



ESCUELA DE INGENIERÍA EN SISTEMAS

Tema:

“APLICACIÓN WEB PARA LA GENERACIÓN DE SOLICITUDES
ACADÉMICAS DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL
ECUADOR SEDE AMBATO”

**Proyecto de investigación y desarrollo previo a la obtención
del título de Ingeniero de Sistemas y Computación**

Líneas de Investigación:

Ingeniería de Software y/o Plataformas Educativas

Autor:

MARIO ALEJANDRO MORENO MONTERO

Director:

Mg. ENRIQUE XAVIER GARCÉS FREIRE

Ambato – Ecuador

Mayo – 2017

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

SEDE AMBATO

HOJA DE APROBACIÓN

Tema:

“APLICACIÓN WEB PARA LA GENERACIÓN DE SOLICITUDES
ACADÉMICAS DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL
ECUADOR SEDE AMBATO”

Línea de Investigación:

Ingeniería de Software y/o Plataformas Educativas

Autor:

MARIO ALEJANDRO MORENO MONTERO

Enrique Xavier Garcés Freire, Ing. Mg. f. _____

CALIFICADOR

Santiago Alejandro Acurio Maldonado, Ing. Mg. f. _____

CALIFICADOR

Galo Mauricio López Sevilla, Ing. Mg. f. _____

CALIFICADOR

Teresa Milena Freire Aillón, Ing. Mg. f. _____

DIRECTORA DE LA ESCUELA DE

INGENIERÍA EN SISTEMAS

Hugo Rogelio Altamirano Villarroel, Dr. f. _____

SECRETARIO GENERAL PUCESA

Ambato – Ecuador

Mayo – 2017

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Mario Alejandro Moreno Montero portador de la cédula de ciudadanía No.180452381-7 declaro que los resultados obtenidos en la investigación que presento como informe final, previo la obtención del título de Ingeniero de Sistemas y Computación son absolutamente originales, auténticos y personales.

En tal virtud, declaro que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo propuesto de investigación y luego de la redacción de este documento son y serán de mi sola y exclusiva responsabilidad legal y académica.

Mario Alejandro Moreno Montero

CI. 180452381-7

AGRADECIMIENTO

Quiero agradecer a mis padres Mario Moreno y Alexandra Montero por darme la oportunidad de realizar mis estudios y por su apoyo constante.

Un sincero agradecimiento a los docentes de la Escuela de Ingeniería en sistemas por aportar cada uno en mi formación académica y de vida.

Finalmente agradezco a las personas que fueron parte de este proyecto, especialmente a mi tutor y amigo Enrique Garcés, por su apoyo durante el mismo, por hacer que mire al mundo y las oportunidades de manera diferente y por enseñarme que nunca se deja de aprender.

Mario Alejandro Moreno Montero

RESUMEN

El objetivo del presente proyecto es desarrollar una aplicación web para la generación de solicitudes académicas de la PUCESA. El autor mediante observación directa ha logrado evidenciar el largo proceso que los estudiantes siguen para realizar las solicitudes y que sean aceptadas por las secretarías de escuela. La aplicación cumple con ciertos estándares de usabilidad y es accesible desde cualquier dispositivo tecnológico sea un teléfono móvil, tableta, laptop o computadora de escritorio, garantizando una buena experiencia de usuario, la misma que informa cuál es el proceso que la solicitud sigue para ser aprobada luego que el estudiante la presenta. Los sistemas informáticos avanzan de manera rápida, es el caso de este tipo de aplicaciones que han ido evolucionando en cuanto a las metodologías, *frameworks* y lenguajes que se pueden ocupar para su desarrollo. Para su elaboración se utilizó un lenguaje muy conocido y usado en la actualidad llamado Python con el *framework* de desarrollo escrito en el mismo lenguaje de nombre Django. La metodología manejada fue “Extreme Programming” elegida por ser ágil y la metodología de campo a través del diálogo con los futuros usuarios para tener como resultado SolicitudesApp una aplicación que demostró ser de gran ayuda a los estudiantes de la PUCESA.

Palabras clave: aplicación web, extreme programming, Django, Python.

ABSTRACT

The objective of this project is to develop a web application for automatic generation of academic requisitions at PUCESA. The author by means of direct observation has evidenced the long process that students must follow to complete requests and the submission process by the department secretaries. The application meets certain usability standards and can be accessed from any technological device such as: a mobile phone, tablet, laptop, or desktop computer, which guarantees a good user experience and it would keep the student posted on what is the next step for the requirements to be accepted. Information systems advance rapidly; such is the case of this type of applications that have evolved when it comes to methodologies, frameworks, and languages that can be used for their development. Python, a well-known and widely used language, was applied for its development, along with the development of Django framework, which is written in the same language. Extreme Programming was used as a methodology which was chosen because it is fast; and field methodology was performed by talking to the future users. As a result SolicitudesApp is an application that demonstrated to be of great help for PUCESA students.

Key words: web application, extreme programming, Django, Python.

ÍNDICE DE CONTENIDOS

PRELIMINARES

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD.....	iii
AGRADECIMIENTO	iv
RESUMEN.....	v
ABSTRACT.....	vi
ÍNDICE DE CONTENIDOS.....	vii
ÍNDICE DE GRÁFICOS.....	xi

INTRODUCCIÓN.....	1
CAPÍTULO I.....	2
FUNDAMENTOS TEÓRICOS.....	2
1.1 Antecedentes.....	2
1.2 Problema.....	3
1.2.1 Descripción del problema.....	3
1.2.2 Preguntas básicas.....	4
1.3 Justificación.....	4
1.4 Objetivos.....	5
1.4.1 General.....	5
1.4.2 Específicos.....	5
1.5 Fundamentos teóricos.....	6
1.5.1 Ingeniería de software.....	6
1.5.2 Aplicaciones web.....	7
1.5.3 Programación web.....	8
1.5.3.1 HTML5.....	9
1.5.3.2 CSS.....	10
1.5.3.3 JavaScript.....	11
1.5.3.4 Python.....	12
1.5.3.5 Diseño web responsivo.....	13
1.5.3.6 Bootstrap.....	14
1.5.3.7 PyCharm.....	14
1.5.4 Frameworks de desarrollo web.....	14

