inasistencias.
- Se reduce la posibilidad de errores en la carga de inasistencia.

IV.2.2 Diseñar workflow para el proceso de solicitud de remoción de inasistencia.

Existen dos escenarios para solicitar una remoción de inasistencia. Pudo haber sido porque efectivamente el profesor no dictó la clase, o bien, porque el profesor olvidó firmar su asistencia. En el primer caso, el docente debe solicitar la recuperación de las horas que perdió y solicitar la remoción desde la aplicación web.

Sin embargo, el segundo escenario solo requiere la solicitud de remoción desde la aplicación web por parte del profesor.

En la Figura 6 se puede ver, a través de la comparación de diagramas de flujo, la diferencia entre el proceso actual y el nuevo proceso para la remoción de inasistencias.

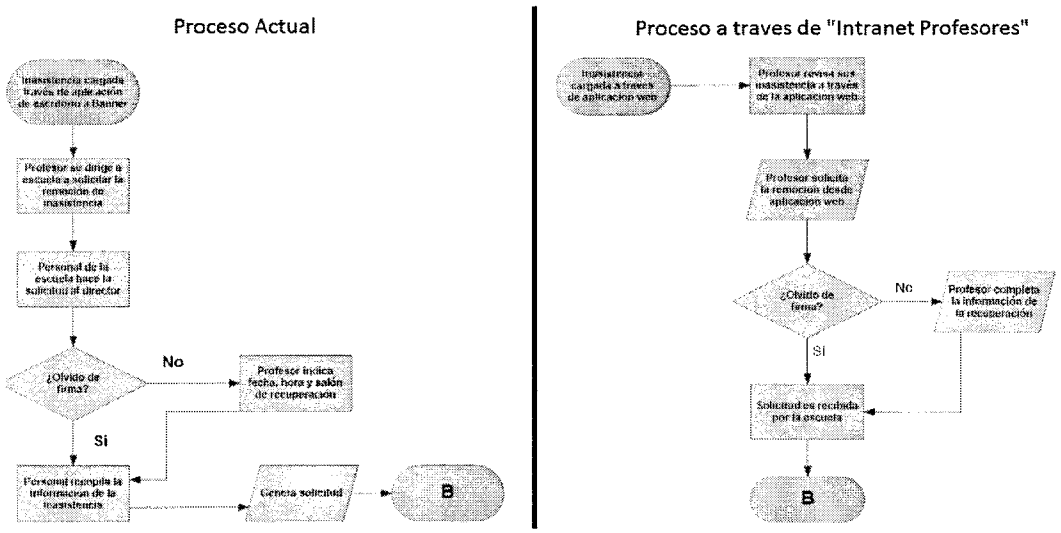


Figura 6. Comparación de diagramas de flujo para "Remoción de Inasistencia". Fuente: Elaboración propia.

IV.3 Sprint 2

IV.3.1 Definir la estructura del API de servicios web que será utilizada por la aplicación

Se realizaron varias reuniones con el equipo de desarrollo de software de la DTI para obtener una estructura apropiada y estándar para los nuevos servicios web,

ya que existen servicios desarrollados que son consumidos por la aplicación UCABdroide.

Los servicios web que utiliza Ucabdroide son servicios Restful desarrollados en C# bajo la arquitectura MVC4. Estos fueron los estándares que se utilizaron para desarrollar el API de la aplicación web. Los servicios se comunicarán con las bases de datos de: Banner 8, Picasso, LDAP y Lenel, Intranet_Profesores.

- Servidor de Banner 8: Contiene la información académica y administrativa de la universidad (Profesores, Estudiantes, Materias, horarios de clases, entre otros).
- Picasso: Contiene la información sobre recursos humanos de la universidad.
- Servidor LDAP: Gestiona la cuentas de usuario de la UCAB, contiene la información de nombre de usuario y contraseña.
- Servidor Lenel: Sistema de control de acceso de la universidad, contiene la información del uso del carnet.
- Intranet_Profesores: Contiene información sobre las solicitudes de remoción de inasistencias y ofertas académicas.

Por otra parte, se define establecer un mecanismo de seguridad para el consumo del API. Este mecanismo está basado en que la API sólo podrá ser consumida desde la dirección IP donde se encuentra alojada la aplicación. Los servicios web no podrán ser consumidos desde ninguna otra dirección.

Luego de las distintas reuniones se llegó a la conclusión que la mejor estructura para el API es aquella que cuente con las siguientes condiciones:

- El único acceso a los servidores será a través del API de servicios web.
- Los únicos usuarios que podrán iniciar sesión son los profesores de la UCAB.
- Se usará de la dirección IP como medida de seguridad de los datos. Esta dirección IP solo será conocida por el API de servicios Web.
- Los servicios serán publicados en un servidor IIS (Internet Information Services) para el consumo de diversas aplicaciones.

En la Figura 8 puede verse la arquitectura del proyecto, incluyendo la comunicación con la API de servicios web.

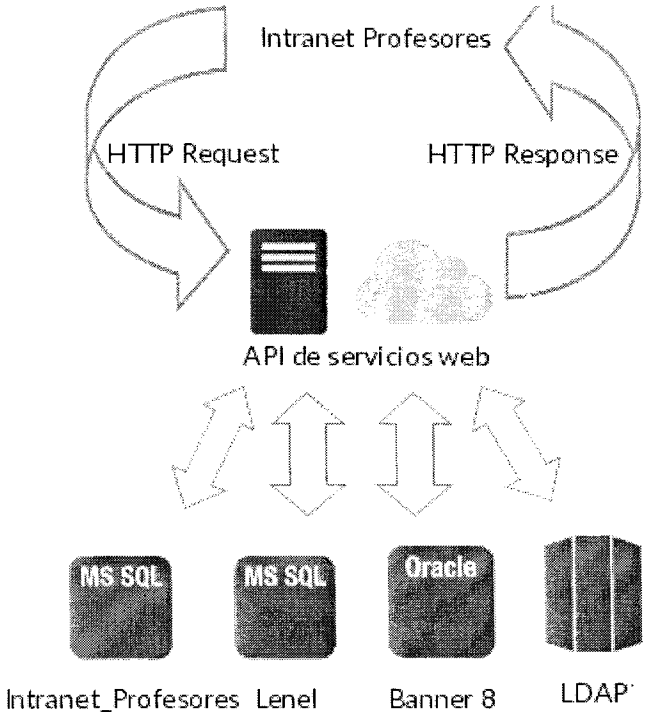


Figura 8. Estructura del proyecto.

Fuente: Elaboración propia.

IV.3.2 Integración de la aplicación con API de servicios web.

Para la integración de la API, se diseñó una clase en la aplicación llamada "ConsumeWS" en la que se encuentra cada llamada a los distintos web services que utiliza la aplicación, cada llamada está conformada por los parámetros de conexión hacia el API de servicios web (ip, nombre de función, JSON que recibe la función en el servidor, entre otros).

Luego de la creación de la clase que contiene todas las llamadas a los web services, se realizaron un conjunto de pruebas para comprobar la correcta integración de la aplicación con la API, estas pruebas fueron realizadas realizando peticiones a la función "getPersonaFull" que se encontraba previamente desarrollada. La respuesta de los servicios web siempre mantendrá el formato JSON.

En el Figura 9 puede verse la respuesta del servidor durante las pruebas a los servicios web.

