

**Prototipo de Aplicación Web para la Gestión de la Carga Académica de los
Profesores de la Universidad del Valle**

Daniel Henao Arboleda
1123054-3743
daniel.henao@correounivalle.edu.co

Director
Doctor Carlos Mauricio Gaona Cuevas
mauricio.gaona@correounivalle.edu.co



Escuela de ingeniería de sistemas y computación
Facultad de ingeniería
Universidad del Valle
Santiago de Cali, Noviembre de 2016

Resumen

Este proyecto presenta el desarrollo de una aplicación web para gestionar la carga académica de los profesores de la Universidad del Valle, siguiendo las normas establecidas en la resolución #022 del Consejo superior. La aplicación fue desarrollada usando el framework Django de Python en conjunto con otras tecnologías web como HTML5, CSS3 y JavaScript. La aplicación permite realizar tareas como registro, edición y eliminación de actividades de profesores, calcular la cantidad de horas de dichas actividades y la carga total de cada profesor, además ofrece las funcionalidades necesarias para aprobar o rechazar dicha carga por parte de los directores de escuela o facultad; posee también funcionalidades para generación de reportes y muestra a los usuarios diferentes funcionalidades dependiendo del rol que ejercen a través del manejo de tipos de usuario.

Abstract

This project presents the development of a web application in charge of managing the academic load of the professors of Universidad del Valle, following the Standards set out in the resolution # 022 of the Superior Council. The application was developed using the Python Django framework in conjunction with other web technologies such as HTML5, CSS3 and JavaScript. The application allows to perform tasks such as registration, editing and elimination of professors' activities, calculating the number of hours of such activities and the total load of each professor, and provides the necessary functionalities to approve or reject such load by principals of school or faculty; also has functionalities for generating reports and shows users different functionalities depending on the role they exercise through the management of user types.

Tabla de Contenidos

Capítulo 1 Introducción	9
1.1 Planteamiento y formulación del problema	10
1.2 Justificación	11
1.2.1 Justificación social	11
1.2.2 Justificación académica	11
1.3 Objetivos.....	12
1.3.1 General	12
1.3.2 Específicos	12
Capítulo 2 Marco referencial.....	13
2.1 Estado del arte.....	13
2.2 Marco teórico	15
Capítulo 3 Estado de la gestión de la carga académica.....	20
3.1 Proceso de gestión	20
3.2 Tecnologías usadas en la gestión	21
3.3 Sistematización del proceso de gestión	21
Capítulo 4 Modelo computacional	22
4.1 Fórmulas	23
4.1.1 Actividades de docencia	23
4.1.2 Actividades de investigación, Actividades orgánicas y de extensión	27
4.1.3 Actividades académico administrativas	28
4.1.4 Número total de horas de la carga académica.....	28
4.2 Modelo del proceso de gestión de la carga académica	29
Capítulo 5 SICAP	30

	5
5.1 Iteración cero	30
5.1.1 Idea	31
5.1.2 Arquitectura del sistema	31
5.1.3 Módulos.....	32
5.1.4 Miembros del equipo	33
5.1.5 Prácticas ágiles	33
5.1.6 Release plan.....	34
5.1.7 Tecnologías.....	37
5.1.8 Actores.....	37
5.1.9 Historias de usuario	39
5.1.10 Requerimientos no funcionales.....	42
5.2 Detalles de implementación.....	43
5.2.1 Responsive design	43
5.2.2 Seguridad.....	43
5.2.3 Usabilidad	44
5.2.4 Gestión de usuarios	45
5.2.5 Gestión de dependencias	45
5.2.6 Gestión de asignaturas	46
5.2.7 Gestión de asignación de carga	46
5.2.8 Reportes	46
5.3 Pruebas.....	47
Capítulo 6 Resultados.....	50
6.1 Producto final	50
6.1.1 Usuario Profesor.....	52
6.1.2 Usuario Director	55
6.1.3 Usuarios Vicedecano y Decano	58
6.1.4 Usuario Operador	58
6.1.5 Usuario Administrador	60
Capítulo 7 Validaciones	61
7.1 Validación con periodo académico pasado.....	61

	6
7.2 Validación usando encuesta	63
Capítulo 8 Conclusiones y trabajo futuro	70
8.1 Conclusiones.....	70
8.2 Trabajo futuro.....	71
Lista de referencias.....	72

Lista de tablas

Tabla 1: Release Plan	34
Tabla 2: Actores de SICAP	37
Tabla 3: Historia de usuario	39
Tabla 4: Actividades de docencia	62
Tabla 5: Resultados de los cálculos de las actividades de docencia	63
Tabla 6: Preguntas usadas en la encuesta	64
Tabla 7: Escala de Likert	65

Lista de Ilustraciones

Ilustración 1: Diagrama Procesos de aprobación de carga	29
Ilustración 2: Iteración cero	30
Ilustración 3: Arquitectura de Django.....	31
Ilustración 4: Módulos de SICAP.....	33
Ilustración 5: Release plan.....	34
Ilustración 6: Ejemplo historia de usuario	42
Ilustración 7: Casos de prueba	48
Ilustración 8: Aceptación de casos de prueba	48
Ilustración 9: Issues Bitbucket	49
Ilustración 10: Características SICAP	51
Ilustración 11: Página principal SICAP.....	51
Ilustración 12: Dashboard profesor.....	52
Ilustración 13: Edición carga académica	53
Ilustración 14: Gestión actividades de docencia.....	54
Ilustración 15: Proponer carga	54
Ilustración 16: Detalles carga académica.....	55
Ilustración 17: Pagina de elección de rol de usuario.....	56
Ilustración 18: Aceptar/Rechazar carga	57
Ilustración 19: Modal de confirmación Aceptar/rechazar	57
Ilustración 20: Reporte Cargos profesores.....	59
Ilustración 21: Reporte General cargas académicas	59
Ilustración 22: Reporte carga académica profesor.....	60
Ilustración 23: Resultados pregunta 1	65
Ilustración 24: Resultados pregunta 2.....	66
Ilustración 25: Resultados pregunta 3.....	66
Ilustración 26: Resultados pregunta 4.....	67
Ilustración 27: Resultados pregunta 5.....	67
Ilustración 28: Resultados pregunta 6.....	68
Ilustración 29: Resultados pregunta 7.....	68
Ilustración 30: Resultados pregunta 8.....	69

Capítulo 1

Introducción

Las TIC (Tecnologías de la información y las comunicaciones) en las universidades han experimentado un gran desarrollo en todo el mundo. El cambio que éstas han sufrido se ha producido de forma gradual, pero constante. Por ello, hoy en día las TIC ya forman parte de la vida universitaria ordinaria y su integración, en todos los ámbitos y estamentos es, en general, una realidad.

En este sentido la gestión de los procesos en las universidades se constituye como una de las actividades más importantes para poder prestar un servicio eficiente a la comunidad. La carga académica de los profesores es uno de los procesos que requieren una gestión adecuada la cual constituye un esfuerzo que pretende ayudar al proceso de institucionalización de las prácticas universitarias. La implementación de la carga académica de una institución educativa consiste en definir las actividades que se realizarán en un periodo académico, lo que demanda una serie de procesos que permitan gestionar estas actividades de manera rápida y eficiente; así mismo los diversos actores deben tener claridad acerca de la estructura de la implementación y seguimiento de esta.

Para el caso de la Universidad del Valle el proceso de gestión de la carga académica de los profesores consiste en asignar una cierta cantidad de horas a las diferentes actividades que estos realizarán durante un periodo académico ya sea dando clases a los estudiantes, haciendo investigaciones o realizando procesos administrativos. Estas actividades deben

asignarse siguiendo una serie de normas establecidas por la universidad en la resolución 022 expedida por el consejo superior de la Universidad del Valle en mayo del 2001, en las cuales se especifica cómo asignar el número de horas a las actividades, el máximo o mínimo número de horas que se pueden asignar a una actividad dependiendo del tipo de actividad y otros factores.

En este proyecto se propone un proceso para gestionar la carga académica de los profesores de la Universidad del valle apoyado por una aplicación web para su gestión.

A Continuación se presenta el problema los objetivos y la metodología para la implementación de la aplicación descrita anteriormente.

1.1 Planteamiento y formulación del problema

La Universidad del Valle es una institución universitaria que posee una cantidad significativa de profesores a los cuales se les debe asignar por semestre un número de horas que estos dedicaran a ciertas actividades académicas y administrativas; a esto se le conoce como carga académica, asignar esta carga académica es una tarea que requiere una gran cantidad de trabajo, además del cumplimiento de ciertas normas, las cuales están especificadas en la Resolución 022 expedida por el consejo superior de la Universidad del Valle en mayo 8 de 2001 (Universidad del Valle, 2001).

Es de suma importancia que este proceso se realice de manera rápida y organizada para así evitar retrasos en otros procesos que puedan depender de este.

Actualmente la facultad de ingeniería cuenta con una aplicación para la gestión de la carga académica de los profesores de la facultad de ingeniería que fue desarrollada hace

aproximadamente doce años. Esta aplicación es considerada obsoleta ya que presenta una serie de dificultades para el registro y procesamiento de los datos de la carga académica lo cual causa demoras para carga de datos, dificultades para generar los reportes requeridos, adicionalmente el sistema no puede ser consultado en línea por los profesores. Por lo anterior en este proyecto se propone el desarrollo de una aplicación web para la gestión de la carga académica de los profesores de la escuela de ingeniería de sistemas y computación

1.2 Justificación

1.2.1 Justificación social

La aplicación a realizar es importante ya que, permitirá a la comunidad universitaria, en especial a los administradores y profesores, realizar los procesos de gestión de la carga académica de los profesores de forma organizada, de fácil utilización y persistentes. Lo que permitirá reducir tiempo; obtener información clara y concisa y evitar inconsistencias entre las actividades de los implicados.

1.2.2 Justificación académica

El desarrollo de la aplicación le permitirá al estudiante aplicar conocimientos adquiridos a lo largo de su carrera como son: el desarrollo de software, programación web, programación en bases de datos, aplicación de algoritmos estadísticos; también reforzar conocimientos y aprender nuevas técnicas de desarrollo y programación

1.3 Objetivos

1.3.1 General

Desarrollar una aplicación web para la gestión de la carga académica de la escuela de ingeniería de sistemas y computación (EISC) de la Universidad del Valle.

1.3.2 Específicos

- Elaborar un estudio del estado actual de la gestión de la carga académica de los profesores de la Universidad del Valle.
- Diseñar un modelo computacional basado en actores y responsabilidades para la gestión de la carga académica de los profesores de la EISC de la Universidad del Valle
- Diseño de un prototipo de la aplicación basado en el modelo diseñado
- Implementación del prototipo que contenga los módulos especificados en el diseño del modelo
- Validación del prototipo implementado en la EISC

Capítulo 2

Marco referencial

2.1 Estado del arte

A nivel global la gestión de la carga académica se ve en su mayoría enfocada a la gestión de actividades académicas estudiantiles , existen pocas aplicaciones en el mercado que se enfoquen a la carga académica de los profesores dado que cada institución define sus propios procesos de manera particular; sin embargo la mayoría de las universidades poseen sistemas de gestión de carga académica, a través de la implementación de programas locales o algunos ofertados por empresas especializadas que ofrecen las aplicaciones a clientes potenciales como en el caso de la empresa PwC (2010-2016) la cual desarrollo una aplicación web para la Universidad Curtin de Australia denominada AWMS (ACADEMIC WORLOAD MANAGEMENT SISTEM) (Curtin, 2016).

AWMS es un sistema de asignación que requiere que el Director de la Escuela ejerza un juicio sobre la asignación de la carga de trabajo académico dentro de un marco que permita alcanzar un volumen de total de trabajo razonable, existen procesos de gestión del rendimiento planificación de la escuela y de gestión de recursos en la Universidad.

Entre las aplicaciones de acceso general se tiene a Mimosa un programa de origen Finlandés.