1.5.4.1 Django.....	15
1.5.4.2 Modelo-Plantilla-Vista	16
1.5.5 Base de Datos	17
1.5.5.1 PostgreSQL	18
1.5.5.2 Mapeo Objeto-Relacional.....	18
1.5.6 Solicitudes académicas.....	19
1.5.6.1 Formato PDF (Portable Document Format)	19
1.5.6.2 Procesos	20
1.5.7 Pontificia Universidad Católica del Ecuador Sede Ambato	20
CAPÍTULO II	21
METODOLOGÍA	21
2.1 Metodología de la investigación.....	21
2.1.1 Metodología analítica	21
2.1.2 Técnicas e instrumentos de recolección de datos	22
2.1.2.1 Entrevista	22
2.1.2.1.1 Realización de entrevistas	23
2.1.2.2 Focus group	23
2.1.2.2.1 Focus group realizado.....	23
2.2 Metodología de desarrollo de software	24
2.2.1 Metodología ágil.....	24
2.2.2 Programación Extrema (Extreme Programming, XP)	25
2.2.2.1 Fase 1: Planeación	26
2.2.2.1.1 Historias de usuario	26
2.2.2.1.2 Plan de iteración	27
2.2.2.2 Fase 2: Diseño	27
2.2.2.2.1 Diseño simple	27
2.2.2.2.2 Tarjetas CRC	28
2.2.2.2.3 Prototipos.....	28
2.2.2.3 Fase 3: Codificación	29
2.2.2.3.1 Programación.....	29
2.2.2.3.2 Rediseño.....	29
2.2.2.4 Fase 4: Prueba	30
2.2.2.4.1 Prueba unitaria.....	30
2.2.2.4.2 Prueba de aceptación	30

CAPÍTULO III	31
RESULTADOS.....	31
3.1 Software y licencias	31
3.2 Fase I: Planeación	32
3.2.1 Entrevista	32
3.2.2 Focus group	35
3.2.3 Historias de usuario	38
3.2.3.1 Planificación de desarrollo de Historias de Usuario	43
3.2.3.2 Planificación de desarrollo de Historias de Usuario	44
3.2.3.3 Tiempo calendario planificado	44
3.2.3.4 Esfuerzo de desarrollo en base a un programador	44
3.2.3.5 Estimación de entregables.....	45
3.3 Fase II: Diseño.....	46
3.3.1 Módulos de la aplicación.....	46
3.3.1.1 Metáfora de uso de la aplicación	46
3.3.1.2 Metáfora de administrador	47
3.3.1.3 Metáfora de solicitudes	47
3.3.1.4 Metáfora de cuenta estudiante.....	47
3.3.2 Diagrama de clases	48
3.3.3 Tarjetas Clase-Responsabilidad-Colaboración (CRC).....	49
3.3.4 Diseño arquitectónico	50
3.3.5 Diseño de interfaz	52
3.4 Fase III: Codificación	53
3.4.1 Estándares de codificación	53
3.4.2 Cronograma de actividades	54
3.4.3 Codificación de historias de usuario	57
3.4.3.1 Módulo: Base de datos	57
3.4.3.1.1 Historia de usuario: Base de datos.	57
3.4.3.2 Módulo: Uso de aplicación.....	58
3.4.3.2.1 Historia de usuario: Formulario de ingreso/registro	58
3.4.3.2.2 Historia de usuario: Utilización del servicio	61
3.4.3.3 Módulo: Administrador	62
3.4.3.3.1 Historia de usuario: Ingresar/modificar solicitudes	62
3.4.3.3.2 Historia de usuario: Modificar datos carrera/escuela	62

3.4.3.4 Módulo: Solicitudes	63
3.4.3.4.1 Generar solicitud	63
3.4.3.4.2 Visualizar solicitud.....	64
3.4.3.5 Módulo: Cuenta estudiante	65
3.4.3.5.1 Modificar contraseña/recuperar cuenta.....	65
3.4.3.5.2 Modificar/eliminar usuario	66
3.4.3.5.3 Cierre de sesión	68
3.5 Fase IV: Prueba	69
3.5.1 Pruebas unitarias	69
3.5.1.1 Prueba unitaria: Registro de usuario estudiante	69
3.5.1.2 Prueba unitaria: Login de usuario estudiante.....	70
3.5.1.3 Prueba unitaria: Modificación de perfil de usuario estudiante	71
3.5.1.4 Prueba unitaria: Cambio de contraseña de usuario estudiante.....	72
3.5.1.5 Prueba unitaria: Eliminar cuenta de usuario estudiante	72
3.5.1.6 Prueba unitaria: Generación de solicitud	73
3.5.2 Pruebas de aceptación	73
3.6 Acceso a la aplicación	77
CAPÍTULO IV.....	78
ANÁLISIS Y VALIDACIÓN DE RESULTADOS.....	78
4.1 Análisis de resultados	78
4.1.1 Aplicación web con diseño responsivo	78
4.1.2 SolicitudesApp	79
4.2 Validación de resultados	113
4.2.1 Encuesta de satisfacción	113
4.2.1 Entrevista a usuario administrador.....	115
CONCLUSIONES Y RECOMENDACIONES	118
CONCLUSIONES	118
RECOMENDACIONES.....	120
BIBLIOGRAFÍA.....	122
ANEXOS.....	125