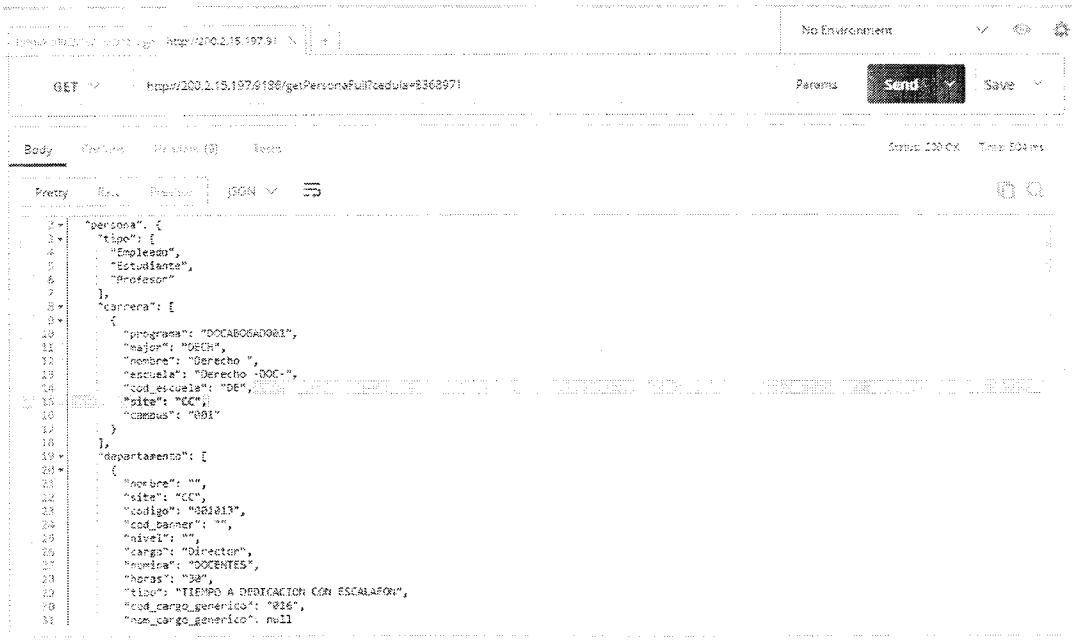


Figura 9. Respuesta de servicio web.

Fuente: Elaboración propia.

IV.4 Sprint 3

IV.4.1 Diseño de las interfaces de usuarios

A partir de la información levantada en las distintas reuniones, y conocer cómo se maneja actualmente la gestión de inasistencias, se realizó la primera visión de las interfaces de usuarios que servirán como guía al comenzar con la programación de la aplicación.

En la Figura 10 se muestran algunas de las pantallas realizadas utilizando la herramienta Balsamiq Mockups. Se desarrollaron pantallas de cada uno de los módulos que conforman la aplicación (I.2, Capítulo I).

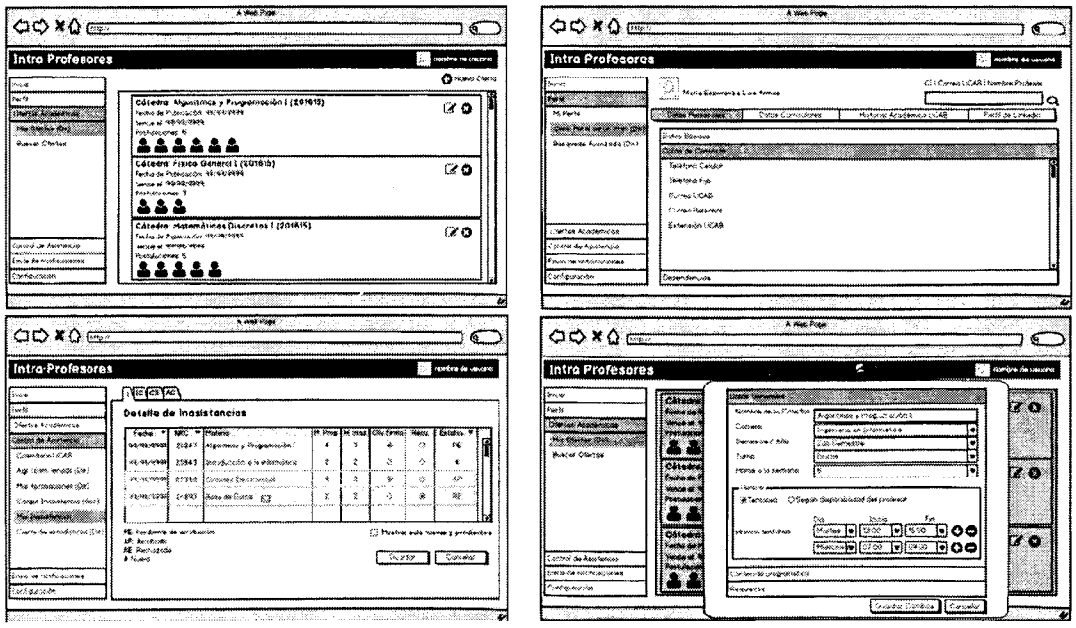


Figura 10. Interfaces de usuario.

Fuente: Elaboración propia.

IV.4.2 Login de Usuario

El login de usuario es una pantalla que permite la autenticación del usuario para ingresar a la aplicación web. El usuario debe ingresar el nombre de usuario UCAB y contraseña, la aplicación utiliza el Protocolo Simplificado de Acceso a Directorios (LDAP) para la validación de los datos y permite o no el acceso a la aplicación.

En la Figura 11 se muestra la pantalla de inicio de sesión para todos los usuarios de la aplicación.

Figura 11. Login de usuario.

Fuente: Elaboración propia.

IV.4.3 Desarrollo de servicios web para el módulo de perfil

Se desarrollaron los distintos métodos en la API de servicios web que contienen las consultas a Banner y Picasso para obtener la información de los profesores.

IV.4.4 Desarrollo de módulo de Perfil

El módulo de Perfil consta de dos funcionalidades, las cuales son: Mi Perfil, Consultar Perfil.

Mi Perfil: El usuario podrá ver sus datos personales, curriculares y académicos.

Datos Personales: Se podrán consultar los datos básicos, datos de contacto y las dependencias actuales dentro de la UCAB.

Datos Curriculares: El usuario podrá consultar sus títulos académicos, cursos o certificaciones que se encuentren registradas en Recursos Humanos.

Historial Académico: Se mostrará una tabla con toda la información de los cursos dictados por el profesor dentro de la UCAB. Esta tabla puede ser filtrada en los campos de texto ubicados en el título de cada columna.

Consultar Perfil: El usuario podrá introducir en el campo de texto la cédula, nombre, apellido o correo electrónico de la UCAB. Al hacer click en el botón "Buscar" se mostrará un modal que contiene una tabla con los resultados (Ver Apéndice A1 Figura 11-A). Estos resultados pueden ser filtrados en los campos de

texto ubicados en el título de cada columna. Al seleccionar el profesor el cual desea consultar se mostrarán los "Datos Personales", "Datos Curriculares" e "Historial Académico".

IV.5 Sprint 4

IV.5.1 Desarrollo de servicios web para el módulo de datos curriculares

Se desarrollaron los distintos métodos en la API de servicios web que contienen las consultas a Picasso e "Intranet_Profesores" para obtener la información curricular de los profesores.

IV.5.2 Desarrollo de módulo de datos curriculares

Luego de distintas reuniones con profesores y directores de escuela surgió la necesidad de implementar una nueva funcionalidad.

Los profesores podrán agregar o eliminar nuevos datos curriculares (Títulos y/o Cursos y Certificaciones) los cuales serán aprobados o rechazados por los directores de escuela a través de la aplicación web.

Para implementar esta nueva funcionalidad se decidió, en primer lugar, modificar la base de datos "Intranet_Profesores" para almacenar en ella los nuevos datos curriculares.