“Mimosa es un software de programación versátil. A diferencia de la mayoría de otras aplicaciones de programación, Mimosa no fue desarrollado como una herramienta para

apoyar el proceso de planificación de una organización en particular. En su lugar, se centra en los retos fundamentales de la programación y ofrece una solución centrada en el usuario. Como resultado de ello puede ser utilizado en cualquier organización para cualquier tarea de programación. “Mimosa software Ltd. (2016)

Otra aplicación de software institucional es la AUS (Academic Unit System) de la universidad NTU (Nanyang Technological University 2012) que consta de varios componentes entre ellos se tienen:

- Requisitos curriculares que presenta la opción de visualizar: Estructura Curricular, Prerrequisitos, Estructura Académica, Período de Candidatura, Clasificación de los Estudiantes, Requisitos Residenciales y de Graduación, Transferencia de Unidades Académicas.
- Inscripción de Cursos que presenta la opción de visualizar Sistema de Registro Estudiantil Automatizado, cuotas pendientes, Añadir / Eliminar Cursos, Carga Académica, Repetición Restringida
- Exámenes y calificaciones que presenta la opción de visualizar: Exámenes, Sistemas de Clasificación, Opción Satisfactorio / Insatisfactorio, Freshman Year GPA Exemption, Clasificación de Grado, Permanente Académica, Lista del Decano.

La universidad de Exeter en el Reino Unido posee un software denominado “*SWARM*” University of Exeter.(2014-2015) “es el sistema de planificación de la carga de trabajo de la Universidad. Es una manera accesible de mostrar cómo se ha aplicado la

política de la carga de trabajo de la universidad. *SWARM* Está diseñado para proporcionar una asignación de carga de trabajo transparente, que debe ser utilizado por disciplinas para trabajar en pro de una distribución equitativa de trabajo para el personal académico en el próximo año académico. También puede ser utilizado para ayudar a los departamentos administrar cargas de trabajo sus académicos 'de manera eficiente.'”

Es denotar que muchas universidades en Colombia, usan hojas de Excel como herramienta para elaborar la carga académica de sus profesores.

2.2 Marco teórico

El presente proyecto se enmarca en el desarrollo de software web (aplicaciones web) para la gestión de procesos administrativos, en este caso la gestión de la carga académica de los profesores de la universidad del valle.

La carga académica de un profesor: se refiere a la cantidad de horas (semanales, mensuales o semestrales) dedicadas a una determinada actividad de docencia-investigativa o académico-administrativa.

Actividades de docencia: son las actividades en las que un profesor tiene contacto directo con los estudiantes tales como las clases magistrales, los seminarios, los talleres, laboratorios y las direcciones de tesis (trabajos de grado)

Actividad de investigación y desarrollo tecnológico, sistematización, creación artística y humanística: Es una actividad intelectual que se constituye en el eje de la

evolución del docente, de las transformaciones curriculares y de la modernización de la Universidad.

Actividad de extensión: es una actividad que cumple con función misional de la Universidad, a través de la cual integra y articula la docencia, la investigación y la creación artística con su entorno cultural, institucional, social y económico, validándose como institución responsable y comprometida con el destino y retos comunes

Actividades académico administrativas: son las actividades relacionadas con cargos administrativos en la universidad como directores de escuela, representantes académicos entre otros.

Actividades orgánicas complementarias: son actividades que se derivan de la misión de la universidad. Estas actividades son, entre otras, las siguientes: participación en actividades extracurriculares, participación en comités institucionales y atención a los estudiantes.

Carga programada: son las actividades que un profesor planea realizar en un periodo académico, esta planeación puede tener cambios por diferentes factores como son: cancelación o adición de cursos o proyectos entre otros.

Carga ejecutada: son las actividades que el profesor realizó en periodo académico.

Aplicaciones web: son herramientas de software que se ejecutan en un entorno web conectándose a un servidor a través de internet o intranet mediante un navegador, estas se codifican en lenguajes soportados por los navegadores web los cuales las ejecutan. Puesto que estas aplicaciones se ejecutan en un servidor, el usuario no requiere instalar una copia de esta en su computadora ya que accede a ella a través del navegador

Django es un framework de Python creado bajo las premisas de agilidad en el desarrollo, no repetición de código, simplicidad en el acceso a los datos y administración nativa. Estas características lo hacen ideal para recortar enormemente el esfuerzo requerido y los tiempos de desarrollo a la vez que se libera al proyecto de la creación de secciones de código que, aunque son críticas para la operación, no constituyen como tal el propósito del sistema.

Según la página oficial de Django Software Foundation (2005-2016). **Django** es un framework web de Python de alto nivel que fomenta el rápido desarrollo y diseño limpio y pragmático. Construido por los desarrolladores con experiencia, que se encarga de gran parte de la molestia de desarrollo Web, para que pueda centrarse en la escritura de su aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto.

Python es un lenguaje de programación de propósito general creado a finales de 1980. Es un lenguaje interpretado, de uso libre y gratuito disponible en las plataformas más utilizadas (Linux, BSD, UNIX, Windows y MAC). Es ampliamente utilizado en el desarrollo de aplicaciones web y se puede extender: se han creado más de 70 mil paquetes (bibliotecas o librerías) para ampliar su funcionalidad. Tiene una comunidad muy grande de soporte a nivel mundial y es uno de los lenguajes más usados por empresas e instituciones como Google, Intel, la Nasa, Instagram, Yahoo, ElasticHosts Cloud Servers, Blender, PyGame, University of California – Irvine, New Zealand Digital Library, entre miles más.

PostgreSQL es un sistema de base de datos, de código abierto objeto-relacional. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una

sólida reputación por su fiabilidad, integridad de datos y corrección. Se ejecuta en todos los sistemas operativos, incluyendo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. Es totalmente compatible con ACID, tiene soporte completo para claves foráneas, combinaciones, vistas, triggers y procedimientos almacenados (en varios idiomas). Incluye más de SQL: 2008 tipos de datos, incluyendo INTEGER, numéricos, booleanos, CHAR, VARCHAR, DATE, INTERVAL y TIMESTAMP. También es compatible con el almacenamiento de grandes objetos binarios, incluyendo imágenes, sonidos o vídeos. Cuenta con interfaces de programación nativo de C / C ++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, entre otros, y documentación excepcional.

Twitter Bootstrap es un framework o conjunto de herramientas de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales.

Es el proyecto más popular en GitHub y es usado por la NASA y la MSNBC junto a demás organizaciones.

Hoja de estilo en cascada o **CSS** (siglas en inglés de *cascading style sheets*) es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML. El World Wide Web Consortium (W3C) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

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

HTML5 (*HyperText Markup Language*, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: una «clásica», HTML (text/html), conocida como *HTML5*, y una variante XHTML conocida como sintaxis *XHTML5* que deberá servirse con sintaxis XML (application/xhtml+xml). Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo. La versión definitiva de la quinta revisión del estándar se publicó en octubre de 2014.

Al no ser reconocido en viejas versiones de navegadores por sus nuevas etiquetas, se recomienda al usuario actualizar su navegador a la versión más nueva, para poder disfrutar de todo el potencial que provee HTML5.

El desarrollo de este lenguaje de marcado es regulado por el Consorcio W3C.

Responsive design: Es una filosofía de diseño de interfaz para aplicaciones web que consiste en permitir a los componentes gráficos como formularios y botones adecuarse al tamaño y resolución del dispositivo donde se visualiza tales como: PCs, Smartphones, tablets, smart TVs entre otros. Lo que permite a la aplicación adaptarse a los diferentes dispositivos sin tener una versión para cada uno de ellos (Ethan Marcotte, 2010).

Capítulo 3

Estado de la gestión de la carga académica

La gestión de la carga académica se realiza siguiendo las pautas establecidas en la resolución 022 del consejo superior del 2001, en esta están especificadas como debe asignarse el número de horas a cada profesor dependiendo del tipo de actividad a realizar o el cargo que va a ejercer.

3.1 Proceso de gestión

Este proceso parte de una serie de actividades académicas, administrativas e investigativas que la Universidad planea realizar en un periodo académico. Posteriormente el profesor junto con un funcionario se encargan de asignar las diferentes actividades que el profesor realizará. De los cursos programados, el profesor acuerda con el director de la escuela cuales dictara. Las actividades que requieran de procesos extras como las de investigación y las académico-administrativas, se les realizan sus respectivos procesos previos antes de ser asignadas. Una vez se asignan las actividades, la carga académica se envía al director de escuela para que este la revise, si el director aprueba la carga, la firma y es enviada a la vicedecanatura para una nueva revisión de lo contrario se le regresa la profesor para que realice las correcciones necesarias; una vez la carga llega a la vicedecanatura, es revisada, si es aprobada es firmada por él vicedecano, y se envía al decano de la facultad para que confirme la aprobación.

3.2 Tecnologías usadas en la gestión

La carga académica de los profesores de la Universidad del Valle se gestionaba anteriormente a través de hojas de cálculo (Excel o similares). Las horas asignadas a los profesores para las distintas tareas (investigación, clases de pregrado, clases de postgrado, etc.) eran tabuladas en hojas de cálculo, una por cada profesor, y eran enviadas impresas o en dispositivos de almacenamiento a la decanatura de la facultad correspondiente. Finalmente, la recopilación de las hojas de cálculo era enviada a la vicerrectoría para ser revisada y aceptada.

3.3 Sistematización del proceso de gestión

Aproximadamente en el año 2004, se realizó una primera aplicación para sistematizar el proceso y alrededor del año 2011 se le hizo una actualización. Esta aplicación se denomina SSARA (sistema de soporte para la asignación y registro de las actividades académicas de los profesores de la facultad de ingeniería). Esta aplicación se utiliza en toda la facultad de ingeniería como herramienta de apoyo para la gestión de la carga académica, en ella se permite registrar las diferentes actividades que los profesores pueden realizar y asignarlas a sus respectivas cargas académicas. Si bien la aplicación cumple con las normas establecidas en la resolución 022 tiene problemas de usabilidad y no posee funcionalidades de aprobación/rechazo de las cargas académicas; Adicionalmente solo es usada por un solo usuario en la escuela y otro en la decanatura y los profesores solo pueden revisar la carga en papel o en un archivo pdf cuando esta lista; aspectos que busca mejorar la aplicación propuesta en este proyecto.

Capítulo 4

Modelo computacional

El modelo usado está basado en actores y responsabilidades lo que significa que las funciones que se pueden realizar en el sistema dependen del tipo de actor que esté interactuando en el momento con él. Los actores del sistema son:

Administrador

Quien se encarga de gestionar la parte de los usuarios asignaturas y dependencias del sistema.

Operador de oficina

Quien registra edita y elimina las actividades de las cargas académicas de los profesores de la escuela.

Profesor

Tiene las mismas funcionalidades del operador salvo que esté solo gestiona las cargas académicas de sí mismo.

Director

Quien revisa que las cargas académicas de los profesores de la escuela estén correctamente diligenciadas y quien las aprueba para que sean enviadas a la vicedecanatura.

Vicedecano

Quien revisa que las cargas académicas de la facultad estén correctamente diligenciadas y quien las aprueba o rechaza para que el decano de la aprobación final.

Decano

Quien aprueba o rechaza las cargas académicas dependiendo de la decisión del vicedecano y las políticas de la facultad.

La información detallada de las funcionalidades de los distintos actores puede verse en el Capítulo 5.

4.1 Fórmulas

La gestión de la carga académica requiere el uso de fórmulas para el cálculo de las horas de los diferentes tipos de actividades. Estas fórmulas están definidas en la resolución 022 del 2001 del consejo superior. A continuación, se presentan las fórmulas y las restricciones usadas en la aplicación por cada tipo de actividad de la carga académica.

4.1.1 Actividades de docencia

Para el cálculo del número de horas de las actividades de docencia se utiliza un multiplicador de docencia el cual varía dependiendo del tipo de curso y es aplicado al número de horas del curso o a los créditos dependiendo también del tipo de curso. A continuación se presentan los distintos tipos de actividades de docencia.