ÍNDICE DE GRÁFICOS

Ilustraciones

Ilustración 1.1: Infografía ilustrativa de Responsive Design.....	13
Ilustración 1.2: Generaciones del desarrollo web	15
Ilustración 1.3: Funcionamiento del MTV de Django.....	17
Ilustración 2.1: Proceso de la Programación Extrema	26
Ilustración 3.1: Diagrama de clases	48
Ilustración 3.2: Diseño arquitectónico de Django	51
Ilustración 3.3: Diseño de interfaz.....	53
Ilustración 3.4: Cronograma de actividades 1	55
Ilustración 3.5: Cronograma de actividades 2	56
Ilustración 4.1: Pantalla de inicio vista móvil	79
Ilustración 4.2: Pantalla de inicio vista de tableta.....	79
Ilustración 4.3: Pantalla de inicio vista de laptop.....	80
Ilustración 4.4: Registro de usuario vista móvil	80
Ilustración 4.5: Registro de usuario vista tableta.....	81
Ilustración 4.6: Registro de usuario vista laptop.....	81
Ilustración 4.7: Menú de visitantes vista móvil	82
Ilustración 4.8: Menú de visitantes vista tableta.....	82
Ilustración 4.9: Menú de visitantes vista laptop.....	82
Ilustración 4.10: Inicio de sesión incorrecto	83
Ilustración 4.11: Menú de solicitudes de usuario vista móvil.....	84
Ilustración 4.12: Menú de solicitudes de usuario vista tableta	84
Ilustración 4.13: Menú de solicitudes de usuario vista laptop	84
Ilustración 4.14: Formulario de solicitud básica visitante vista móvil.....	85
Ilustración 4.15: Formulario de solicitud básica visitante vista tableta	85
Ilustración 4.16: Formulario de solicitud básica visitante vista laptop	86
Ilustración 4.17: Formulario de solicitud con campos adicionales visitante vista móvil	86
Ilustración 4.18: Formulario de solicitud con campos adicionales visitante vista tableta.....	87
Ilustración 4.19: Formulario de solicitud con campos adicionales visitante vista laptop.....	87
Ilustración 4.20: Formulario de solicitud básica usuario vista móvil	88
Ilustración 4.21: Formulario de solicitud básica usuario vista tableta.....	88
Ilustración 4.22: Formulario de solicitud básica usuario vista laptop.....	88
Ilustración 4.23: Formulario de solicitud con campos adicionales usuario vista móvil	89
Ilustración 4.24: Formulario de solicitud con campos adicionales usuario vista tableta	89
Ilustración 4.25: Formulario de solicitud con campos adicionales usuario vista laptop	90

Ilustración 4.26: Formulario de solicitud con campo adicional de inglés/pastoral visitante.....	90
Ilustración 4.27: Formulario de solicitud básica de inglés/pastoral usuario..	90
Ilustración 4.28: Solicitud vista móvil	91
Ilustración 4.29: Solicitud vista tableta.....	91
Ilustración 4.30: Solicitud vista laptop	92
Ilustración 4.31: Opción de Abrir en iBooks en iPhone	92
Ilustración 4.32: Opción Más en iPhone	93
Ilustración 4.33: Solicitud descargada en dispositivo Android	93
Ilustración 4.34: Solicitud descargada en Laptop.....	93
Ilustración 4.35: Modificar usuario vista móvil.....	94
Ilustración 4.36: Modificar usuario vista tableta	94
Ilustración 4.37: Modificar usuario vista laptop	95
Ilustración 4.38: Sección de eliminar cuenta.....	95
Ilustración 4.39: Confirmación eliminar usuario vista móvil.....	96
Ilustración 4.40: Confirmación eliminar usuario vista tableta	96
Ilustración 4.41: Confirmación eliminar usuario vista laptop	96
Ilustración 4.42: Cambio de contraseña vista móvil	97
Ilustración 4.43: Cambio de contraseña vista tableta.....	97
Ilustración 4.44: Cambio de contraseña vista laptop.....	98
Ilustración 4.45: Restablecer contraseña vista móvil	98
Ilustración 4.46: Restablecer contraseña vista tableta.....	99
Ilustración 4.47: Restablecer contraseña vista laptop.....	99
Ilustración 4.48: Email de recuperación.....	99
Ilustración 4.49: Formulario de cambio de contraseña vista móvil.....	100
Ilustración 4.50: Formulario de cambio de contraseña vista tableta	100
Ilustración 4.51: Formulario de cambio de contraseña vista laptop	100
Ilustración 4.52: Mensaje de error de enlace utilizado	101
Ilustración 4.53: Ingreso a la administración de Django.....	101
Ilustración 4.54: Mensaje de error de ingreso.....	102
Ilustración 4.55: Vista administrativa.....	102
Ilustración 4.56: Vista administrativa módulo Usuarios.....	103
Ilustración 4.57: Vista administrativa agregar administrador.....	104
Ilustración 4.58: Vista administrativa modificar administrador	104
Ilustración 4.59: Vista administrativa eliminar administrador	105
Ilustración 4.60: Vista administrativa módulo Directores de escuela	105
Ilustración 4.61: Vista administrativa agregar director de escuela	106
Ilustración 4.62: Vista administrativa modificar director de escuela	106
Ilustración 4.63: Vista administrativa eliminar director de escuela	107
Ilustración 4.64: Vista administrativa módulo carrera.....	107
Ilustración 4.65: Vista administrativa agregar carrera	107
Ilustración 4.66: Vista administrativa modificar carrera.....	108
Ilustración 4.67: Vista administrativa eliminar carrera.....	108
Ilustración 4.68: Vista administrativa módulo proceso	109
Ilustración 4.69: Vista administrativa agregar proceso.....	109
Ilustración 4.70: Vista administrativa modificar proceso	110

Ilustración 4.71: Vista administrativa eliminar proceso	110
Ilustración 4.72: Vista administrativa módulo solicitudes	111
Ilustración 4.73: Vista administrativa agregar solicitud	111
Ilustración 4.74: Vista administrativa modificar solicitud	112
Ilustración 4.75: Vista administrativa eliminar solicitud	112
Ilustración 4.76: Resultados pregunta 1	113
Ilustración 4.77: Resultados pregunta 2.....	113
Ilustración 4.78: Resultados pregunta 3.....	114
Ilustración 4.79: Resultados pregunta 4.....	114
Ilustración 4.80: Resultados pregunta 5.....	114

Tablas

Tabla 2.1: Modelo de tarjeta CRC.....	28
Tabla 3.1: Costos de licencias	31
Tabla 3.2: Modelo de tarjeta de historia de usuario	39
Tabla 3.3: Historia de usuario 1	39
Tabla 3.4: Historia de usuario 2	39
Tabla 3.5: Historia de usuario 3	40
Tabla 3.6: Historia de usuario 4	40
Tabla 3.7: Historia de usuario 5	40
Tabla 3.8: Historia de usuario 6	41
Tabla 3.9: Historia de usuario 7	41
Tabla 3.10: Historia de usuario 8	41
Tabla 3.11: Historia de usuario 9	42
Tabla 3.12: Historia de usuario 10	42
Tabla 3.13: Estimación de historias de usuario.....	43
Tabla 3.14: Tiempos planificados	44
Tabla 3.15: Esfuerzo de desarrollo	44
Tabla 3.16: Estimación de entregables.....	45
Tabla 3.17: Tarjeta CRC: User.....	49
Tabla 3.18: Tarjeta CRC: Estudiante	49
Tabla 3.19: Tarjeta CRC: Carrera	49
Tabla 3.20: Tarjeta CRC: Director.....	50
Tabla 3.21: Tarjeta CRC: Solicitud.....	50
Tabla 3.22: Tarjeta CRC: Proceso.....	50
Tabla 3.23: Prueba de aceptación N° 1	74
Tabla 3.24: Prueba de aceptación N° 2	74
Tabla 3.25: Prueba de aceptación N° 3	74
Tabla 3.26: Prueba de aceptación N° 4	74
Tabla 3.27: Prueba de aceptación N° 5	75
Tabla 3.28: Prueba de aceptación N° 6	75
Tabla 3.29: Prueba de aceptación N° 7	75
Tabla 3.30: Prueba de aceptación N° 8	76