En la Figura 12 se muestran las opciones del módulo "Perfil", incluyendo "Datos Curriculares por Aprobar".

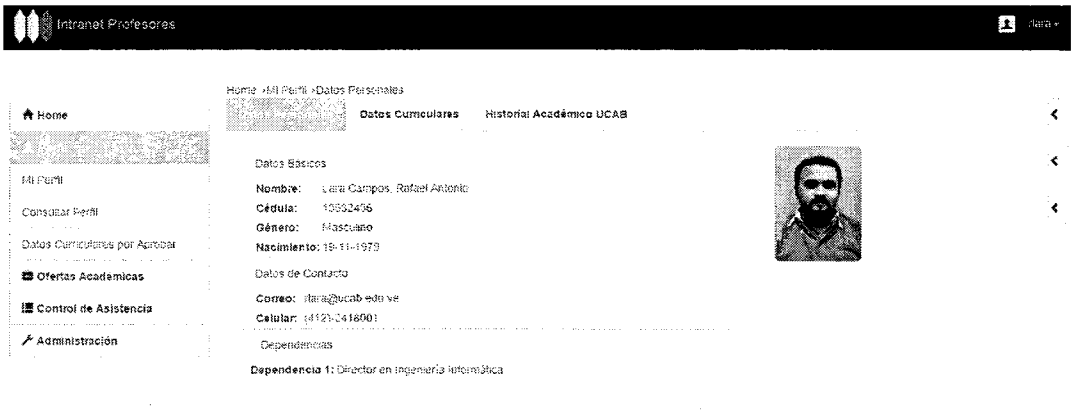


Figura 12. Módulo "Perfil".

Fuente: Elaboración propia.

En la sección de datos curriculares se agregó un botón que permite a los usuarios agregar nuevo título académico, el titulo está conformado por: grado, mención e institución. Estas tres opciones son requeridas, el usuario las debe seleccionar de tres listas de opciones desplegables (Ver Apéndice A2 Figura 12-A).

También se agregó un botón que permite agregar un nuevo curso o certificación, éstos están conformados por: nombre, duración, instituto y año. El usuario debe introducir los cuatro campos para poder agregar un nuevo curso o certificación.

Luego de agregado un título o curso estos cambian a estatus "PE" (Pendiente). Mientras se encuentren en este estatus el usuario los puede eliminar. Los cursos o títulos que ya se encuentran cargados en recursos humanos aparecerán con estatus "AP" (Aprobado) y no podrán ser eliminados. Los cursos que se encuentren con estatus "RE" (Rechazado) podrán ser eliminados.

Al módulo de Perfil se le agregó una nueva funcionalidad denominada "Datos Curriculares por Aprobar". Esta opción sólo estará disponible para los directores de escuela, los cuales podrán aprobar o rechazar las solicitudes de títulos o cursos realizadas por los profesores que pertenezcan a la escuela de la cual es director. Se tiene previsto, a corto plazo, la creación de un rol "Director RR.HH." el cual será el único que podrá tomar decisiones sobres los nuevos datos curriculares y cargar estos directamente en la base de datos de Recursos Humanos, a través de un servicio web que se comunica con la base de datos Picasso.

Esta funcionalidad incluye una opción de historial, donde se podrá conocer la información sobre quien realizó alguna decisión sobre alguna solicitud de título o curso, además, se permite el cambio de estatus para las decisiones previamente tomadas.

IV.6 Sprint 5

IV.6.1 Servicios web para módulo de inasistencias

Se desarrollaron los distintos métodos en la API de servicios web que contienen las consultas a Lenel y Banner para obtener la información de control de acceso de un profesor en la universidad.

IV.6.2 Desarrollo de módulo de inasistencias

Este módulo se caracteriza por optimizar el proceso de inasistencias, desde la carga de inasistencias, hasta la visualización de inasistencias por parte de profesores.

El módulo consta de 3 funcionalidades: Cargar inasistencias, Mis inasistencias, Mis Aprobaciones (Ver Apéndice A3 - Figura 13-A).

Cargar Inasistencias: En esta opción el usuario deberá introducir en el campo de texto la cédula, nombre, apellido o correo electrónico de la UCAB del profesor al cual desea cargar la inasistencia, se mostrará un modal que contiene una tabla con los resultados. Esta tabla puede ser filtrada en los campos de texto ubicados en el título de cada columna.

Al seleccionar el usuario se mostrará una tabla que contiene la información de las materias que está dictando el profesor en el periodo actual. Para cargar la inasistencia, el usuario debe hacer click en el ícono del calendario y seleccionar la fecha de la inasistencia. El usuario no puede elegir fechas futuras ni días distintos a los que pertenezca esa materia. Luego debe completar las horas de inasistencia, estas horas no pueden ser mayores a las horas programadas y no pueden ser ni cero ni vacíos.

En la Figura 13 se muestran las materias actuales de un profesor, a las cuales se les puede agregar una inasistencia.

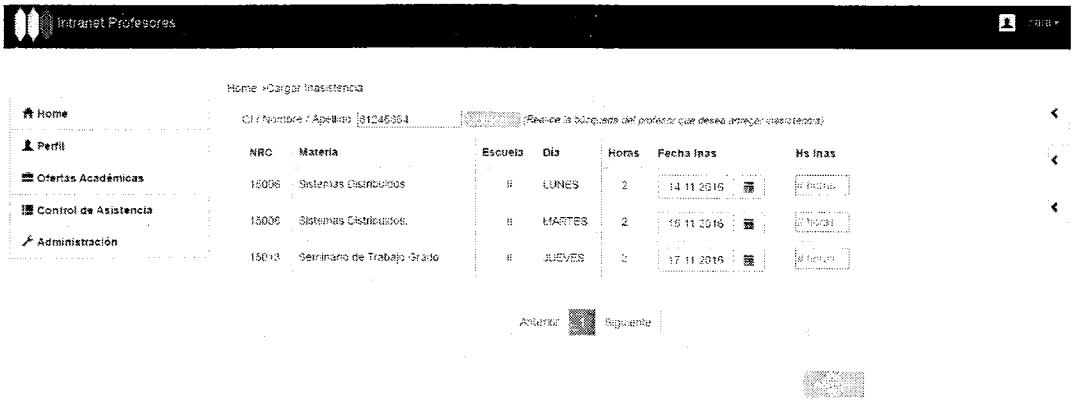


Figura 13. Materias a cargar inasistencia.

Fuente: Elaboración propia.

Para cargar la inasistencia debe hacer click en el botón "Agregar". El usuario puede enviar varias inasistencias para distintas materias. Se mostrará una advertencia con los resultados: Agregado con éxito, No agregado, No agregada porque existe una inasistencia ya cargada para ese día.

Mis Inasistencias: Al hacer click en esta opción se mostrará una tabla con las inasistencias del usuario en el periodo actual (Ver Apéndice A4 - Figura 14-A).

Las inasistencias tienen cuatro estatus: AP: Aprobado, PE: Pendiente, RE: Rechazado, #: Nueva.

El usuario tiene la opción de solicitar la remoción de inasistencia mediante dos opciones:

- Haciendo click en el botón "Olvido de firma" se mostrará una alerta de confirmación si desea solicitar la remoción por olvido de firma, al aceptar se mostrará otra alerta con el resultado. De ser exitosa la solicitud, la inasistencia cambiará el estatus de nueva "#" a pendiente "PE".