Cursos magistrales, seminarios y clases

Para este tipo de cursos el multiplicador es de 4.0 y se aplica al número de horas del curso y al porcentaje de dedicación del profesor para este curso. El multiplicador puede incrementar o disminuir dependiendo de los siguientes factores:

- Cuando un mismo curso se dicta a dos o más grupos distintos durante un mismo semestre, el multiplicador de docencia se reduce en un punto (1.0) para el segundo grupo y en dos (2.0) para el tercero.
- Cuando un curso es asistido por un monitor, el multiplicador se reduce en un punto (1.0).
- Teniendo en cuenta que para los cursos de pregrado se considera un grupo de 35 estudiantes y para los de posgrado se considera un grupo de 25 estudiantes. El multiplicador de docencia aumenta en 0.1 por cada grupo adicional de 3 estudiantes o fracción más y disminuye en 0.1 por cada grupo de 3 estudiantes o fracción menos.

Las horas de taller, horas de laboratorio y horas de trabajo de campo de un curso se les aplican su propio multiplicador por separado, 1.7 para las horas de talleres y 2.0 para las horas de laboratorio y horas de trabajo de campo.

Teniendo en cuenta las restricciones anteriores la fórmula para el cálculo de horas queda de la siguiente manera:

$$total = (factor \times (mag + sem + clase)) + (1.7 \times taller) + (2 \times (campo + lab)) \\ \times ded$$

Donde:

total: número de horas de la actividad de docencia.

factor: Multiplicador de docencia.

mag: horas magistrales

sem: horas seminario

clase: horas de clase

taller: horas de taller

campo: horas de trabajo de campo

lab: horas de laboratorio

ded: dedicación (toma valores entre 0 y 1)

Curso dirigido

Los cursos donde se trata de tener una equivalencia con los cursos regulares pueden tener un máximo de 9 estudiantes. Para este tipo de cursos solo se toma el número de estudiantes para calcular el número de horas.

Se asigna 22 horas para 1 estudiante y 121 horas para 9 estudiantes. La fórmula que cumple con esta relación lineal es la siguiente.

Se tienen los dos puntos (x, y) : $(1, 22)$ y $(9, 121)$ donde “y” es el número de horas y “x” es el número de estudiantes

Teniendo en cuenta la ecuación de la recta $y = ax + b$ se halla primero “a” que es la pendiente de la recta de la siguiente manera:

$$a = \frac{121 - 22}{9 - 1} = 99/8$$

Posteriormente se tiene el sistema de ecuaciones

$$22 = 99/8 \times 1 + b$$

$$121 = 99/8 \times 9 + b$$

Se suman las dos ecuaciones y queda

$$143 = 990/8 + 2b$$

Se despeja b

$$b = 77/8$$

Finalmente la ecuación para hallar el número de horas de los curso dirigidos queda de la siguiente manera

$$y = \frac{99}{8}x + \frac{77}{8}$$
 Donde “y” es el número de horas resultante y “x” es el número de

estudiantes.

Direcciones de trabajos de grado

Las direcciones de trabajo de grado son un tipo de actividades de docencia que no están relacionadas a un curso por los que para el cálculo de horas se utiliza un valor fijo por estudiante de la siguiente manera:

22 horas para trabajos de pregrado, 33 horas para trabajos de maestría y 44 horas para trabajos de doctorado.

El tope para el número de horas que se pueden asignar por el total de trabajos de grado en un semestre es de 216 horas.

Cursos regulares individualizados y prácticas supervisadas

Para este tipo de cursos se calcula el número de horas multiplicando el número de semanas del periodo académico por el número de créditos del curso por el factor de docencia. Para la práctica supervisada el factor toma valores entre 1.2 y 1.5; para el curso individualizado el factor es de 1.2.

4.1.2 Actividades de investigación, Actividades orgánicas y de extensión

Puesto que el número de horas depende de la investigación como tal, no se realiza ningún cálculo es decir las horas se definen en el proyecto. Lo mismo ocurre con las actividades orgánicas y las de extensión. El número de horas se registrara explícitamente y se definen en la facultad o escuela. El número máximo de horas que se pueden asignar por cada uno de estos tipos de actividades es de 440 horas.

4.1.3 Actividades académico administrativas

El número de horas depende del cargo que realice el profesor, el número de profesores en la escuela, el número de estudiantes matriculados y el tipo de dedicación del profesor. La asignación tiene un rango de entre 110 horas y 550 horas dependiendo de los criterios anteriormente mencionados, y como no hay una fórmula para este cálculo de horas, se deben registrar explícitamente en la aplicación.

4.1.4 Número total de horas de la carga académica

El total de horas de una carga académica es la sumatoria del número de horas de las actividades asignadas a esta.

$$total = \sum \text{ docencia} + \sum \text{ tesis} + \sum \text{ orgánicas} + \sum \text{ extensión} + \sum \text{ investigación} + \sum \text{ administrativas}$$

Donde:

total: Número de horas de la carga académica que realiza el profesor en el periodo.

docencia: Número de horas de las actividades de docencia.

tesis: Número de horas de los trabajos de grado.

orgánicas: Número de horas de las actividades orgánicas.

extensión: Número de horas de las actividades de extensión.

investigación: Número de horas de las actividades de investigación.

administrativas: Número de horas de las actividades académico administrativas.

4.2 Modelo del proceso de gestión de la carga académica

La Ilustración 1 muestra el diagrama de procesos realizados para el registro y aprobación de la carga académica de los profesores de la universidad del valle.

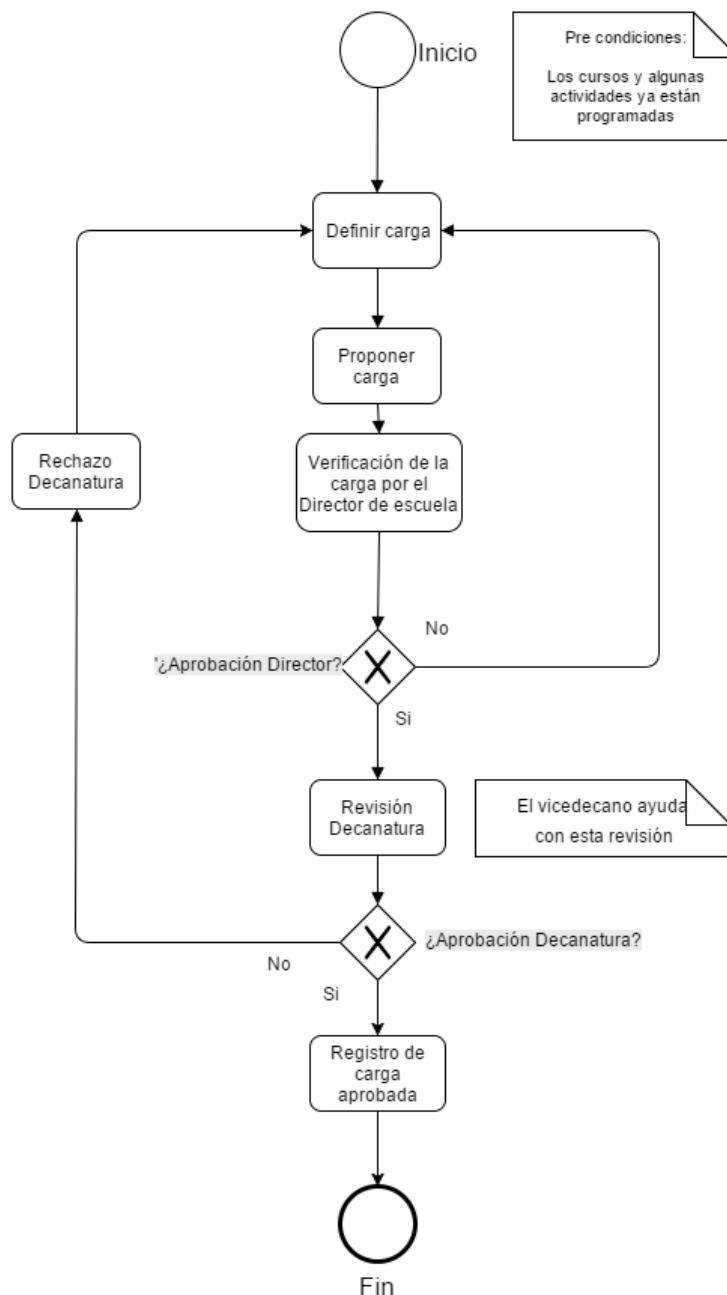


Ilustración 1: Diagrama Procesos de aprobación de carga

Capítulo 5

SICAP

La aplicación web desarrollada en este trabajo de grado se denomina SICAP (Sistema de gestión de la carga académica de los profesores) y se desarrolló siguiendo practicas agiles basadas en la metodología XP. A continuación se detalla el proceso de desarrollo el cual sigue estas prácticas.

5.1 Iteración cero

La iteración cero es donde se identifican las necesidades y donde se da una visión de cómo será la aplicación web.



Ilustración 2: Iteración cero

5.1.1 Idea

Realizar una aplicación web para la gestión de la carga académica de los profesores de la Universidad del Valle que permita lo siguiente:

- El registro de actividades.
- La edición de actividades
- La eliminación de actividades.
- La aprobación y rechazo de cargas académicas.
- Reportes.

5.1.2 Arquitectura del sistema

Para el desarrollo de la aplicación web SICAP se utilizó la arquitectura multicapas propuesta por el framework Django la cual está compuesta por el patrón MTV (Model Template View) que está basado en el patrón MVC (Modelo Vista Controlador)

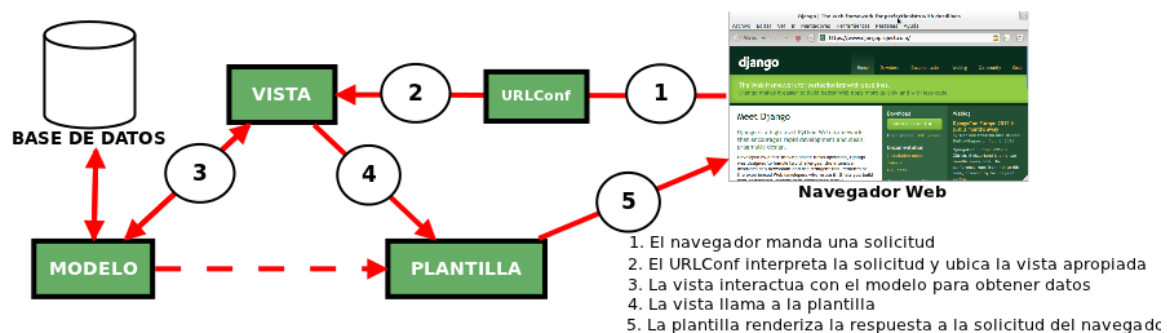


Ilustración 3: Arquitectura de Django

En el patrón MVC, el modelo se refiere a la capa de almacenamiento de datos, la vista es la capa que decide qué mostrar y cómo mostrarlo y el controlador es el encargado de decidir que vista usar dependiendo de la interacción del usuario con el sistema. Django sigue el patrón MVC solo que la parte del controlador es manejada por el mismo framework y lo más importante se produce en los modelos, las plantillas y las vistas hace que se le conozca como un framework MTV.

M significa "Model" (Modelo), la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.

T significa "Template" (Plantilla), la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web u otro tipo de documento.

V significa "View" (Vista), la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada (L. Web, 2016).