Tabla 3.31: Prueba de aceptación N° 9	76
Tabla 3.32: Prueba de aceptación N° 10	76
Tabla 3.33: Prueba de aceptación N° 11	77
Tabla 3.34: Prueba de aceptación N° 12	77

INTRODUCCIÓN

En la presente disertación, previa a la obtención del título de Ingeniero en Sistemas y Computación titulada “APLICACIÓN WEB PARA LA GENERACIÓN DE SOLICITUDES ACADÉMICAS DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE AMBATO”, se implementa una tecnología muy utilizada en la actualidad como es el diseño responsivo. En el primer capítulo se muestra la problemática actual y justificación teórica, en el segundo y tercer capítulo se describe e implementa la metodología utilizada respectivamente y en el último capítulo se muestran los resultados obtenidos seguido de las conclusiones de la investigación.

CAPÍTULO I

FUNDAMENTOS TEÓRICOS

1.1 Antecedentes

La Pontificia Universidad Católica del Ecuador Sede Ambato (PUCESA), al ser una institución de educación superior forma profesionales en las distintas carreras que mantiene, toda institución actualmente está ligada a las TICs las cuales ayudan a los usuarios a optimizar su tiempo resolviendo problemas en su mayoría de casos cotidianos y simples donde con la tecnología que existe es fácil realizarla.

La tecnología ha ido evolucionando a lo largo de los años, aportando con desarrollos importantes, haciendo referencia a los computadores, celulares y *tablets* que la mayoría de personas posee, como innovación más reciente están los relojes donde distintas aplicaciones pueden funcionar en dicho dispositivo al igual que en un *Smartphone*.

El presente proyecto de grado, en vista de lo mencionado anteriormente se centra en la realización de una aplicación web multiplataforma donde todos los usuarios puedan acceder desde su *smartphone* o *tablet* y no solamente sea accesible desde una computadora, todo esto gracias al diseño responsivo que nos brinda la tecnología de desarrollo en la actualidad.

Los estudiantes están en constante realización de solicitudes académicas, y para las mismas actualmente no existe una ayuda que les permita un considerable ahorro de tiempo donde mediante un enlace se pueda ingresar a realizarlas desde cualquier dispositivo.

1.2 Problema

1.2.1 Descripción del problema

El problema que la aplicación pretende eliminar se centra en la redacción de las solicitudes académicas que todo estudiante en su vida universitaria debe realizar, esta aplicación ayudará sustancialmente al estudiante a poder generarlas desde cualquier lugar, en cualquier dispositivo electrónico con acceso a internet y a cualquier hora del día. Mediante observación directa el problema que se ha podido identificar es que las solicitudes que los estudiantes realizan no son aceptadas por las secretarías de escuela inmediatamente dado a que no existe un modelo específico en el cual se pueda tener de referencia para realizarlas, en el caso de nuevos estudiantes no conocen el nombre de su Director de Escuela al cual se dirigirá la solicitud. Este problema nace por la falta de información acerca de la ubicación y uso del quiosco de impresión para generar solicitudes que la universidad posee, aunque el mismo en ocasiones se encuentra apagado y subir al cuarto piso por una solicitud a la mayoría de personas no les gusta. Cabe mencionar que el mismo quiosco se encarga de imprimir la solicitud, no de guardarla en algún formato como .pdf o a su vez .docx, .odt, etc., quien quisiera generar la

solicitud, luego tendría que transcribirla para poder imprimir en las hojas membretadas de la institución.

Una aplicación web es un gran paso del *software* instalable que la mayoría de ocasiones funcionan únicamente para entornos de escritorio con sistema operativo Windows, a una aplicación web, que se la podría utilizar en cualquier dispositivo inclusive en un móvil o una tableta, así como explican (Caivano & Villoria, 2009) se espera que en un futuro dejen de existir aplicaciones de escritorio y sean sustituidas por aplicaciones web.

1.2.2 Preguntas básicas

- ¿Cuál es la manera en que las solicitudes académicas de la PUCESA son realizadas por los estudiantes?
- ¿Qué metodologías y herramientas son esenciales en el desarrollo del servicio de solicitudes académicas de la PUCESA?
- ¿Qué beneficio traería una aplicación web para la elaboración de solicitudes a los estudiantes de la PUCESA?

1.3 Justificación

La razón del desarrollo proyecto, es apoyar al estudiante en su vida universitaria para resolver los problemas mencionados anteriormente, se busca agilizar la creación de solicitudes académicas donde cada uno pueda acceder desde cualquier dispositivo con acceso a internet y simplemente

generarla, para así tener un ahorro significativo de tiempo al buscar un modelo o solicitud similar existente y adaptarla a su necesidad.

La aplicación que se va a realizar será multiplataforma utilizando un diseño responsivo el cual sea compatible con la mayoría de *gadgets* tecnológicos donde al momento de acceder sea agradable visualmente, en la actualidad al ser los dispositivos móviles mayormente utilizados, todas las páginas y aplicaciones web han tenido que utilizar este tipo de diseño, además en la Universidad existe un porcentaje bajo de investigaciones relacionadas.

Adicionalmente en el proyecto se aplicarán todos los conocimientos adquiridos para resolver problemas del mundo profesional en el desarrollo e investigación. La realización de la aplicación no será costosa en licenciamientos por el uso de *software* libre en su desarrollo.

1.4 Objetivos

1.4.1 General

Desarrollar una aplicación web para la generación de solicitudes académicas de la PUCESA.

1.4.2 Específicos

- Fundamentar de manera teórica y metodológica el desarrollo web.

- Diagnosticar la situación actual sobre la manera en la que se realizan las solicitudes académicas.
- Construir los elementos necesarios para integrarlos en el producto final.
- Realizar pruebas en cada avance del proyecto.