- Haciendo click en el botón "Recuperación" se mostrará un modal en el cual el profesor debe completar con la información de la recuperación, no debe dejar el campo "horas" ni "salón" en blanco. Luego debe hacer click en "Enviar", mostrará otra alerta con el resultado. De ser exitosa la solicitud, la inasistencia cambiará el estatus de nueva "#" a pendiente "PE".

Al hacer la solicitud pasarán a la escuela y serán revisadas para su aprobación o rechazo según sea el caso.

El usuario puede filtrar las ofertas por "Mostrar solo nuevas y pendientes" haciendo click en el "check" ubicado en la parte superior derecha.

Si el usuario desea eliminar una solicitud cargada que se encuentra con estatus "PE", debe seleccionar la solicitud en la tabla y se habilitará el botón "Eliminar solicitud", ubicado en la parte inferior izquierda. Se mostrará una alerta de confirmación, de aceptar se mostrará una alerta con el resultado, de ser exitoso cambiara el estatus de "PE" a "#".

La escuela puede enviar comentarios sobre las decisiones sobre las inasistencias, estos comentarios pueden ser vistos por el usuario haciendo click en el ícono de la carta. La escuela tiene la opción de enviar comentarios pero no permitir al profesor su visualización.

Mis Aprobaciones: Al hacer click en esta opción se cargarán pestañas superiores dependiendo de las escuelas en las cuales el usuario tenga permisos para aprobar o rechazar solicitudes de remoción.

Se mostrarán dos pestañas intermedias, que son los casos de remoción de inasistencia "Recuperaciones" y "Omisión de firma".

Dentro de cada pestaña intermedia se cargarán dos nuevas pestañas:

"Por aprobar": Se muestra una tabla que contiene las solicitudes de remoción realizadas por los profesores para la escuela seleccionada. Al seleccionar alguna de las solicitudes se mostrará la información del profesor, el uso de la tarjeta TAI para el día de la recuperación, y la información de la inasistencia. Para ver los detalles de la recuperación, el usuario debe hacer click en el botón con el símbolo "i" en la parte inferior derecha. Para aprobar o rechazar debe hacer click en el botón "Aprobar|Rechazar", se mostrará un modal donde debe seleccionar la decisión "Aprobar" o "Rechazar", el usuario puede escribir algún motivo y permitir o no que el profesor pueda visualizarlo. Para terminar debe hacer click en "Terminar" y se mostrará una alerta con el resultado de la solicitud.

"Historial": Se muestra una tabla con la información de las decisiones tomadas por los distintos usuarios pertenecientes a la escuela, el usuario podrá seleccionar alguna decisión y hacer click en el botón "Marcar como pendiente" el cual eliminará la decisión y la colocaría de nuevo como pendiente.

La pestaña de "Omisión de firma" mantiene las mismas funcionalidades, solamente que las solicitudes no muestran el botón de detalles porque no tienen ninguna información adicional.

En la Figura 14 se muestra un ejemplo de una solicitud de remoción de inasistencia pendiente por aprobar o rechazar.

Figura 14. Remoción de inasistencia.

Fuente: Elaboración propia.

IV.7 Sprint 6

IV.7.1 Rediseñar el proceso de búsqueda de profesores para las escuelas.

El módulo de ofertas es, por decirlo de alguna manera, el corazón del proyecto. La universidad no cuenta con un sistema similar. Luego de distintas reuniones, se consolidó la estructura del nuevo proceso de ofertas. La Figura 15 muestra el diagrama de flujo del proceso.

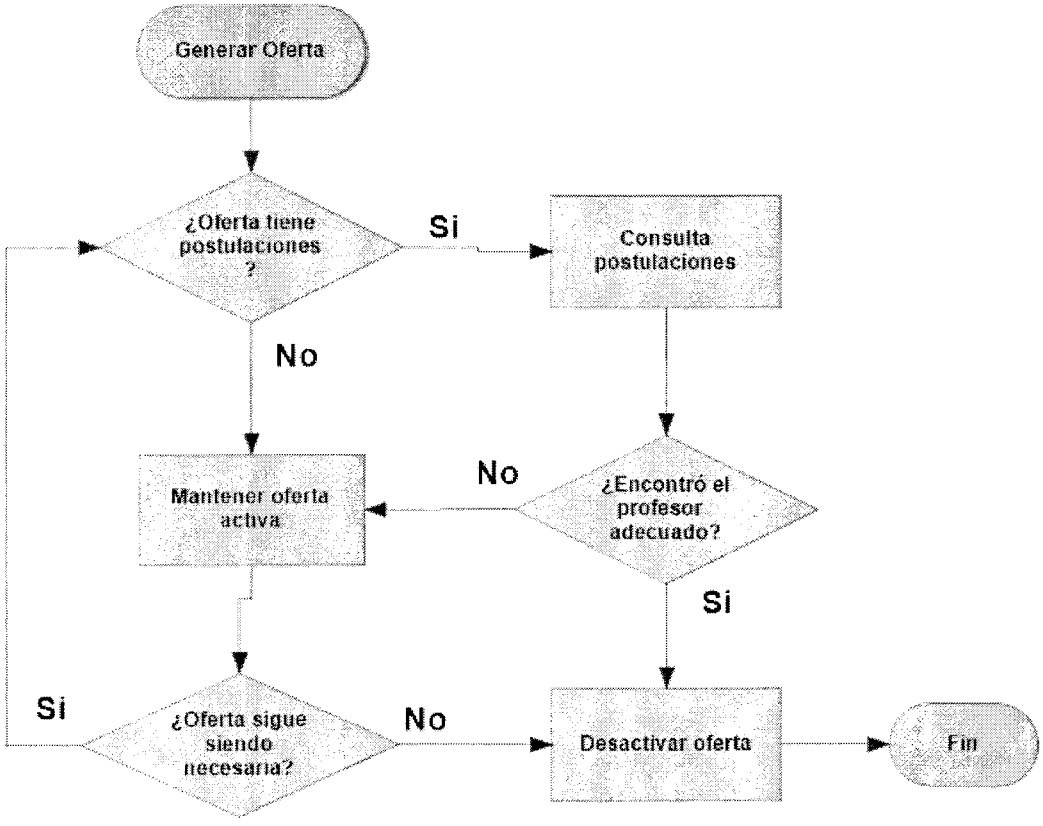


Figura 15. Diagrama de flujo del proceso de Ofertas Académicas.

Fuente: Elaboración propia.

IV.7.2 Desarrollo de servicios web para el módulo de ofertas.

Se desarrollaron los distintos métodos en la API de servicios web que contienen las consultas a "Intranet_Profesores" y Banner 8 para obtener la información de las distintas ofertas y sus profesores postulados.

IV.7.3 Desarrollo de módulo de ofertas.

El módulo consta de 3 funcionalidades: Crear oferta, Mis ofertas, Buscar ofertas.

Crear Oferta: Esta opción mostrará un modal que contiene toda la información necesaria para crear una oferta (Ver Apéndice A5 - Figura 15-A):

Escuela: No se puede modificar y es la escuela a la que pertenece el usuario.

Campus: El usuario debe seleccionar de la lista el campus al cual va dirigida la oferta, solo se muestran los campus donde existe la escuela. La lista de campus se despliega al hacer click en el ícono de la flecha a la derecha del campo, el usuario también puede escribir letra por letra su búsqueda y el campo le mostrará las opciones válidas. Debe seleccionar una opción válida, de lo contrario se genera un error.

Nombre de la materia: El usuario debe seleccionar la materia para la oferta, se muestran las materias disponibles para la escuela. La lista de materias se despliega al hacer click en el ícono de la flecha a la derecha del campo, el usuario también puede escribir letra por letra su búsqueda y el campo le mostrará las

opciones válidas. Debe seleccionar una opción válida, de lo contrario se genera un error.