5.1.3 Módulos

La Ilustración 4 muestra los módulos en las que se divide SICAP, por cada módulo se creó un grupo de historias de usuario:

- Gestión de usuarios.
- Gestión de dependencias.
- Gestión de asignaturas.
- Gestión de asignación de carga
- Reportes

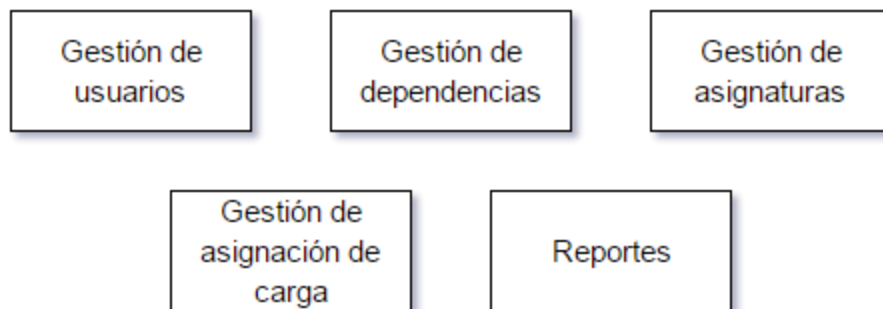


Ilustración 4: Módulos de SICAP

5.1.4 Miembros del equipo

Los miembros del equipo que participaron en el desarrollo de la aplicación fueron Daniel Henao Arboleda (Desarrollador) y Carlos Mauricio Gaona (Director del proyecto)

5.1.5 Prácticas ágiles

Las prácticas ágiles que se aplicaron para el desarrollo de la aplicación fueron:

- Historias de usuario
- Release plan
- Iteraciones de pocas semanas
- Pequeñas entregas
- Reuniones constantes
- desarrollo orientado a pruebas
- Estándar de codificación

5.1.6 Release plan

El release plan para este proyecto se realizó en la en la herramienta eXPert (Lenis, 2015) y la tabla muestra en detalle el release plan construido

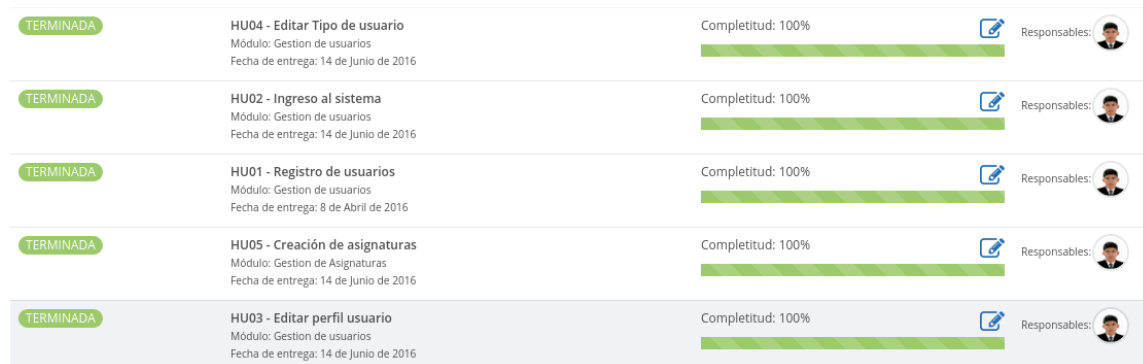


Ilustración 5: Release plan

Tabla 1: Release Plan

Módulo	Historia de usuario	% Avance	Fecha de entrega
Iteración 0			
-	Plan inicial del proyecto	100%	06/04/2016
-	Definición de arquitectura del sistema	100%	06/04/2016
-	Definición de los módulos de la aplicación	100%	06/04/2016
-	Definición de historia de usuario	100%	
-	Configuración ambiente de trabajo	100%	06/04/2016
-	Selección de plantilla	100%	06/04/2016

 Iteración 1

Gestión de usuarios	HU01-Registro de usuarios	100%	08/04/2016
Gestión de usuarios	HU02-Ingreso al sistema	100%	14/06/2016
Gestión de usuarios	HU03-Editar perfil usuario	100%	14/06/2016
Gestión de usuarios	HU04-Editar tipo de usuario	100%	14/06/2016
Gestión de asignaturas	HU05-Creación de asignaturas	100%	14/06/2016
Gestión de asignaturas	HU06-Editar asignaturas	100%	14/06/2016

 Iteración 2

Gestión de dependencias	HU26 - Edición de escuelas	100%	14/06/2016
Gestión de asignación de Carga	HU17 - Crear periodos académicos	100%	14/06/2016
Gestión de asignación de Carga	HU19-Crear actividades docencia magistrales	100%	14/06/2016
Gestión de asignación de Carga	HU25-Crear actividades académico administrativas	100%	15/06/2016
Gestión de asignación de Carga	HU29-Edición y borrado de actividades orgánicas	100%	15/06/2016
Gestión de dependencias	HU16-Edición de programas académicos	100%	14/06/2016
Gestión de asignación de Carga	HU14-Modificación de carga académica director	100%	17/06/2016
Gestión de dependencias	HU12-vinculación a dependencias	100%	14/06/2016
Gestión de asignación de Carga	HU07-Consultar carga académica	100%	22/06/2016

Gestión de asignación de Carga	HU09-Proponer carga académica	100%	21/06/2016
Gestión de asignación de Carga	HU22-Crear actividades de investigación	100%	15/06/2016
Gestión de dependencias	HU18-Crear escuelas	100%	14/06/2016
Gestión de dependencias	HU11-creación de programas académicos	100%	14/06/2016
Gestión de asignación de Carga	HU08-Planeación carga académica	100%	17/06/2016
Gestión de asignación de Carga	HU24-Crear actividades de extensión	100%	14/06/2016
Gestión de asignación de Carga	HU20-Editar y borrar actividades de docencia magistrales	100%	15/06/2016
Gestión de asignación de Carga	HU23-crear actividades orgánicas	100%	15/06/2016
Gestión de asignación de Carga	HU28 - Edición y borrado de actividades de investigación	100%	15/06/2016
Gestión de asignación de Carga	HU30 - Edición y borrado de actividades de extensión	100%	15/06/2016
Gestión de asignación de Carga	HU13 - Aprobación/Rechazo de carga académica	100%	22/06/2016
Gestión de asignación de Carga	HU31 - Edición y borrado de actividades académico administrativas	100%	15/06/2016

5.1.7 Tecnologías

Las tecnologías utilizadas para el desarrollo de la aplicación son las siguientes:

- Front-end (lógica de presentación): HTML5, CSS(Bootstrap3), JavaScript
- Back-end (lógica del negocio): Python 3.4, Django 1.9
- Gestor de base de datos: PostgreSQL
- Control de versiones: Git/Bitbucket

Adicionalmente para los componentes de interfaz gráfica se utilizó la plantilla INSPINIA de WrapBootstrap (Wrapbootstrap, 2016)

5.1.8 Actores

La Tabla 3 muestra a los actores del sistema con sus correspondientes funcionalidades que pueden realizar dentro de este.

Tabla 2: Actores de SICAP

Actor	Funcionalidades
Administrador	Registrar usuarios
	Editar rol de usuarios
	Habilitar/inhabilitar usuario
	Registrar y editar asignaturas
	Registrar y editar escuelas

Registrar y editar programas académicos

Registrar y editar periodos académicos

Profesor	Crear y editar sus Cargas académicas
	Crear, editar y borrar Actividades de docencia
	Crear, editar y borrar Actividades investigación
	Crear, editar y borrar Direcciones de tesis
	Crear, editar y borrar Actividades de extensión
	Crear, editar y borrar Actividades orgánicas
	Crear, editar y borrar Actividades académico administrativas
	Proponer sus cargas o cancelar la propuesta
	Ver comparación carga programada/ejecutada
Ver reporte personal de carga programada/ejecutada	

Operador de oficina	Crear y editar Cargas académicas de los profesores de la escuela
	Crear, editar y borrar Actividades de docencia
	Crear, editar y borrar Actividades investigación
	Crear, editar y borrar Direcciones de tesis
	Crear, editar y borrar Actividades de extensión
	Crear, editar y borrar Actividades orgánicas

Crear, editar y borrar Actividades académico administrativas

Proponer cargas o cancelar la propuesta

Ver resumen de las cargas académicas programadas/ejecutadas de la escuela

Ver reporte personal de carga programada/ejecutada

Director	Aprobar/rechazar cargas académicas
	Editar cargas académicas de la escuela a la que pertenece
Vicedecano	Aprobar/rechazar cargas académicas
Decano	Aprobar/rechazar cargas académicas

5.1.9 Historias de usuario

Las funcionalidades del sistema se especificaron como historias de usuarios y se listan en la tabla por cada módulo de la aplicación.

Tabla 3: Historia de usuario

Módulo	ID	Título historia de usuario
Gestión de usuarios	HU01	Registro de usuarios
	HU02	Ingreso al sistema
	HU03	Editar perfil usuario
	HU04	Editar Tipo de usuario
	HU33	Cambiar tipo de usuario

	HU37	Habilitar/inhabilitar usuario
Gestión de dependencias	HU11	Creación de programas académicos
	HU12	vinculación a dependencias
	HU16	Edición de programas académicos
	HU18	Crear escuelas
	HU26	Edición de Escuelas
Gestión de asignaturas	HU05	Creación de asignaturas
	HU06	Editar asignaturas
Gestión de asignación de carga	HU07	Consultar carga académica
	HU08	Planeación carga académica
	HU09	Proponer carga
	HU13	Aprobación/Rechazo de carga académica
	HU14	Modificación de carga académica Director
	HU17	Crear periodos académicos
	HU19	Crear actividades de docencia magistrales
	HU20	Editar y borrar actividades de docencia magistrales
	HU21	Crear direcciones de tesis
	HU22	Crear actividades de investigación

	HU23	Crear actividades orgánicas
	HU24	Crear actividades de extensión
	HU25	Crear actividades académico administrativas
	HU27	Edición y borrado de tesis
	HU28	Edición y borrado de actividades de investigación
	HU29	Edición y borrado de actividades orgánicas
	HU30	Edición y borrado de actividades de extensión
	HU31	Edición y borrado de actividades académico administrativas
	<hr/>	
	HU32	Reporte cargas académicas ejecutadas por escuela
	HU34	Reporte cargas académicas programadas por escuela
Reportes	HU35	Reporte carga académica programada ejecutada de un profesor
	HU36	Reporte cargos por periodo profesores de escuela
	HU38	Dashboard profesor
	HU39	Dashboard Director Vicedecano y Decano
	<hr/>	

Se usó la herramienta eXPert (Lenis, 2015) para el registro y control de las Historias de Usuario definidas anteriormente. En total se trabajaron 37 Historias de Usuario. En la Ilustración 6 se muestra un ejemplo de una historia de usuarios en eXPert.

Historia de usuario
✕

Identificador: HU02	Nombre: Ingreso al sistema	
Fecha de creación: 25 de Octubre de 2016 a las 14:30	Autor: Daniel Henao Arboleda	Puntos estimados: 2
Actor(es): Operador de oficina, Administrador, Profesor, Decano, Director	Modulo: Gestion de usuarios	

Descripción:

como usuario deseo hacer uso de un login para ingresar al sistema

Observaciones:

Para el ingreso se usa el nombre de usuario y la contraseña

Archivos Adjuntos:

No hay ningún archivo adjunto asociado

Pruebas de aceptacion:

👁 Ver pruebas

✍ Editar prueba

Ilustración 6: Ejemplo historia de usuario

5.1.10 Requerimientos no funcionales

La siguiente es la lista de requerimientos no funcionales.

- La interfaz gráfica debe ser usable, clara y consistente.
- La interfaz gráfica debe tener design responsive

- Los permisos de acceso al sistema podrán ser cambiado únicamente por el administrador del sistema.
- Los mensajes de error deben ser claros y concisos.
- Los sistemas operativos que soportan la herramienta deben ser Windows, GNU/Linux y Macintosh.

5.2 Detalles de implementación

5.2.1 Responsive design

Para la implementación del responsive design en la interfaz gráfica de SICAP se utilizó la plantilla INSPINIA (Wrapbootstrap, 2016) la cual tiene componentes con la propiedad “responsive”. Esta propiedad permite a los formularios y otros componentes de la interfaz adecuarse al tamaño de la pantalla del dispositivo en el que se visualiza. Además, se modificaron los CSS sin interferir con el responsive design de la plantilla, y se mejoró en partes donde la plantilla no incluye esta propiedad.