1.5 Fundamentos teóricos

1.5.1 Ingeniería de *software*

El *software* ha estado en constante crecimiento y desarrollo desde el uso de las computadoras, ha ido evolucionando hasta que construir *software* se convirtió en una industria, permite a los desarrolladores realizar programas de fácil mantenimiento y escalables. Esta disciplina de acuerdo a (Sommerville, 2005) abarca todo lo relacionado a la producción de software desde el comienzo del desarrollo hasta lo que será el mantenimiento del mismo luego de que ha sido entregado y utilizado por los usuarios finales, también explica que en la ingeniería de *software* no se incluye únicamente procesos técnicos sino distintas labores como la gestión de proyectos de *software* y la creación de teorías, métodos y herramientas de ayuda a la hora de producir software.

Es muy útil que la codificación cumpla normas las cuales ayuden al programador a identificar qué es lo que realiza cada parte de la misma y poder realizar los cambios que se requiera. La ingeniería de *software* propuesta por (Pressman, 2010) indica que el software está presente en cada momento de

nuestras vidas ya que el mismo es utilizado en la mayoría de actividades cotidianas, también abre la posibilidad de construir sistemas complejos con la más alta calidad posible en un tiempo aceptable.

1.5.2 Aplicaciones web

Como describe (Pressman, 2010) en sus inicios los sitios web consistían únicamente de varios archivos enlazados los cuales mostraban la información de texto e imágenes. El desarrollo ha evolucionado tanto en todo sentido y la creación de sitios web no fue la excepción, HTML evolucionó mediante el uso de herramientas de desarrollo como XML o Java que permitieron que los sitios web trabajen de manera conjunta con una base de datos y otras aplicaciones.

Como indican (Caivano & Villoria, 2009) las aplicaciones web tienen una gran ventaja que independientemente del sistema que el usuario final tenga instalado en su computadora, la aplicación debería funcionar igual. En lugar de que el desarrollador realice un programa para Mac OS, GNU/Linux, Windows u otro sistema operativo, este tipo de aplicaciones se las codifica una sola vez y se ejecuta de igual manera en cualquier entorno incluso móvil.

Como se ha mencionado, las aplicaciones web tienen una gran ventaja respecto a las aplicaciones que se las realiza para una plataforma en específico, en lo que se refiere a los móviles se puede lograr que una web se adapte a cualquier dispositivo asemejándolo a una aplicación nativa con la diferencia de que no es necesario que se descargada e instalada, basta utilizar

el navegador para acceder a ella. Otra de las ventajas son las actualizaciones así (Cuello & Vittone, 2013) indican que no es necesario descargar una nueva versión cada vez que se actualice el sistema, el usuario siempre estará utilizando la última versión.

1.5.3 Programación web

La programación de Aplicaciones Web ha ido evolucionando como se analizó anteriormente, hasta que existen muchos lenguajes de programación que se los puede utilizar, de acuerdo a lo que indica (Caivano & Villoria, 2009) los lenguajes que actualmente destacan son:

- PHP
- ASP/ASP.NET
- Java, mediante las tecnologías Java Servlets y JavaServer Pages (JSP)
- Perl
- Ruby
- Python
- HTML
- XML

Y aclara que ASP es una arquitectura de desarrollo web en la que se pueden utilizar distintos lenguajes como VB.NET o VBScript/JScript para ASP. En la programación de aplicaciones web es importante la correcta disposición de

cada elemento con el que el usuario interactúa, para lo cual se junta el desarrollo con el diseño para hacerlo agradable visualmente y funcionalmente solvente.

1.5.3.1 HTML5

HTML es un lenguaje donde su código se escribe dentro de etiquetas que posteriormente son interpretadas por un navegador. Para crear una página web no es necesario un programa especializado, basta con el uso de un editor de texto, actualmente la mayoría de desarrolladores no usan un editor de texto común en cambio han optado por desarrollar en lo que se denomina una API (Interfaz de Programación de Aplicaciones).

Como explica (Gauchat, 2012) HTML5 no es una nueva versión de HTML con el que se ha venido trabajando desde inicios de internet, HTML5 propone un nuevo concepto al desarrollo, en el que combina la computación en la nube, dispositivos móviles y trabajos en red. Como principal objetivo que HTML tenía era ubicar texto en internet organizando su contenido y compartiéndolo por este medio, esta fue la motivación de grandes empresas de crear nuevos lenguajes y nuevas características a la web las cuales nunca habían sido utilizadas.

(Gauchat, 2012) Además, indica que HTML5 es una combinación de HTML, CSS (Hoja de estilo en cascada) y JavaScript, de esta manera HTML5 provee a la web una alta funcionalidad proporcionada por JavaScript, estilo por CSS

y estructura brindada por HTML. Este tipo de tecnologías son dependientes como se indicará en temas posteriores, pero actúan en una sola unidad cuando de HTML5 se trata.

1.5.3.2 CSS

Los estilos predeterminados de HTML5 son CSS, como expone (Sawyer, 2012) CSS es una hoja de estilos en cascada la cual permite tener al desarrollador una página web con una disposición de elementos y un diseño agradable. Aunque parezca complicado dar estilo a toda una página web completamente, es todo lo contrario porque CSS permite dar un mismo estilo a diferentes etiquetas HTML.

El uso de CSS posee varios beneficios entre los que destacan:

- Dar cualquier estilo a una página web, brindando un mejor formato y disposición de elementos haciendo que una simple página web escrita únicamente en HTML, tenga un diseño de revista o periódico digital.
- En lo que son opciones de formato, CSS lo realiza en un espacio menor a las de HTML haciendo que las páginas web se carguen rápidamente.
- Se puede usar solo una hoja de estilos anexada a cada página del sitio, así al momento de actualizar un determinado estilo, este cambio se replicará en toda la página donde dicho estilo aparece.

Con la aparición de CSS3 los estilos han evolucionado más allá y como señala (Nixon, 2014) CSS3 añade más interacción del usuario con el sitio web donde esa interacción era posible únicamente gracias al uso de JavaScript.

1.5.3.3 JavaScript

JavaScript es ejecutado del lado del cliente y dentro del navegador enuncia (Nixon, 2014). Tiene sus inicios gracias a Java, y el mismo se lo puede escribir en cualquier computadora y poder ser ejecutado en otra, aunque esto estará condicionado por la versión y el tipo de navegador donde se lo quiera ejecutar.