Semestre / año, horas a la semana: Se cargan automáticamente luego de seleccionar la materia correspondiente.

Horario:

"Tentativo": El usuario debe completar los días de clase los cuales se dictará la materia, no se podrán agregar más de cinco días de clase para la oferta. El usuario debe seleccionar: día, hora de inicio y hora fin para cada día de clase. La lista de opciones se despliega al hacer click en el ícono de la flecha a la derecha del campo, el usuario también puede escribir letra por letra su búsqueda y el campo le mostrará las opciones válidas. Debe seleccionar una opción válida, de lo contrario se genera un error. Para agregar un nuevo día de clase no debe estar vacío ningún campo.

"Disponibilidad del profesor": El usuario no debe completar horario de clases, se oculta el horario a llenar.

Contenido programático:

El usuario podrá escribir información sobre la materia en el área de texto que estará habilitada.

El usuario podrá cargar archivos que considere necesarios sobre el contenido de la materia, podrá cargar un máximo de 5 archivos y no se permitirán los archivos repetidos. Estos archivos luego podrán ser descargados por los usuarios al momento

de consultar la oferta. La Figura 16 muestra un ejemplo de los archivos cargados a una oferta al momento de ser creada.

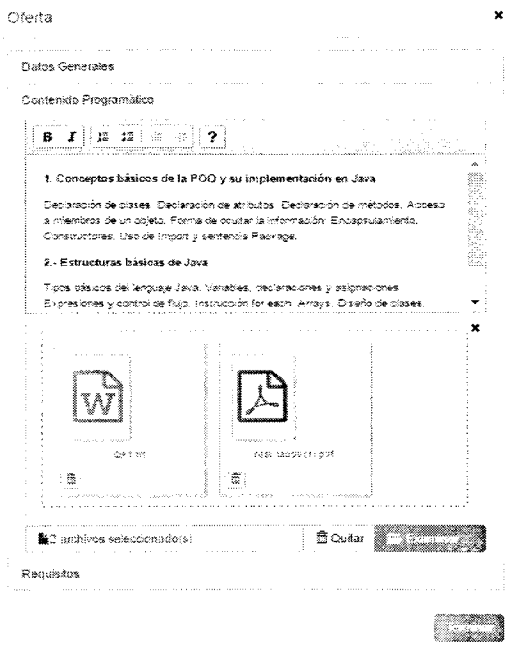


Figura 16. Archivos de contenido programático agregados a una oferta.

Fuente: Elaboración propia.

Requisitos:

El usuario podrá escribir los requisitos necesarios para la materia, el área de texto estará habilitada.

Para terminar la creación de la oferta el usuario debe hacer click en "Terminar", Deben estar todos los campos completos, de lo contrario se mostrará una advertencia y no se creará la oferta.

Mis Ofertas: Esta opción mostrará las ofertas activas correspondientes a la escuela la cual pertenece el usuario que está con sesión iniciada (Ver Apéndice A6 - Figura 16-A).

Al hacer click sobre el nombre de la materia o sobre el ícono del ojo, se mostrará un modal con la información detallada de la oferta, ninguno de los campos dentro del modal puede ser editable.

Al hacer click sobre el ícono del lápiz se mostrará un modal con la información detallada de la oferta, el usuario puede modificar el campus de la oferta, no podrá dejar este campo vacío y debe contener una opción válida, las opciones válidas serán mostradas al hacer click en el ícono de la flecha ubicado en la parte derecha del campo.

El usuario puede modificar el horario de la oferta, si selecciona horario "Tentativo" no debe dejar ningún campo vacío y no podrá agregar más de cinco clases a la semana. De igual manera debe seleccionar una opción válida, las cuales serán mostradas al hacer click en el ícono de la flecha ubicado en la derecha del campo.

El usuario puede editar el texto en "Contenido Programático" y "Requisitos".

El usuario puede eliminar los "Archivos Subidos" previamente cargados en la creación de la oferta. Estos archivos eliminados ya no tendrán relación con dicha

oferta. Para eliminarlos debe hacer click en el símbolo de la "X" a la derecha de cada archivo.

El usuario puede cargar archivos nuevos a esa oferta, el sistema no permitirá que se suban archivos que ya están previamente cargados, al seleccionar un archivo para subir, si se encuentra previamente cargado, este desaparecerá de la lista de archivos para subir.

Al realizar todos los cambios requeridos el usuario debe hacer click en el botón "Terminar" del modal, la oferta debe actualizar todos los campos.

Al hacer click sobre el ícono de la "X" a la derecha de cada oferta, esta se eliminará.

Al hacer click sobre el ícono de “Encendido”, se activará o desactivará la oferta. El ícono tendrá color verde si esta activa y roja si esta inactiva. Al desactivar una oferta, esta no podrá ser vista por ningún usuario que esté consultando ofertas disponibles.

Cada oferta listada muestra el número de postulados al lado del ícono color verde, para ver en detalle los profesores postulados debe hacer click sobre el ícono verde, se abrirá un modal con la información de los profesores. Para ver la información detallada del profesor debe hacer click en el botón "ver perfil" que se encuentra en la parte inferior de la ficha de cada postulado.

Buscar Ofertas: Al hacer click en esta opción accederá directamente a la pestaña "Más postulaciones" la cual muestra las ofertas activas con más postulantes ordenadas de mayor a menor, el usuario al hacer click sobre el ícono del ojo o sobre el nombre de la oferta, se mostrará un modal donde podrá ver la oferta, se habilita el botón "Postularse" el cual agrega al usuario a la lista de profesores postulados.

Al postularse, aparecerá un ícono de "Check" a la derecha de la oferta, esto significa que el usuario está postulado para dicha oferta.

Si intenta postularse de nuevo el sistema le mostrar una alerta y no permitirá su postulación de nuevo.

El usuario no podrá hacer click sobre el ícono verde que muestra los postulados, más si podrá ver el número total de postulados.

Al hacer click sobre la pestaña de "Más recientes" se mostraran las ofertas activas en orden cronológico, el más reciente primero. Esta pestaña mantiene las mismas funcionalidades que la pestaña "Más postulaciones".

Al hacer click sobre la pestaña de "Por escuelas" se mostrará del lado izquierdo una tabla que contendrá tres columnas: "Escuela", "Campus", "Nivel". El usuario puede filtrar desde el tope de cada columna según su criterio de búsqueda, al hacer click sobre alguna fila, al lado derecho se cargarán las ofertas creadas para dicha escuela.

Las ofertas que se muestran del lado derecho tienen las mismas características de este submódulo: ver en detalle y permite la postulación del usuario.

Al hacer click sobre la pestaña de "Mis Postulaciones" se muestran las ofertas a las cuales el usuario se ha postulado, el usuario al hacer click sobre el ícono del "ojo" o sobre el nombre de la oferta, se mostrará un modal donde podrá ver la oferta, no aparecerá el botón de postularse.

Al hacer click sobre el ícono de la "X" a la izquierda de la oferta, se eliminará la postulación del usuario a la oferta, más no la oferta.

El usuario no podrá hacer click sobre el ícono verde que muestra los postulados, más si podrá ver el número total de postulados.

La Figura 17 muestra las postulaciones de un usuario.