5.2.2 Seguridad

La seguridad de SICAP es proporcionada por el framework de Django, ya que posee un sistema de permisos, encriptación de contraseñas y control para ataques CSRF (un problema de seguridad popular, que fuerza al navegador web a enviar una petición a una aplicación web vulnerable). También, se realizaron validaciones del lado del cliente con Javascript y del lado del servidor con Python. Estas validaciones corresponden a los tipos de datos, los permisos y lo que se muestra al cliente evitando mostrar información sensible.

5.2.3 Usabilidad

En SICAP se tuvieron en cuenta diferentes aspectos de usabilidad (Hassan Montero & Martín Fernández, 2002):

- **Manejo de colores:** Se manejaron los colores propuestos por la plantilla teniendo lo que se mostrará en formularios la barra de navegación y los botones.
- **Lenguaje amigable con el usuario:** El texto usado en los formularios, mensajes, títulos entre otros es lo más familiar posible a un usuario de tipo docente.
- **Jerarquía visual:** La posición del menú principal, las opciones del perfil de usuario, el logo de la aplicación, los títulos, los subtítulos y los formularios están ubicados de tal manera que el usuario entienda en donde estarán estos componentes de acuerdo a su jerarquía y evitar que este se pierda (no saber en qué parte del sitio se encuentra).
- **Cancelar e inicio:** los botones cancelar, inicio y atrás están siempre en la misma posición lo que facilita al usuario encontrarlos siempre.
- **Estandarización:** Todos los componentes de la interfaz gráfica que se repiten como: botones, paneles, títulos, textos, enlaces, tablas y la barra de navegación. conservan siempre el mismo estilo y color.
- **Dashboard o página de inicio:** El dashboard de SICAP además de mostrar información necesaria y resumida, también permite dirigirse directamente a las páginas que permiten gestionar la información mostrada en él.

5.2.4 Gestión de usuarios

Se usó el módulo por defecto que trae Django para la gestión de usuarios como los son creación, edición, inhabilitación, permisos e ingreso, con algunas modificaciones de acuerdo a los requerimientos de la aplicación.

Puesto que la eliminación de usuarios puede generar problemas en las dependencias de los datos así como en los reportes, los usuarios de Django tienen un atributo por defecto que determina si el usuario está activo o no (es una especie de borrado lógico y este permite que el usuario no pueda interactuar con la aplicación si está en estado inactivo).

Se crearon seis grupos de usuarios que son: Administrador, Profesor, Director, Vicedecano, Decano y Operador con sus respectivas restricciones y permisos lo que especifica que puede hacer un usuario. Y se implementó una página de permiso denegado que se muestra cuando el usuario ingresa a una URL sin estar registrado o autorizado en el sistema.

5.2.5 Gestión de dependencias

En este módulo se registran y editan las dependencias que son los programas académicos y las escuelas. Se crearon los modelos, vistas y plantillas necesarias para el CRUD (Crear, Leer, Actualizar y Borrar). Puesto que el borrado de las dependencias afecta la integridad de la base de datos, la funcionalidad borrar no se implementó en la aplicación, esta acción solo puede realizarse desde el sitio administrativo de Django con el Súper Usuario y desde la base de datos directamente.

5.2.6 Gestión de asignaturas

En este módulo se registran y editan las asignaturas que son usadas para el cálculo del número de horas de las actividades de docencia. Se crearon los modelos vistas y templates requeridos para el CRUD además de una función de inhabilitación de asignaturas puesto que eliminarlas genera conflictos en la base de datos.

Se implementó un selector de escuela para que el administrador gestione las asignaturas de una manera más organizada.

5.2.7 Gestión de asignación de carga

En este módulo se registran se editan y borran las cargas académicas además de todas las actividades asociadas a estas.

Las actividades están divididas en seis tipos: Actividades de docencia, direcciones de tesis, actividades de investigación, actividades orgánicas, actividades de extensión y actividades académico administrativas. Para cada uno de los tipos de actividad se implementaron funciones que calculan el número de horas de cada actividad dependiendo de los datos que esta contiene, las fórmulas utilizadas están especificadas en la resolución 022 y se ejecutan cuando una actividad es creada o editada, en el caso de la eliminación se resta el número de horas de la actividad del total de horas de la carga académica.

Se implementó también las funciones de proponer carga y la aprobación/rechazo de cargas académicas por parte de los usuarios Director, Vicedecano y Decano.

5.2.8 Reportes

Para el módulo de reportes se utilizó la librería MorrisJS charts para los gráficos de barras y datatables para los reportes en tablas. Los reportes especificados son:

- **Reporte carga académica de un profesor:** Muestra la información de la carga académica así como de las actividades asociadas a esta de un profesor en un periodo seleccionado. Se puede elegir entre carga programada y carga ejecutada. Este reporte se muestra en una tabla la cual puede imprimirse.
- **Reporte de carga académica de una escuela:** Muestra la carga académica de los profesores de una escuela con sus respectivas actividades y total de horas. Este reporte facilita la revisión de todas las cargas académicas sin necesidad de generar un reporte por cada una. Se espera que se use para procesos administrativos por lo que tiene la opción de imprimirse.
- **Reporte de cargos de un profesor:** Muestra un historial de los cargos que ha ejercido un profesor a lo largo de los periodos.
- **Reporte de cargos de los profesores de una escuela:** Muestra los cargos académicos que ejercieron los profesores de una escuela en un periodo especificado.
- **Número de horas de las cargas académicas últimos cinco periodos:** Muestra el número de horas de las cargas académicas programadas y ejecutadas en los últimos cinco períodos académicos. Este reporte se muestra en el Dashboard de cada profesor y se representa en un gráfico de barras donde se muestran el número de horas de una carga académica en función del periodo.

5.3 Pruebas

En esta sección se muestran las pruebas realizada a la aplicación, teniendo en cuenta el concepto de pruebas ágiles siendo coherente con la metodología implementada.

Pruebas de aceptación: En el diseño de estas pruebas se crearon los criterios de aceptación para cada historia de usuario usando la herramienta eXPert. Al ejecutarse la prueba se le daba un estado de aceptado o rechazado. En las Ilustraciones 7 y 8 se muestra como se utilizó la herramienta eXPert para las pruebas de aceptación

Tabla de Casos:

Si desea ingresar un nuevo caso de prueba de click en añadir caso

+ Nuevo Caso Buscar:

Resultado Esperado	Observaciones	DEDICACION	GRUPO	MONITOR	NUMERO DE ESTUDIANTES	TIPO	CURSO	Borrar Caso
2da actividad de tipo magistral con mismo curso que pertenece a una misma carga debe tener menos horas	Vacio	100	01	no	35	magistral	Fundamentos de programacion	
Actividad creada correctamente	Vacio	100	01	no	35	magistral	Fundamentos de programacion	
ningun campo debe estar vacio	Vacio	100	01	no	Vacio	magistral	Fundamentos de programacion	

Mostrando 1 a 3 de 3 registros < 1 >

Ilustración 7: Casos de prueba

Casos de Prueba de HU19											
Código	DEDICACION	GRUPO	MONITOR	NUMERO DE ESTUDIANTES	TIPO	CURSO	Resultado Esperado	Resultado Obtenido	Observaciones	Estado	Opción
P-HU19-3	100	01	no	35	magistral	Fundamentos de programacion	2da actividad de tipo magistral con mismo curso que pertenece a una misma carga debe tener menos horas	Vacio		Pendiente	
P-HU19-2	100	01	no	Vacio	magistral	Fundamentos de programacion	ningun campo debe estar vacio	Vacio		Pendiente	
P-HU19-1	100	01	no	35	magistral	Fundamentos de programacion	Actividad creada correctamente	Vacio		Pendiente	

Ilustración 8: Aceptación de casos de prueba

Las anteriores Ilustraciones muestran el control de prueba de aceptación realizadas en la herramienta eXPert

Los errores y las mejoras fueron registradas en la sección issues (incidencias) de Bitbucket. Aquí se hace un control de los errores presentados en la funcionalidad o interfaz gráfica de la aplicación. También se registraron incidencias sobre la usabilidad el diseño y la ortografía. Puesto que las funcionalidades se probaron antes de subir al repositorio se reportaron pocos errores en la lógica de la aplicación en cambio la mayoría de las incidencias fueron sobre la posición o los colores de los elementos gráficos y la ortografía. La Ilustración 9 muestra el registro y control de las incidencias en Bitbucket.

Issues (1–24 of 24)

Title	T	P	Status	Votes	Assignee	Created	Updated
#24: Botones - limpiar ir a edición carga - Sugerencias	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#23: Codigos - Estudiante, Programa y curso - Sugerencia	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#21: Reporte carga de un profesor - Mostrar nombre del profesor completo - Sugerencia	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#22: Mayúsculas, minúsculas, tildes - Sugerencias	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#20: Registrar trabajo de grado - Nivel - Sugerencia	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#19: Menú principal - Negrilla en seccion actual	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#18: Registro de usuarios- Colores bootnes wizard - Sugerencias	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#17: Registrar Carga Académica - Agregar "*" de campos obligatorios - Sugerencia	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#16: Duda reporte carga de profesores	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#15: Profesor - Mensaje de notificación al eliminar	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#14: Administrador - Editar una asignatura	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#13: Usuarios - Imagen por defecto	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#12: Formulario con palabra repetida	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#11: Etiquetas de campos en inglés	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#10: Administrador - Editar nombres de usuario	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#9: Nombrar Buscar... como Listar..	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#8: Cambiar imagen de perfil - Sugerencias	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#7: Centrar contenido de dashboard profesor	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#6: Botón Volver (Regresar) - Sugerencias	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#5: Panel de explicación - Profesor/Mis asignaciones/Listar	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#4: Revisar tildes	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#3: Registrar usuarios - Redirección al registrar usuario - Sugerencia	+	+	NEW		Daniel Henao ...	22 hours ago	22 hours ago
#2: Profesor - Ver más modales - Sugerencia	+	+	NEW		Daniel Henao ...	23 hours ago	23 hours ago
#1: Formularios - Opción limpiar - Sugerencia	+	+	NEW		Daniel Henao ...	23 hours ago	23 hours ago

Ilustración 9: Issues Bitbucket

Capítulo 6

Resultados

En este capítulo se muestra el producto final del trabajo de grado con cada uno de sus componentes.

6.1 Producto final

La aplicación web para efecto de pruebas está alojada en un servidor en la nube en <http://sicap.ruatsonline.com/>. Se puede acceder a ella a través de un usuario y una contraseña según el tipo de usuario, donde cada uno puede realizar las actividades definidas anteriormente.

Puesto que la aplicación es para uso de la Universidad del Valle, los usuarios son funcionarios de la universidad por los que son registrados por el administrador del sistema.

Al ingresar al sitio se puede observar una página de presentación de la aplicación donde se muestran las características, el equipo de desarrollo e información de contacto como se muestra en la Ilustración 10.

CARACTERISTICAS

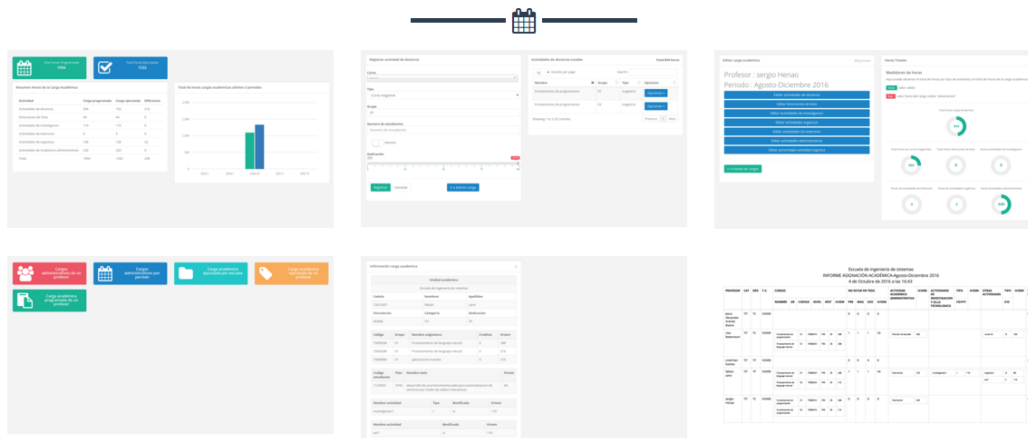


Ilustración 10: Características SICAP

Para ingresar al sistema se debe dar clic en el botón “iniciar sesión” (Ver Ilustración 11).