Según (Sánchez Maza, 2012) JavaScript es el lenguaje de programación de la web, este lenguaje permite que una página realizada con HTML sea más interactiva al usuario, eliminando así la sencillez de la misma. Es un lenguaje basado en objetos, orientado a las aplicaciones cliente-servidor, donde JavaScript responde a cada evento que el usuario origine esto puede ser *clicks* del mouse, navegación entre documentos o datos de formularios.

La principal diferencia de Java con JavaScript es que el código de Java se lo compila al contrario de lo que pasa con JavaScript como se ha mencionado, es interpretado por el navegador esto quiere decir que es ejecutado directamente, adicionando que posee una más rápida ejecución.

1.5.3.4 Python

Python en la actualidad es un lenguaje muy popular por todas las características que posee, entre las más destacadas es que Python es un lenguaje muy poderoso y fácil de aprender y usar. Sobre la base de las consideraciones anteriores (Lutz, 2013) resalta que Python es portable, libre y además de enfocarse en la calidad del *software* también lo hace en la productividad del desarrollador en cualquier proyecto sea grande o pequeño.

Python no es como los otros lenguajes de programación web como PHP o Perl, este lenguaje no nació para eso, aunque ahora Python es uno de los lenguajes que mayormente se utiliza para la misma. (Bassi, 2013) Describe que Python se reorientó en etapa temprana y ahora la web es lo principal, aunque se lo utiliza también en otros desarrollos.

PHP sigue siendo un lenguaje de programación de aplicaciones web muy conocido y usado por lo que, a diferencia de Python o Ruby, tiene una mayor cantidad de desarrolladores y mayores servicios de soporte en lo que se refiere a *hosting*, aunque esto cada vez va cambiando por el constante crecimiento de estos dos últimos lenguajes que son la elección principal de actuales de programadores para el desarrollo de aplicaciones web recientes y modernas.

1.5.3.5 Diseño web responsivo

En lugar de verlo como una necesidad, el *responsive design* o diseño responsivo viene a ser una obligación actual de los sitios web así lo determina (Fantoni, 2015) por lo cual, al ser una nueva tendencia en el desarrollo, la mayoría de páginas web y administradores de contenido lo han ido adoptando para cumplir los requerimientos del mercado que indica que la mayoría de personas utiliza un dispositivo móvil para acceder a internet. El uso de celulares o tabletas está en auge y resulta necesario ofrecer una grata experiencia a todo tipo de usuario, a continuación, una ilustración del diseño responsivo:



Ilustración 1.1: Infografía ilustrativa de *Responsive Design*

Fuente: (Fantoni, 2015)

De acuerdo a (Aubry, 2014) aclara que el diseño web adaptable se lo confunde con el diseño responsivo, el primero quiere decir que se adapta a las características funcionales tanto de los navegadores como de los sistemas operativos, y el segundo agrupa un conjunto de técnicas de creación de sitios web perfeccionados para todo tipo de pantallas.

1.5.3.6 Bootstrap

Como indica su página web (Bootstrap Core Team, s.f.), Bootstrap es un *framework* de código abierto que agiliza la creación de sitios web con diseño responsivo dado que su filosofía es “*mobile first*”, móvil primero en español, aporta con el diseño de todos los elementos básicos que un sitio web puede tener para que, de esta forma, el programador ahorre tiempo en el diseño web del sitio.

1.5.3.7 PyCharm

En su página web (JetBrains, 2016) describe a PyCharm como un Entorno de Desarrollo Integrado (IDE) por sus siglas en inglés, orientado al desarrollo de software de lenguaje Python, aporta a los desarrolladores con varias características como la integración con un sistema de control de versiones y soporta el desarrollo web con Django.

1.5.4 *Frameworks* de desarrollo web

Tal como se ha visto hoy en día la tecnología facilita la vida de las personas, en el desarrollo de aplicaciones las herramientas que hacen posible este concepto han venido mejorando, simplificando el trabajo por el uso de *frameworks* que permiten a los programadores evitar trabajos repetitivos. Hechas las consideraciones anteriores cabe agregar que un *framework* de desarrollo web es considerado una metodología o forma de trabajo añadida a

una librería de código para facilitar la vida al desarrollador citando a (Ordax Cassá & Ocaña Díaz Ufano, 2012).

En efecto (De Luca, 2011) describe que un *framework* ayuda a enfrentar necesidades particulares durante el desarrollo a través de código con el que cuenta. Es el nuevo tipo de desarrollo y por la razón que se usa Django durante este proyecto (Infante Montero, 2012) propone que es parte de la tercera generación de desarrollo web.

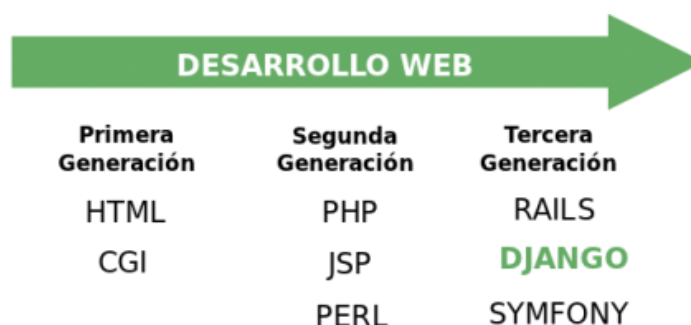


Ilustración 1.2: Generaciones del desarrollo web

Fuente: (Infante Montero, 2012)

1.5.4.1 Django

Como señala (Django Software Foundation, s.f.) en su página, Django es un *framework* de desarrollo web creado con Python el cual impulsa el limpio y rápido desarrollo de aplicaciones, y posee varias ventajas que se señalan a continuación:

- Diseñado para un desarrollo rápido desde su concepto hasta su aplicación.

- Posee varios extras que pueden ser utilizados y no tener que diseñar algo desde cero.
- Ayuda a los programadores a evitar errores comunes de seguridad durante el desarrollo.
- Permite satisfacer demandas de tráfico.
- Es versátil y se lo ha utilizado en distintos tipos de desarrollo.

1.5.4.2 Modelo-Plantilla-Vista

Como explica (Honrubia López, Álvarez Garrote, Pérez Marín, & Bretón, 2013) el patrón de diseño sobre el que se ha trabajado durante muchos años es el Modelo-Vista-Controlador (MVC) y ha sido de mucha ayuda en los desarrollos al clasificar los objetos en roles a medida que se realiza cualquier proyecto de *software*. Pero Django interpreta de otra manera cada rol de desarrollo de acuerdo a como definen en su página web oficial, Django es un *framework* Modelo-Plantilla-Vista (MPV); en inglés *Model-Template-View* (MTV).