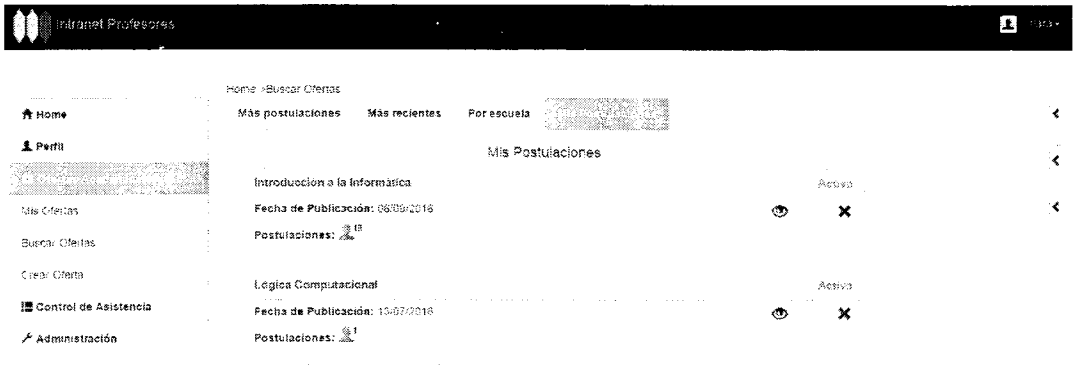


Figura 17. Postulaciones de un Profesor.

Fuente: Elaboración propia.

IV.8 Sprint 7

IV.8.1 Desarrollo de módulo de administración

En este módulo se cargarán distintas pestañas superiores y cada una tendrá el nombre de la escuela en la cual el usuario tiene permiso para editar excepciones, dentro, se mostrará una tabla la cual contendrá el nombre y la cédula del personal a tiempo completo de la escuela seleccionada. Esta tabla puede ser filtrada en los campos de texto ubicados en el título de cada columna. Al seleccionar un usuario de la tabla se mostraran los módulos a los cuales tienen acceso el usuario, los módulos que no se pueden editar pertenecen a los que el usuario maneja por defecto por su rol, los módulos que se pueden editar son aquellas excepciones que se puede permitir o negar a un usuario. (Ver Apéndice A7 - Figura 17-A).

CAPITULO V - RESULTADOS

Desarrollando las iteraciones fijadas por la metodología SCRUM y garantizando que la aplicación funcione correctamente, se logró cumplir con cada uno de los objetivos planteados.

A continuación se detallarán los resultados considerando los objetivos específicos propuestos para este Trabajo Instrumental de Grado.

V.1 Resultados según Objetivos

- **Desarrollar el módulo de autenticación (LDAP).**

En este objetivo se obtuvo como resultado la gestión de las sesiones de usuario, siguiendo los estándares del DTI, se desarrolló un inicio de sesión en el cual los usuarios (en este caso directores, profesores y personal de escuela) utilizan la clave y el usuario de sus cuentas UCAB. Este resultado se detalla en el capítulo IV en el desarrollo del Sprint 3.

- **Desarrollar el módulo administrativo.**

Luego de las distintas reuniones con el personal de la DTI se decidió crear una nuevo sistema que administra los roles de usuario y los permisos de las aplicaciones desarrolladas y por desarrollar en la DTI. Se creó una nueva base de datos que manejará la información de los módulos de las aplicaciones y sus permisos según cada rol. Este resultado se detalla en el capítulo IV en el desarrollo del Sprint 7.

- **Desarrollar el módulo de gestión de datos personales, académicos y curriculares de profesores.**

Se desarrolló una primera versión de este módulo en la cual el usuario con sesión iniciada podía ver sus datos personales, académicos y curriculares. Además se permite la búsqueda de profesores (usando nombre, cédula o correo) para consultar su información. Luego de distintas reuniones surgió un nuevo requerimiento en el cual se permite al usuario modificar sus datos curriculares y estos cambios ser aprobados o rechazados por el director de escuela. Este resultado se detalla en el capítulo IV en el desarrollo del Sprint 4.

- **Desarrollar el módulo de gestión de ofertas académicas.**

Luego de las reuniones con los directores y profesores se identificó este módulo como el más atractivo, ya que es una solución innovadora para las escuelas en la búsqueda de profesores y para éstos en la búsqueda de oportunidades laborales. Se desarrolló un módulo en el cual el director puede crear una oferta de trabajo de su escuela, la cual podrá ser consultada por los distintos profesores y si se sienten capacitados para dictar la materia, se postularían a la oferta. Así la escuela contará con una lista de profesores y seleccionarán al más indicado para el trabajo. Los profesores podrán hacer búsquedas de las ofertas: por escuela, con más postulados o más recientes. Las escuelas pueden activar o desactivar estas ofertas según lo consideren necesario, las ofertas desactivadas no serán visualizadas por los profesores. Este resultado se detalla en el capítulo IV en el desarrollo del Sprint 6.

- **Desarrollar el módulo de control de asistencias.**

Luego de realizar varias entrevistas con personal encargado en las escuelas, se analizaron los procesos involucrados para este módulo, obteniendo como resultado un diagrama de flujo de datos que describe como es el proceso de asistencia del profesor. Se diseñó e implementó una sección en la aplicación web dedicada al control de asistencias, la cual agilizará el proceso de carga de inasistencias a los profesores, además se permitirá a los profesores visualizar sus inasistencias y, en caso de ser necesario, solicitar a la escuela la remoción de ésta. El profesor, al solicitar la remoción, será aprobada o rechazada únicamente por el director de la escuela sobre la materia a la cual le fue cargada una inasistencia. Los directores podrán consultar un historial que contiene todas las decisiones tomadas sobre las solicitudes de remoción, además se les permite cambiar la decisión previamente tomada sobre una remoción. Este resultado se detalla en el capítulo IV en el desarrollo del Sprint 5.

V.2 Aporte Tecnológico

- **Integración con el sistema administrativo-académico Banner 8.**
- **Integración con el sistema de control de acceso Lenel.**
- **Integración con el sistema de gestión de personal Picasso.**

Se desarrolló e implementó una API de servicios web la cual gestiona las distintas consultas y envíos de datos sobre las distintas bases de datos. Obtenido esto, se desarrolló e implementó un algoritmo en la aplicación web que permite la

comunicación a través de la API de servicios web. La comunicación es posible a través del envío de objetos y/o arreglos de tipo JSON. Este resultado se detalla en el capítulo IV en el desarrollo del Sprint 2.

V.3 Aporte Funcional

- **Rediseñar el proceso de control de asistencia de los profesores.**

Para este punto se tomó en cuenta cómo se lleva a cabo el proceso actual de la asistencia del profesor, este proceso se detalla en el capítulo IV en el desarrollo del Sprint 1. La mejora de este proceso se encuentra en la sección de carga de inasistencias, se reduce el tiempo para cargar una o varias inasistencias al profesor y se reduce la probabilidad de cargar alguna inasistencia con errores. Para evitar que el profesor se dirija a la escuela para revisar sus inasistencias o solicitar alguna remoción, en la sección "Mis inasistencias" el profesor podrá hacer estas solicitudes y consultas desde la aplicación web. Se reduce el tiempo de los directores en tomar decisiones sobre remociones de inasistencias, ya que a través de la aplicación web contará con toda la información necesaria para la toma de estas decisiones a través de una interfaz completa e intuitiva.

- **Rediseñar el proceso de búsqueda de profesores para las escuelas.**

Luego de realizar distintos análisis sobre cuál sería el método más adecuado para las escuelas encontrar profesores, se llegó a la conclusión de crear ofertas de trabajo, que además servirán de manera inversa, es decir, que los profesores encuentren a las escuelas. Este sistema permite a las escuelas publicar ofertas,

editarlas, consultar sus postulados, activar o desactivar ofertas. Estas ofertas contienen toda la información sobre la materia, además se permite la carga de archivos (PDF, DOC, PPT, ETC) para indicar a los profesores sobre los requerimientos específicos para dictar la materia. Los profesores podrán consultar estas ofertas y postularse como candidatos a dictar la materia, y si por algún motivo ya no se encuentra disponibles para participar en esta oferta, pueden retirar su postulación. Las escuelas tomaran la decisión sobre quien será el más adecuado para la materia entre los distintos postulados. Este resultado se detalla en el capítulo IV en el desarrollo del Sprint 6.