Ilustración 11: Página principal SICAP

Después de ingresar se despliega la página de inicio de acuerdo al tipo de usuario. La

Ilustración 12 muestra el inicio de un profesor.

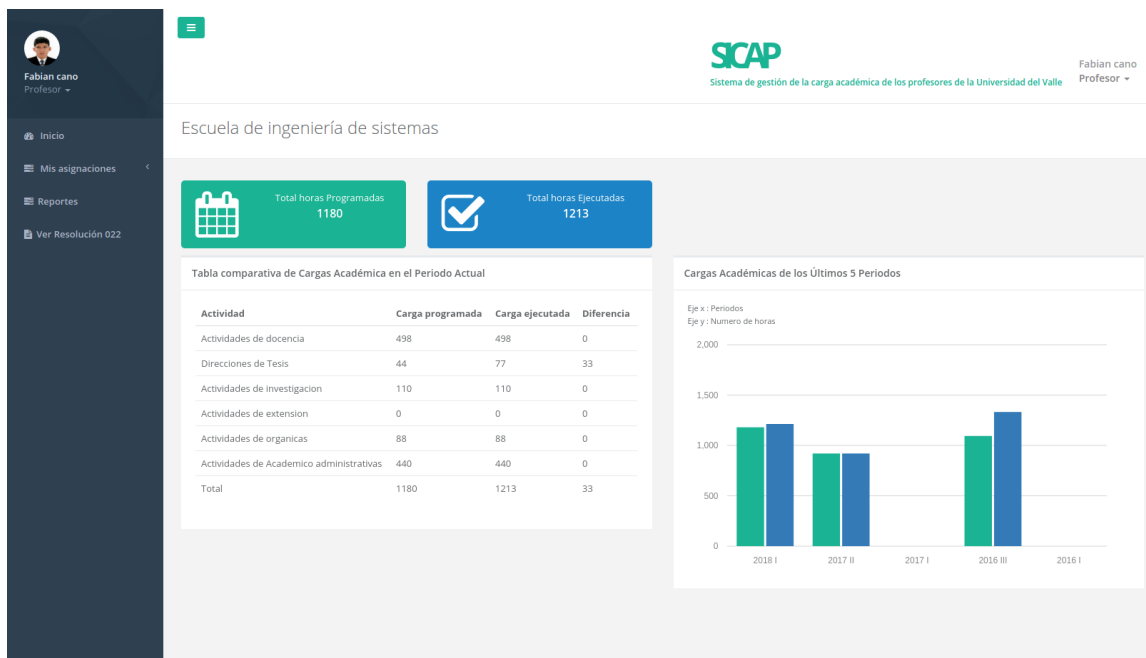


Ilustración 12: Dashboard profesor

6.1.1 Usuario Profesor

Para empezar con la gestión de la carga académica el profesor puede ir al registro de cargas y crear una nueva carga. Posteriormente se despliega la página de gestión de carga académica en donde podrá adicionar, modificar o eliminar las diferentes actividades y ver el número de horas registradas por tipo de actividad y del total de la carga como se muestra en la Ilustración 13.

Edición carga académica

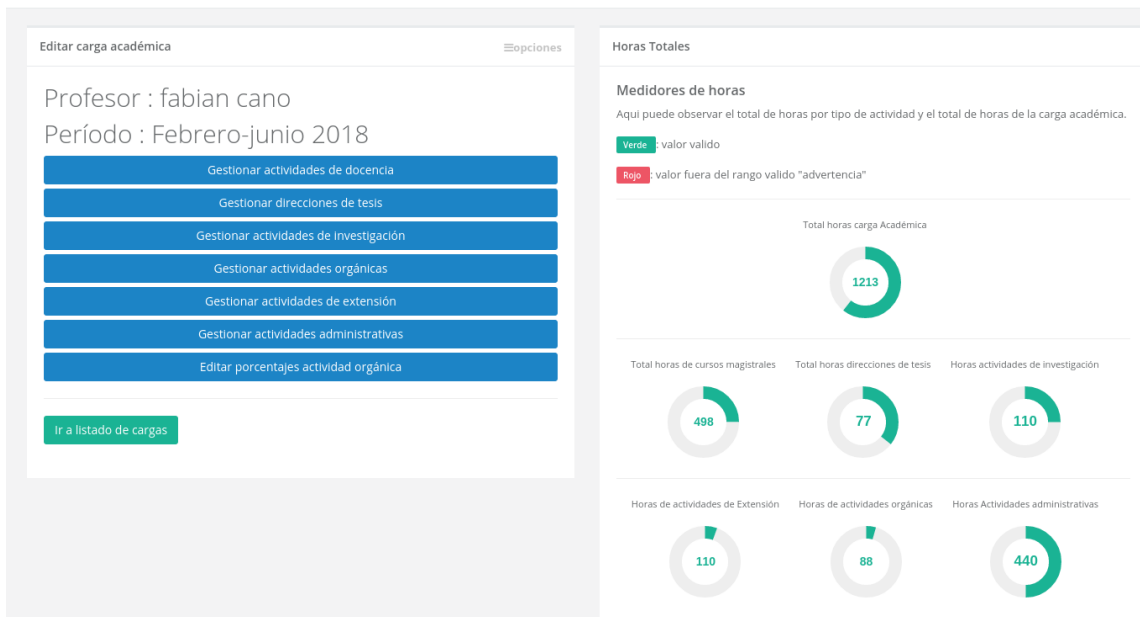


Ilustración 13: Edición carga académica

Para adicionar modificar o eliminar actividades se da clic sobre el respectivo botón, al hacerlo se despliega la página de gestión del tipo de actividad seleccionada. Una vez se despliega la página el profesor puede adicionar una nueva actividad llenando el formulario de la parte izquierda y dando clic en el botón “registrar”, también puede editar o eliminar actividades registradas anteriormente dando clic en el botón “opciones” de la tabla izquierda y dando clic en la opción que desee realizar. La Ilustración 14 muestra la gestión de actividades de docencia.

Gestión actividades de docencia

Registrar actividad de docencia

Curso
Fundamentos de programación

Tipo
Curso magistral

Grupo
01

Numero de estudiantes
35

Monitor

Dedicación
0% 100%

0 25 50 75 100

[Registrar](#) [Limpiar](#) [Ir a edición carga](#)

Actividades de docencia creadas Total:498 horas

Mostrar 10 registros Buscar:

Nombre	Grupo	Tipo	Opciones
Pocesamiento de lenguaje natural	01	magistral	Opciones
Fundamentos de programacion	01	magistral	Opciones

Mostrando registros del 1 al 2 de un total de 2 registros [Anterior](#) 1 [Siguiente](#)

Ilustración 14: Gestión actividades de docencia

Para proponer una carga el profesor se debe dirigir al listado de carga académicas y en la tabla que aparece dar clic en “opciones” y posteriormente en “Proponer” así la carga académica queda propuesta para que sea revisada por el director de escuela. La Ilustración 15 muestra este proceso.

Fabian Cano
Profesor

- Inicio
- Mis asignaciones
- Reportes
- Ver Resolución 022

SCAP
Sistema de gestión de la carga académica de los profesores de la Universidad del Valle

Fabian Cano
Profesor

Lista de cargas académicas Ejecutadas de Fabian Cano

No propuestas Propuestas Rechazadas (0) Aceptadas

Mostrar 10 registros Buscar:

Periodo	Profesor	Horas/semestre	Opciones
2018 I	12412421	1191	Opciones
2017 II	12412421	920	Opciones

Mostrando registros del 1 al 2 de un total de 2 registros [Anterior](#) 1 [Siguiente](#)

[Opciones](#)

- Detalles
- Editar
- Proponer carga

Ilustración 15: Proponer carga

Así mismo si el profesor da clic en el botón “detalles” se desplegará una página (ver Ilustración 16) con la información detallada de la carga académica seleccionada como son: los datos de las actividades y el total de horas.

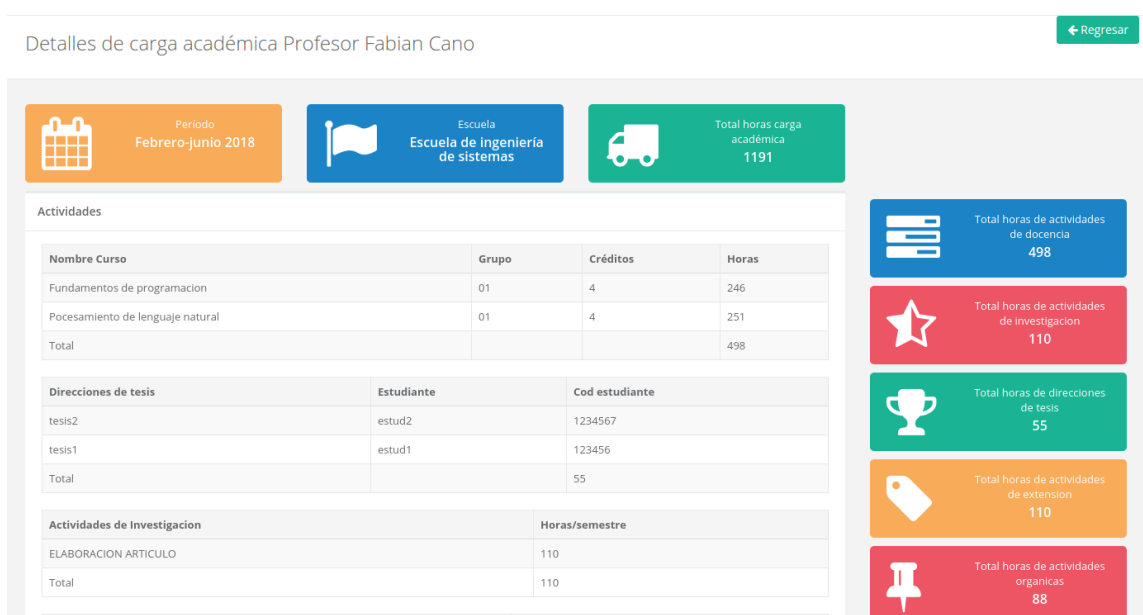


Ilustración 16: Detalles carga académica

6.1.2 Usuario Director

Las funciones del director consisten en revisar las cargas académicas de los profesores de la escuela con el fin de verificar que no haya errores en los datos de las actividades y que estas correspondan con información verídica; Para esto el director al ingresar puede seleccionar en su página de inicio con qué rol desea continuar. Si escoge continuar como profesor tendrá las mismas funcionalidades de un Usuario Profesor y si escoge continuar como director tendrá las funcionalidades del Usuario director que consta de revisar las

carga académicas de los profesores y aprobarlas o rechazarlas según sea el caso. La

Ilustración 17 muestra la selección del rol.