En lo que se refiere a modelo no cambia y son los datos que maneja la aplicación, como la base de datos que interactúa con el sitio, la plantilla viene a ser la capa de presentación donde es importante adoptar las mejores prácticas y mostrar una buena interfaz, finalmente la vista describe el dato que va a ser mostrado en una plantilla apropiada de acuerdo a (Holovaty & Kaplan-Moss, 2007).

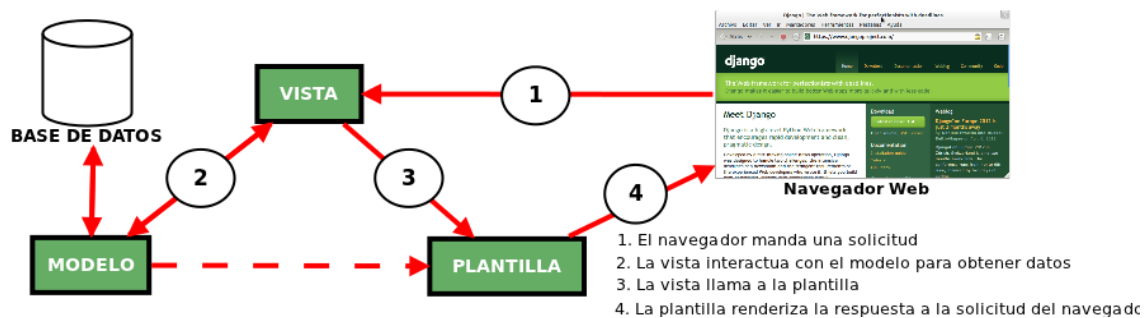


Ilustración 1.3: Funcionamiento del MTV de Django

Fuente: (Infante Montero, 2012)

1.5.5 Base de Datos

Las bases de datos son muy utilizadas por la facilidad que tienen de acceder a los datos y a la información, para (Coronel, 2011) quien coincide con (Villar Varela & Fernández Pérez, 2010) una base de datos es un conjunto de datos que tienen que ver con un tema en especial y se los clasifica de acuerdo a un criterio para su posterior uso de la información que almacena, para manipular la estructura de lo que representa una base de datos y controlar el acceso a los mismos, es necesario un sistema de administración de bases de datos (DBSM), adicionalmente se describen varias ventajas de su uso:

- Posibilita la integridad y consistencia de los datos, esto se refiere a la validez de los datos y obviar redundancias.
- Ha reducido el espacio de almacenamiento al disminuir la redundancia de los datos, así hay un uso menor de memoria secundaria.
- Los sistemas administradores de bases de datos permiten un acceso concurrente a los mismos para mantenerlos íntegros o que la información no se pierda luego de accesos simultáneos.

- El acceso a los datos es simple, los DBSM permiten a los usuarios con pocos conocimientos la realización de consultas.
- La actualización de cualquier dato almacenado no afecta al programa en sí, la modificación afecta a la base de datos y no al programa que accede a los mismos.

1.5.5.1 PostgreSQL

PostgreSQL es un Sistema de Administración de bases de datos de código abierto, de acuerdo a lo que indica (Maymala, 2015) PostgreSQL es un sistema gestor de datos tan poderoso que hace que las aplicaciones que lo utilicen sean confiables y escalables sin incrementar costos al equipo de desarrollo.

Es muy importante explorar distintas soluciones a proyectos de desarrollo que puedan contener características que otros complementos que son importantes para el *software* no lo tengan, es el caso de PostgreSQL que como expone (Obe & Hsu, 2012), este motor posee características únicas que otros motores no poseen. A diferencia de MySQL se enfoca a la integridad y fiabilidad de los datos y un funcionamiento más eficaz.

1.5.5.2 Mapeo Objeto-Relacional

La interacción de Django con la base de datos es a través del mapeo Objeto-Relacional (ORM, por sus siglas en inglés) esta es una de las más

significativas características que posee según (Pinkham, 2015) donde en lugar de crear una base de datos previa, se crean modelos en forma de objetos dentro del código de la aplicación. En efecto este proceso de comunicación con los datos permite a los desarrolladores un ahorro considerable de tiempo durante la realización de un proyecto por lo que las migraciones, modelos y *managers* son muy importantes.

(Forcier, Bissex, & Chun, 2008) indican que Django al igual que otros *frameworks* de desarrollo modernos, se basa en una capa de acceso de datos rica que intenta conectar una base de datos relacional con la naturaleza de Python orientada a objetos.

1.5.6 Solicitudes académicas

De acuerdo a (Real Academia Española, 2016) solicitud es un documento o carta con el cual se realiza un pedido de algo de manera oficial. Las solicitudes pueden ser de empleo, pasaportes, pedidos dentro de lo laboral y para muchos otros fines, pero en lo que este proyecto se basa es en solicitudes académicas que realiza un estudiante dentro de una institución educativa en este caso la PUCESA.

1.5.6.1 Formato PDF (*Portable Document Format*)

Según (Oliver González, Moré López, & Climent Roca, 2011) El formato PDF es propiedad de Adobe, está hecho para que sea posible visualizarlo en

cualquier computadora sin importar el sistema operativo que esté utilizando. Se caracteriza por ser comprimido permitiendo la transmisión eficiente de información que incluya gráficos e imágenes.

Como se ha podido observar, actualmente todos los dispositivos móviles como los *smarthphones* o *tablets* son capaces de leer archivos con este formato, pero el usuario está libre de instalar otro tipo de aplicaciones a su conveniencia para poder leerlos.

1.5.6.2 Procesos

A un proceso la (Real Academia Española, 2016) lo describe como un conjunto de fases en sucesión de una operación artificial o de un fenómeno natural. En el presente proyecto en relación con las solicitudes académicas un proceso se lo describe como los pasos para que dichas solicitudes sean presentadas.

1.5.7 Pontificia Universidad Católica del Ecuador Sede Ambato

En la página web institucional (PUCESA, 2011) explica que la PUCESA forma parte del Sistema Nacional de la PUCE, es una institución académica y de servicio inspirada en los principios cristianos la cual promueve la formación de profesionales con valor humanístico y cultural además el bienestar de sus integrantes con el fin de satisfacer las necesidades de la comunidad.