CAPÍTULO VI - CONCLUSIONES Y RECOMENDACIONES

VI.1 Conclusiones

Luego del desarrollo de este Trabajo Instrumental de Grado, del conocimiento del proceso de gestión de inasistencias y de los requerimientos de la Universidad Católica Andrés Bello, se llegó a las siguientes conclusiones:

- El uso de la metodología SCRUM permitió los cambios durante el transcurso del desarrollo de forma que pudiera ser tomado en cuenta, ya que en cada Scrum diario se discuten mejoras que lograron de la aplicación un producto final de calidad. Tal es el caso del cambio en donde se decidió modificar las funcionalidades que tienen los usuarios en el módulo de "Perfil".

- La implementación de esta aplicación web mejora considerablemente el proceso de gestión de inasistencias, esto debido a que se reduce el tiempo que emplea el profesor para visualizar o justificar sus inasistencias, ofreciendo comodidad al realizar los procesos desde un navegador web.

- El encargado de las asistencias por parte de la Facultad o Escuela, reduce su carga de trabajo, ya que la aplicación facilita el proceso en general de carga de inasistencias. Por otra parte, se tendrá un control mayor de las recuperaciones de clase realizadas por el docente.

- La utilización del API de servicios Web simplifica el acceso a los datos, tomando en cuenta que el uso del API debe tener un nivel de seguridad con

los distintos servidores. Asimismo, si existen otros proyectos en desarrollo que requieran consumir los servidores utilizados se puede reutilizar el API, simplificando carga de trabajo.

- El uso de aplicaciones para generar diseños básicos de las interfaces da un buen punto de arranque para el desarrollo.

- La implementación de la arquitectura MVC facilitó la organización del código durante el desarrollo del proyecto, brindando facilidad en el mantenimiento y escalabilidad.

- Los profesores podrán consultar el resultado de las evaluaciones de sus alumnos en las materias que ha dictado en la UCAB, directamente desde la aplicación sin necesidad de recurrir a la escuela.

- Las ofertas facilitarán, a las escuelas, conseguir los profesores adecuados para las materias, y a los profesores, que tengan disponibilidad, conseguir una oferta laboral.

VI.2 Recomendaciones

Mediante la puesta en marcha de la web y las pruebas realizadas se determinaron las siguientes recomendaciones:

- Realizar un estudio de investigación para determinar el tiempo necesario que debe durar la cookie por cada sesión iniciada.

- Las ofertas de trabajo están limitadas únicamente a profesores dentro de la UCAB, se debería realizar a futuro una aplicación que permita acceder a estas ofertas a usuarios fuera de la universidad.

- Se debe considerar el desarrollo de una aplicación móvil, con éstas mismas funcionalidades, para aumentar el alcance del proyecto.

BIBLIOGRAFÍA

- Alexander, C. (1977). A Pattern Language. Oxford University Press. New York.

Referencias Electrónicas

- [1] Json (Noviembre de 2012). Json. [En línea]. Disponible en: http://www.json.org/json-es.html [2015, Septiembre].
- [2] Ken Schwaber y Jeff Sutherland (Julio de 2013). La Guía de Scrum. [En línea]. Disponible en: http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf [2015, Octubre].
- [3] Proyectos ágiles (2014). ¿Qué es SCRUM?. [En línea]. Disponible en: http://proyectosagiles.org/que-es-scrum/ [2015, Octubre].
- [4] Softeng (2015). Proceso y Roles de Scrum. [En línea]. Disponible en: https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html [2015, Octubre].
- [5] Vass Digital (Septiembre 2012). Manual de referencia [En línea]. Disponible en: http://vassdigital.com/blog/scrum-la-metodologia-de-desarrollo-agil-por-excelencia/ [2015, Septiembre].
- [6] w3c (Noviembre de 2013). Guía Breve de Servicios Web. [En línea]. Disponible en: http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb. [2015, Octubre].
- [7] Luján Mora , S. (2002). Programación de aplicaciones web: historia, principios básicos y clientes web. (1ª edición). Club Universitario. Disponible en: https://gplsi.dlsi.ua.es/almacenes/ver.php?pdf=42
- [8] Alicante, U. d. (2012, Agosto 20). http://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html

- [9] Creative Commons, A.-C. (2013). es.wikibooks.org. Recuperado el 05 de Noviembre de 2014, de es.wikibooks.org: http://es.wikibooks.org/wiki/Lenguaje_HTML/Introducci%C3%B3n
- [10] LibrosWeb. (2014). librosweb.es. Recuperado el 05 de Noviembre de 2014, de librosweb.es: http://librosweb.es/css_avanzado/capitulo_5.html
- [11] Atribución-CompartirIgual, C. C. (2011, Octubre 02). es.wikibooks.org. Recuperado el 05 de Noviembre de 2014, de es.wikibooks.org: http://es.wikibooks.org/wiki/Programaci%C3%B3n_en_JavaScript/Introducci%C3%B3n
- [12] Lessin, J. (2013, Noviembre 06). jorgelessin.com. Recuperado el 05 de Noviembre de 2014, de jorgelessin.com: http://jorgelessin.com/que-es-bootstrap-y-como-funciona-en-el-diseno-web/
- [13] Microsoft. (2014). www.asp.net. Recuperado el 05 de Noviembre de 2014, de www.asp.net: http://www.asp.net/mvc/mvc4
- [14] Jackson, R. (2013, Junio). rjprogramacion.blogspot.com. Recuperado el 05 de Noviembre de 2014, de rjprogramacion.blogspot.com: http://rjprogramacion.blogspot.com/2013/06/sql-server-conceptos-basicos.html
- [15] Hat, R. (2005). web.mit.edu. Recuperado el 12 de Noviembre de 2014, de web.mit.edu: http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ldap.html
- [16] Isla Visual (2014). Diferencias entre Scrum y XP. [En línea]. Disponible en: http://www.islavisual.com/articulos/desarrollo_web/diferencias-entre-scrum-y-xp.php
- [17] jQuery (2014). jQuery. [En línea]. Disponible en: http://jquery.com/

Apéndice

Apéndice A: Figuras

A1. Pantalla de resultados de búsqueda de profesor.

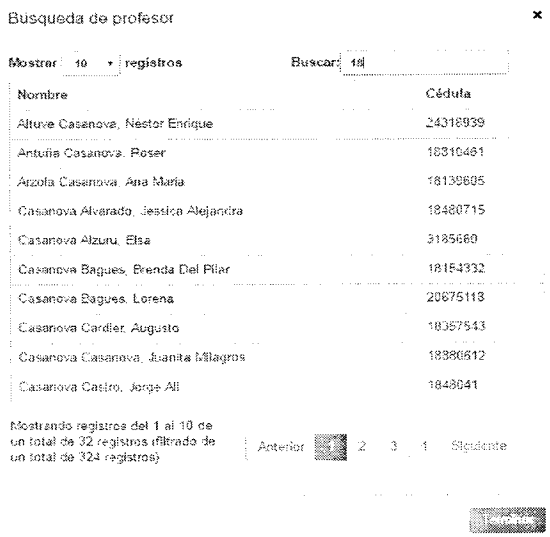


Figura 11-A. Resultados búsqueda de profesor.