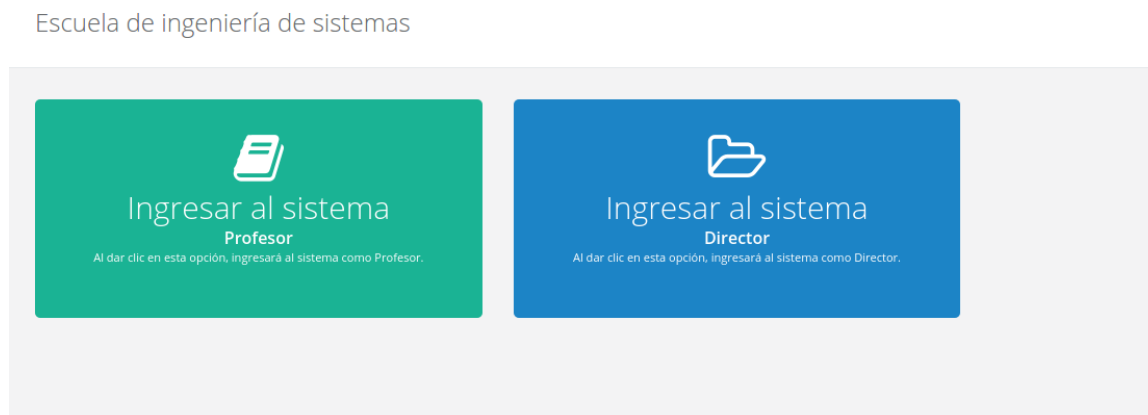


Ilustración 17: Pagina de elección de rol de usuario

Para revisar una carga académica el director da clic en la opción “detalles” de la carga seleccionada. Una vez revisada el director puede aprobar o rechazar la carga dando clic en la opción “Aceptar/Rechazar”, esto desplegará una ventana modal en la que podrá escoger la opción deseada, Si el director acepta la carga será enviada a la vicedecanatura para que el vicedecano compruebe la información y si la rechaza será enviada de nuevo al profesor para que este realice las correcciones necesarias; el director también puede realizar las respectivas correcciones si así lo desea. Las Ilustraciones 18 y 19 muestran el proceso de Aprobación y rechazo de cargas académicas

Escuela de ingeniería de sistemas

Profesores nombrados sin carga académica 4 Ver más

8 Profesores nombrados 4 Cargas programadas por aprobar 0 Cargas programadas rechazadas

2 Profesores no nombrados 0 Cargas ejecutadas por aprobar 0 Cargas ejecutadas rechazadas

10 Total profesores 4 Total pendientes 0 Total rechazadas

Programadas Ejecutadas

Mostrar 10 registros Buscar:

Profesor	Periodo	Aprobación Director	Aprobación Decano	Opciones
Yallie Bermudez	2018 I	Sin definir	Sin definir	Opciones
sergio Henao	2018 I	Sin definir	Sin definir	Ver más
jose bañon	2018 I	Sin definir	Sin definir	Detalles
Jesus Alexander Aranda Bueno	2018 I	Sin definir	Sin definir	Editar
Fabian Cano	2018 I	Aprobada	Aprobada	Aceptar/Rechazar

Mostrando registros del 1 al 5 de un total de 5 registros

Anterior 1 Siguiente

Ilustración 18: Aceptar/Rechazar carga

Aceptar o rechazar carga académica:

si acepta la carga quedara aprobada para ejecutarse; si la rechaza sera enviada para corregirse

Cancelar Aceptar Rechazar

Ilustración 19: Modal de confirmación Aceptar/rechazar

6.1.3 Usuarios Vicedecano y Decano

Los usuarios vicedecano y decano pueden aceptar o rechazar cargas académicas de las escuelas de la facultad. En el caso de vicedecano solo puede visualizar las cargas aceptadas por los directores de escuela y aceptarlas o rechazarlas. Posteriormente el decano es el que visualiza las carga aceptadas o rechazadas por el vicedecano da la aceptación o el rechazo definitivo, si acepta la carga académica queda consolidada y lista para ejecutarse y si la rechaza será enviada al director de escuela correspondiente para que realice las correcciones o la rechace y enviarla de nuevo al profesor. Los pasos para la aceptación y rechazo de las cargas académicas son los mismos del director.

6.1.4 Usuario Operador

El Operador de oficina es un asistente, un funcionario de la universidad que tiene las mismas funcionalidades de un profesor solo que este puede escoger a qué profesor le va a gestionar la carga académica. Puede escoger entre todos los profesores de la escuela a la que pertenece. El operador puede llenar la carga académica pero debe contar con la asesoría del respectivo profesor.

Los directores, operadores de oficina, vicedecanos y decanos pueden visualizar todo los reportes que ofrece la aplicación. Las Ilustraciones 20, 21 y 22 muestran ejemplos de los reportes.

Reportes

Cargos por profesor		
Périodo	Profesor	Cargo
2016 I	Victor Andres Bucheli Guerrero	Director de escuela
2016 I	Raúl Ernesto Gutierrez de piñeres Reyes	Director de programa
2016 I	Pedro Moreno	Representante profesoral consejo academico
2016 I	Pedro Moreno	Jefe de departamento
2016 I	Pedro Moreno	Jefe de departamento

Ilustración 20: Reporte Cargos profesores

Reportes

Reporte cargas																			
PROFESOR	CAT	DED	T.V.	CURSOS						NO ESTUD EN TESIS				ACTIVIDAD ACADÉMICO ADMINISTRATIVA	H/SEM	ACTIVIDADES DE INVESTIGACIÓN Y DILLO TECNOLÓGICO	TIPO I/D/P/F	H/SEM	OTRAS ACTIVIDAD
				NOMBRE	GR	CODIGO	NIVEL	#EST	H/SEM	PRE	MAG	DOC	H/SEM						
Daniel Barragan	AUX	TP	VR	Algoritmia y programacion	01	750001M	PRE	31	210	1	0	0	22					organicas	
				Métodos numéricos	02	750017M	PRE	30	210										
Jose Bañon	TIT	TC	NOMB							0	0	0	0					incapacidad	
Oscar Bedoya	AUX	TC	NOMB	Algoritmia y programacion	08	750001M	PRE	25	199	0	0	0	0		ELABORACION ARTICULO	P	110	organicas	
				Sistemas Operativos	02	750008M	PRE	35	216					ELABORACION ARTICULO	P	110			
				Introducción a la inteligencia artificial	01	750033M	PRE	50	243										
Luz Estela MUñoz	AUX	HR	HR	Algoritmia y programacion	02	750001M	PRE	31	210	0	0	0	0						
				Algoritmia y programacion	07	750001M	PRE	29	151										
				Introducción a la tecnología informática	22	750082M	PRE	34	84										
				Introducción a la tecnología informática	01	750082M	PRE	47	109										

Ilustración 21: Reporte General cargas académicas


Información carga academica 				
Unidad academica				
Escuela de ingenieria de sistemas				
Cédula	Nombres		Apellidos	
6106787	Oscar		Bedoya	
Vinculacion	Categoria	Dedicacion		
NOMB	AUX	TC		
Código	Grupo	Nombre asignatura	Creditos	H/sem
750001M	08	Algoritmia y programacion	3	199
750008M	02	Sistemas Operatovos	3	216
750033M	01	Introducción a la inteligencia artificial	4	243
Código estudiante	Plan	Nombre tesis	H/sem	
Nombre actividad	Tipo	Bonificada	H/sem	
ELABORACION ARTICULO	P	si	110	
ELABORACION ARTICULO	P	si	110	

Ilustración 22: Reporte carga académica profesor

6.1.5 Usuario Administrador

El administrador del sistema es el encargado de registrar a los usuarios y darles su rol en el sistema, también es quien registra las asignaturas, las escuelas, los programas, y los periodos; necesarios para poder realizar la gestión de la carga académica.

Capítulo 7

Validaciones

En este capítulo se muestra las validaciones realizadas a la aplicación SICAP. En primer lugar se validó la aplicación replicando la asignación de carga académica de un periodo pasado, posteriormente se validó a través de una encuesta a profesores de la escuela sobre la usabilidad de la aplicación.

7.1 Validación con periodo académico pasado

Para esta validación se registró las cargas académicas de los profesores de la EISC del periodo Febrero-junio de 2016 en la aplicación. Una vez registradas las cargas y sus respectivas actividades se procedió a comparar el número de horas de las actividades y el total de número de horas de la carga académica calculados por la aplicación con una tabla que corresponde al resultado de los cálculos realizado por la aplicación SSARA para el mismo periodo y los mismos profesores de esta manera comprobar que la aplicación está realizando los cálculos como está establecido en la resolución 022. A continuación se muestran los resultados.

Los cálculos matemáticos obtenidos en la aplicación son iguales a los esperados por lo que se concluye que el modelo quedo bien construido. Los siguientes son 5 ejemplos de actividades de docencia registrados en la aplicación.

Se compara el resultado obtenido con el resultado de la aplicación que se está utilizando actualmente. También se realizó el cálculo en una hoja de cálculo de Excel a manera de verificación.

Actividades de docencia

Tabla 4: Actividades de docencia

	Curso	Grupo	Número de estudiantes	Dedicación	Monitor
Ejemplo 1	Matemáticas discretas II	02	30	100%	Sin monitor
Ejemplo 2	Fundamento de análisis y diseños de algoritmos	01	55	100%	Sin monitor
Ejemplo 3	Complejidad y optimización	01	50	100%	Sin monitor
Ejemplo 4	Algoritmia y programación	08	25	100%	Sin monitor
Ejemplo 5	Sistemas operativos	02	35	100%	Sin monitor

Resultados

Tabla 5: Resultados de los cálculos de las actividades de docencia

	Número de horas según la aplicación SSARA(aplicación actual)	Número de horas según la aplicación SICAP(aplicación propuesta)	Número de horas según la hoja de cálculo de Excel
Ejemplo 1	241	241	241
Ejemplo 2	279	279	279
Ejemplo 3	273	273	273
Ejemplo 4	199	199	199
Ejemplo 5	216	216	216

7.2 Validación usando encuesta

Se realizó una encuesta sobre la usabilidad de la aplicación. Esta encuesta se le realizó a un grupo de 7 profesores y a la secretaria encargada de la escuela de ingeniería de sistemas y computación los cuales interactuaron con la aplicación intentando registrar la carga académica que realizaran el periodo Febrero-Junio de 2017. Con base en esto se les preguntó sobre su experiencia usando la aplicación. A continuación se muestran las preguntas realizadas y sus resultados.

Tabla 6: Preguntas usadas en la encuesta

Pregunta	Valor
Pregunta 1	¿Entiende el propósito general de la aplicación?
Pregunta 2	¿La aplicación es fácil de usar?
Pregunta 3	¿Los mensajes y los títulos de los enlaces le permiten saber que funcionalidad realizara o realizó?
Pregunta 4	¿Los colores de la aplicación le parecen adecuados?
Pregunta 5	¿Los componentes de medición mostrados en la página de edición de carga son claros?
Pregunta 6	¿Se entiende el significado de cada uno de los campos de los formularios de registro de las actividades?
Pregunta 7	¿La distinción entre carga programada y carga ejecutada es clara en la aplicación?
Pregunta 8	¿Los reportes visualizados presentan la información correctamente?

Para las respuesta se utilizó la escala de Likert (Llauradó, 2014) la cual permite medir el nivel de conformidad del encuestado con respecto a los que se le pregunta. Las posibles respuestas por cada pregunta se muestran a continuación

Tabla 7: Escala de Likert

Totalmente de acuerdo
De acuerdo
Ni de acuerdo ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

Siguiendo esta escala se muestran las los resultados de la encuesta realizada

¿Entiende el propósito principal de la aplicación? (8 respuestas)

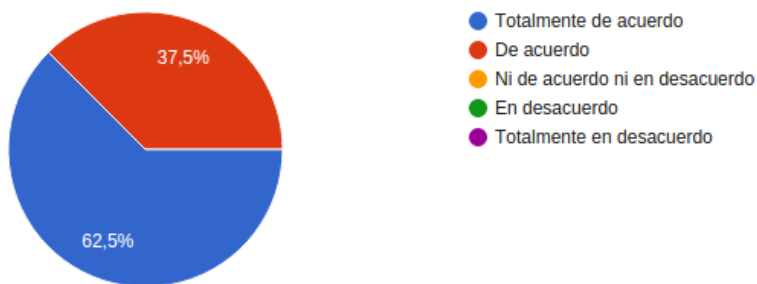


Ilustración 23: Resultados pregunta 1

¿La aplicación es fácil de usar? (8 respuestas)

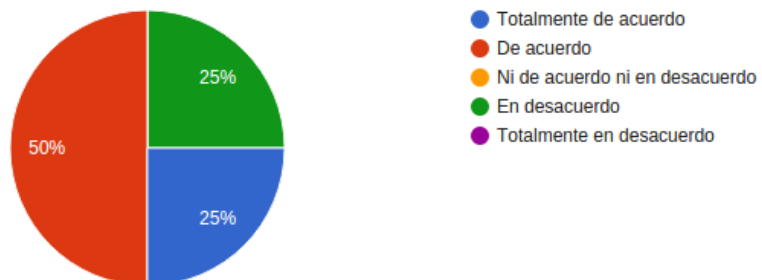


Ilustración 24: Resultados pregunta 2

¿Los mensajes y los títulos de los enlaces le permiten saber que funcionalidad realizará o realizó ?

(8 respuestas)

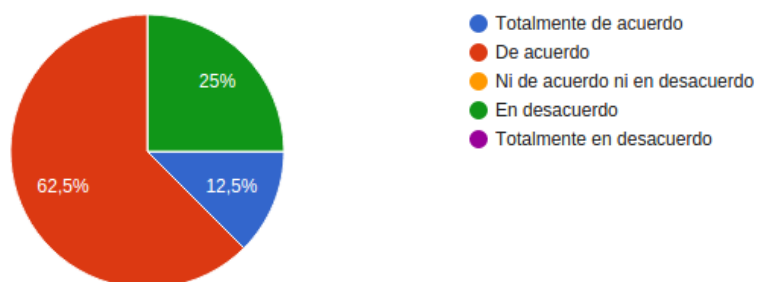


Ilustración 25: Resultados pregunta 3

¿Los colores de la aplicación le parecen adecuados ? (8 respuestas)

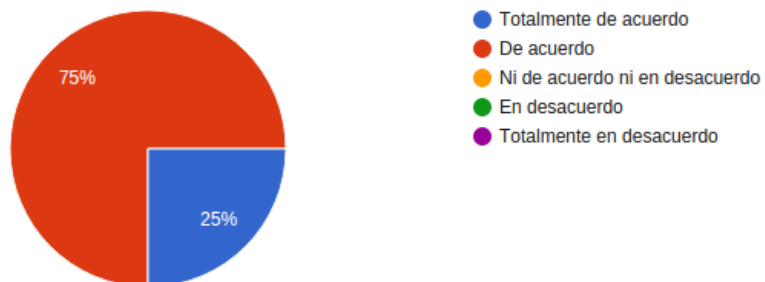


Ilustración 26: Resultados pregunta 4

¿Los componentes de medición mostrados en la página de edición de carga son claros ? (8 respuestas)

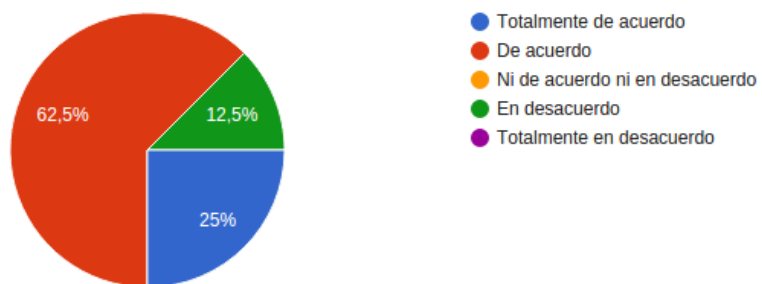


Ilustración 27: Resultados pregunta 5

¿Se entiende el significado de cada uno de los campos del formulario de registro de las actividades?

(8 respuestas)

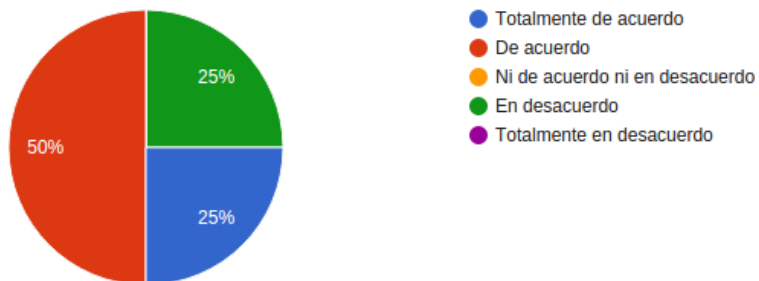


Ilustración 28: Resultados pregunta 6

¿La distinción entre carga programada y carga ejecutada es clara en la aplicación?

(8 respuestas)

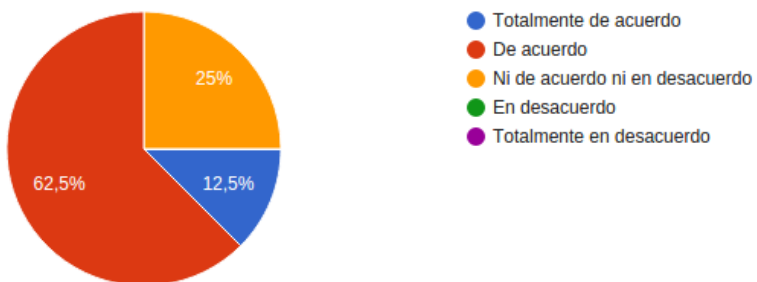


Ilustración 29: Resultados pregunta 7

¿Los reportes visualizados presentan la información correctamente?

(8 respuestas)

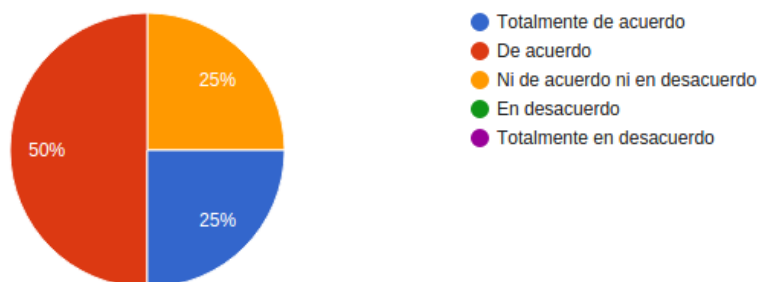


Ilustración 30: Resultados pregunta 8

Puesto que la opción “De acuerdo” fue la que más se seleccionó se puede concluir que la aplicación tuvo una buena respuesta por parte de los usuarios y cumple con el objetivo principal. Las personas encuestadas manifestaron que la aplicación era entendible y comprendieron cual era el propósito de esta. El 75% de los encuestados manifestaron que la aplicación era fácil de usar y que los colores eran adecuado o agradables a la vista. Sobre los procesos de gestión de la carga académica la mayoría de los encuestados (75%) mostraron que comprendían el significado de los datos que requerían ingresar. También confirmaron que los reportes de la aplicación mostraban la información requerida.

Capítulo 8

Conclusiones y trabajo futuro

8.1 Conclusiones

La gestión de la carga académica de los profesores es un proceso importante y complejo el cual requiere de un manejo adecuado por parte de los involucrados por lo que los sistemas computarizados ofrecen la facilidad, la seguridad y el control tanto de los datos como de quien puede alterarlos.

Involucrar a los profesores en el proceso de gestión de la carga académica acelera los tiempos de gestión, además de que se tiene mejor conocimiento y control por parte de los profesores de cómo está distribuida su carga académica a los largo de los diferentes periodos académicos.

Según las pruebas realizadas, el modelo diseñado cumple con las normas establecidas por la resolución 022 del 2001 del consejo superior de la Universidad del valle, lo que muestra que la abstracción del modelo es correcta.

El hecho de que la aplicación implementada sea web facilita su divulgación por toda la Universidad y la estandarización en el código facilita su mantenimiento.

8.2 Trabajo futuro

Interoperar la aplicación con el sistema de registro académico para para que la información sobre los cursos las dependencias y el número de estudiantes matriculados pueda extraerse de este sistema y así tener información actualizada y fiable que mejore aún más la automatización de ciertas funcionalidades de la aplicación.

Implementar una funcionalidad que permita marcar los errores o inconsistencias en las diferentes actividades de una carga académica cuando el director, el vicedecano o el decano están en el proceso de revisión.

Implementar firmas digitales al momento de aprobar una carga académica para evitar que se tenga que firmar los reportes impresos y así disminuir el uso de papel y acelerar el proceso de aprobación de las cargas.

Lista de referencias

Brat Tech LLC, Google and community. (2009).AngularJs by Google. EE.UU. Retrieved from <https://angularjs.org/>

Consejo superior Universidad del Valle. (2001). Resolución No. 022 [PDF]. Retrieved from <http://eiee.univalle.edu.co/Archivos/CS-022.pdf>

Django Software foundation. (2005-2016).Django. Kansas EE.UU. Retrieved from <https://www.djangoproject.com/>

Ethan Marcotte (2010). Diseño Web Responsivo (Responsive Web Design). Retrieved from http://xn--diseowebresponsivo-q0b.org/?utm_source=redirects&utm_medium=dise%25C3%25B1owebresponsivo.com.a&utm_campaign=301_Redirects

Lenis, M. (2015). eXpert herramienta web como apoyo para la adopción de prácticas ágiles usando como marco de referencia la metodología ágil extreme programming(XP). Universidad Del Valle. Retrieved from <http://mapsoft.cedesoftexpert.com>

Mark Otto. (2011-2016). Bootstrap from Twitter. Twiter Blogs. Retrieved from <https://blog.twitter.com/2011/bootstrap-from-twitter>

Mark Otto y Jacobd Thornton de Twitter. (2011-2016).Bootstrap.San Francisco EE.UU. Retrieved from <http://getbootstrap.com/>

Mimosa software Ltd. (2016). Mimosa scheduling software. YLOJARVI FINLAND. Retrieved from <http://www.mimosasoftware.com/>

MorrisJs (2016). Morris.js good-looking charts shouldn't be difficult. Retrieved from <http://morrisjs.github.io/morris.js/>

NANYANG TECHNOLOGICAL UNIVERSITY. (2012.). Academic Unit System. SINGAPORE. Retrieved from [http://www.ntu.edu.sg/Students/Undergraduate/AcademicServices/Pages/AcademicUnitSystem\(AUS\).aspx](http://www.ntu.edu.sg/Students/Undergraduate/AcademicServices/Pages/AcademicUnitSystem(AUS).aspx)

Oriol Llauradó (2014). La escala de Likert: Qué es y como utilizarla. Retrieved from <http://www.netquest.com/blog/es/la-escala-de-likert-que-es-y-como-utilizarla/>

PWC. (2010- 2016).Academic workload management system. Victoria, Australia. Retrieved from <http://www.pwc.com.au/education/academic-workload.html>

Python Software Foundation. (1991-2016). Python. Paises bajos. Retrieved from <https://www.python.org/about/>

The PostgreSQL Global Development Group. (1996-2016).PostgreSQL. University of California, Berkeley. Retrieved from <http://www.postgresql.org/about/>

University of Exeter. (2014-2015).Academic workload planning. Gran Bretaña. Retrieved from <http://www.exeter.ac.uk/staff/employment/workloadplanning/>

Web, L. (2016). El patrón de diseño MTV. Retrieved from http://librosweb.es/libro/django_1_0/capitulo_5/el_patron_de_diseno_mtv.html

Wrapbootstrap.(2016). Plantilla INSPINIA. Retrieved from <http://wrapbootstrap.com/preview/WB0R5L90S>

W3C. (1994–2016). Cascading Style Sheets home page. Francia. Retrieved from <http://www.w3.org/Style/CSS/>

W3C. (2014–2016). HTML, The Web’s Core Language. Francia. Retrieved from <http://www.w3.org/html/>