Fuente: Elaboración propia.

A2. Pantalla de agregar datos curriculares.

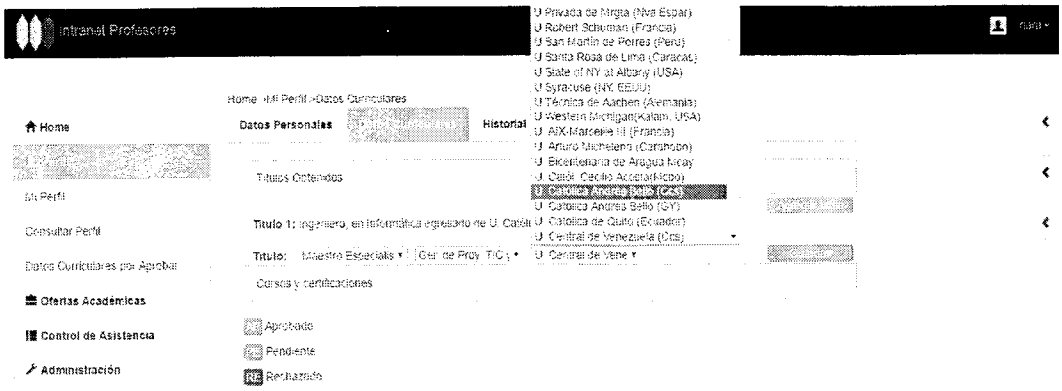


Figura 12-A. Lista de sedes, agregar datos curriculares.

Fuente: Elaboración propia.

A3. Pantalla módulo de inasistencias.

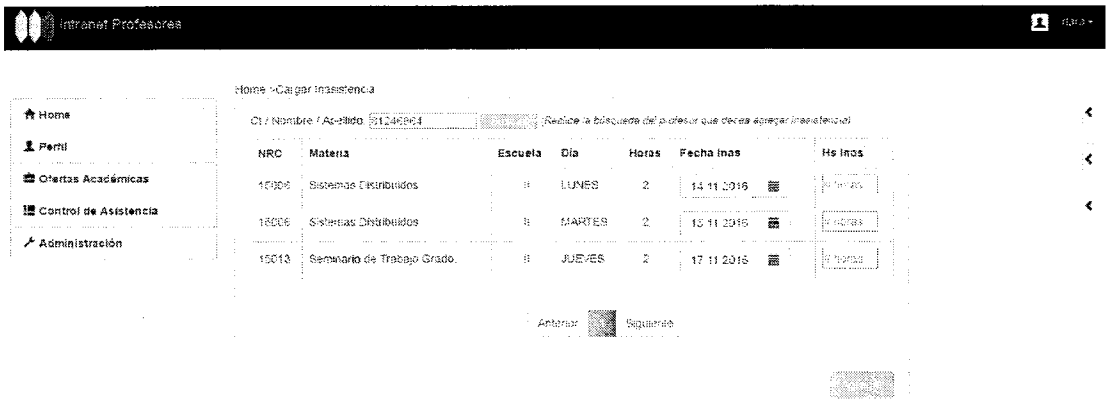


Figura 13-A. Módulo de inasistencias, "Cargar Inasistencia".

Fuente: Elaboración propia.

A4. Pantalla "Mis Inasistencias".

Figura 14-A. "Mis Inasistencias".

Fuente: Elaboración propia.

A5. Pantalla "Crear Oferta".

Oferta ×

Datos Generales

Escuela: Ingeniería Informática

Campus: UCAB Montalbán

Nombre de la materia: Algoritmos y Programación II

Semestre / Año: Cuarto Semestre

Curso: 00086

Horas a la semana: 6

Horario:

Tentativo | Disponibilidad de profesor

Día	Hora Inicio	Hora Fin	
Martes	2:00 PM	4:00 PM	Eliminar clase
Viernes	11:00 AM	1:00 PM	Eliminar clase

Contenido Programático

Requisitos

Figura 15-A. Crear Oferta.

Fuente: Elaboración propia.

A6. Pantalla "Mis Ofertas".

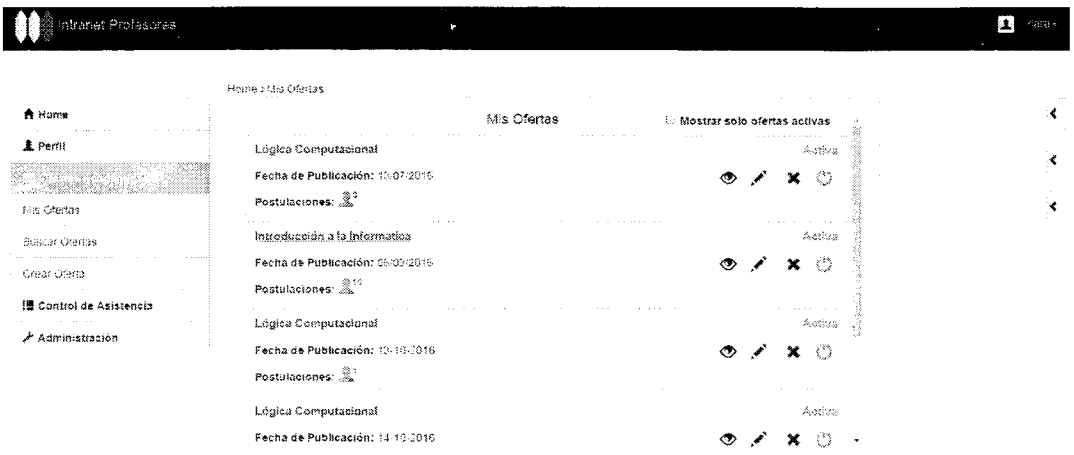


Figura 16-A. "Mis Ofertas".

Fuente: Elaboración propia.

A7. Pantalla "Administración".

Figura 17-A. "Administración".

Fuente: Elaboración propia.

A8. Bases de datos SQL Server

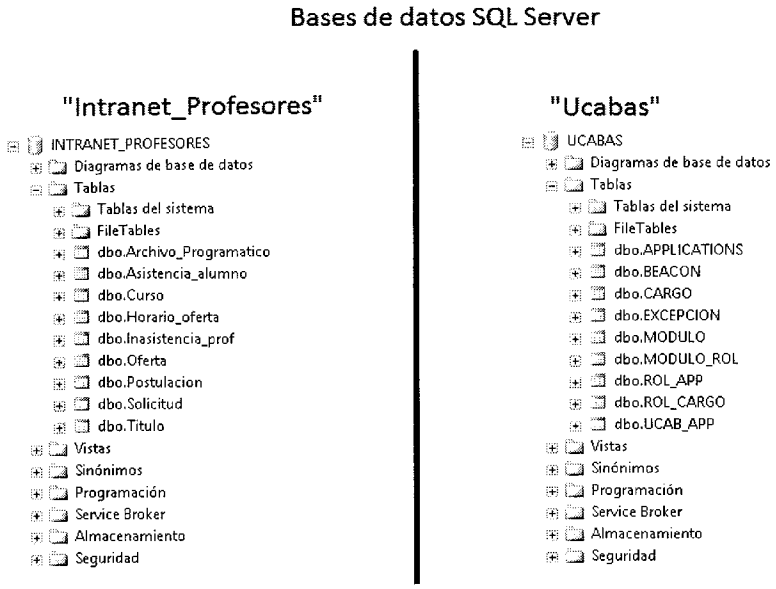


Figura 18-A. Bases de datos SQL Server.

Fuente: Elaboración propia.