CAPÍTULO II

METODOLOGÍA

2.1 Metodología de la investigación

La investigación científica posee varias etapas mediante las cuales el investigador busca deducir, enmendar, validar y aplicar el conocimiento adquirido, a través de la utilización del método científico para así recolectar información de importancia veraz que sirva de aporte para resolver el problema planteado así lo indica (Rodríguez, 2005).

Para la recolección de información significativa para la presente investigación, se utilizó la metodología analítica.

2.1.1 Metodología analítica

Como indica (Hurtado & Toro, 2007) la metodología analítica es separar por partes de manera mental el objeto de estudio para así entender y alcanzar nuevos conocimientos sobre dicho objeto. Estos procesos inducen a las personas a pensar y buscar soluciones factibles a problemas planteados, también cada proceso resultante de la descomposición demuestra que es necesario e íntimamente relacionado a los demás porque el método analítico termina con la unión de la desintegración realizada.

El uso de esta metodología acompañada de apoyo perteneciente a fuentes bibliográficas, permite recolectar datos útiles para el desarrollo del proyecto investigativo.

2.1.2 Técnicas e instrumentos de recolección de datos

La recolección de información es importante para saber la forma que se debe actuar de cara a la búsqueda de una solución, (Rodríguez, 2005) describe que todo tipo de técnicas de recolección de datos es la manera más frecuente de adquirir información acerca del mundo que nos rodea, en el proyecto presente se utilizan dos tipos de instrumentos investigativos:

- Entrevista
- *Focus group*

2.1.2.1 Entrevista

El autor (Hurtado & Toro, 2007) manifiesta que la entrevista junto con el cuestionario son técnicas habituales en la recolección de información. La entrevista consiste en una correspondencia directa entre el investigador y la realización de su estudio a través de personas o grupo de personas con la intención de documentar evidencias de forma oral.

2.1.2.1.1 Realización de entrevistas

La entrevista (ver Anexo 1) para determinar los requisitos de *software* se la realizó a la secretaria del departamento de la Escuela de Ingeniería en Sistemas y la secretaria de la Escuela de Psicología de la PUCESA.

2.1.2.2 Focus group

Conocido también como entrevista enfocada o en grupo está orientada a grupos pequeños cuyos participantes guardan ciertos criterios como actitudinales, de consumo o sociodemográficos, se reúnen por un tiempo expreso para charlar sobre un tema fijo y así extraer la información necesaria al moderador así lo declara (Llopis, 2004).

2.1.2.2.1 Focus group realizado

El siguiente *focus group* (Ver anexo 2) se realizó a un grupo de estudiantes pertenecientes a varias carreras de distintos semestres de estudio con el objetivo principal de conocer qué es lo que el usuario desea y es de importancia para así hacer que el proyecto cumpla todo lo demandado por parte del grupo que representan a los consumidores principales de la aplicación.

2.2 Metodología de desarrollo de *software*

En el sitio web de la Universidad de Murcia (Barzana, 2006) sostiene que una metodología sirve de ayuda al desarrollador ofreciendo técnicas y procedimientos para una correcta documentación y la elaboración del producto final, se lo relaciona con una guía de pasos a seguir, que personas deben participar y sus roles dentro del proyecto, además de la información que se debe juntar como consecuencia de la producción del producto y la información que se necesita para empezar a realizarlo.

2.2.1 Metodología ágil

Como se ha observado en la actualidad los desarrollos de nuevos sistemas computacionales vienen trabajando bajo este tipo de metodologías y varios eventos, workshops o conferencias realizados en los últimos años, metodologías de desarrollo ágil siempre están presentes, así lo destaca de igual forma (Canós, Letelier, & Penadés, 2003), de igual forma señala que una de las características más importantes es su simplicidad, pero sin perder la calidad del producto. Entre las metodologías ágiles que menciona están:

- Programación Extrema (*Extreme Programming, XP*)
- SCRUM
- *Crystal Methodologies*
- *Dynamic System Development Method (DSDM)*
- *Adaptative Software Development (ASD)*

- *Feature-Driven Development (FDD)*
- *Lean Development (LD)*

2.2.2 Programación Extrema (*Extreme Programming, XP*)

Esta metodología es la más utilizada actualmente junto a la metodología Scrum, ha llegado a ser exitosa porque está orientada a equipos pequeños, con un plazo de tiempo corto al igual que el plazo de tiempo para realizarlo, como su nombre lo indica se trata de una programación rápida con interacción constante con el usuario final así lo enfatiza (Mendoza, 2004) que a su vez da a conocer varias características de esta metodología:

- Pruebas unitarias orientadas a las actividades más importantes que debe realizar el programa.
- Re-fabricación, esta característica tiene que ver con desarrollos futuros para reutilizar el código implementado en proyectos ya realizados.
- Programación en pareja donde los desarrolladores programan junto a un compañero en un mismo lugar de trabajo.
- Simplicidad al programar los módulos del sistema, comunicación constante con el usuario final y *feedback* entre las fases y los participantes del proyecto.

El proceso que se debe seguir de esta metodología establecidas en las fases descritas por (Pressman, 2010) se detalla a continuación en la Ilustración 2.1:



Ilustración 2.1: Proceso de la Programación Extrema

Fuente: Adaptado de (Pressman, 2010)

2.2.2.1 Fase 1: Planeación

2.2.2.1.1 Historias de usuario

Las historias del negocio como lo propone (Pressman, 2010) durante la conversación con el cliente, se van generando varias historias que van a detallar algunos de los requerimientos del programa, estas tarjetas se las califica de acuerdo a la prioridad que tienen conforme a al negocio, para así pasar a los programadores quienes se encargan de estimar las semanas de desarrollo que tomará realizarlo y el costo, si sobrepasa las tres semanas, se pedirá al cliente descomponer en partes más pequeñas la futura entrega. A medida que avanza el proyecto se pueden ir agregando nuevas historias.

2.2.2.1.2 Plan de iteración

Los clientes y desarrolladores deben trabajar juntos para organizar así las historias de usuario y planificar fechas tentativas de entrega de cada historia. (Pressman, 2010) agrega que el tiempo que se demore en entregar cada historia de usuario será el tiempo que va a durar el proyecto.

(Beck, 2000) expresa que el plan de iteración, se utilizan de igual forma las tarjetas de usuario realizadas, para evaluar los entregables dados al usuario en el tiempo establecido. Aquí el cliente tiene mucha participación en la que asume el direccionamiento del proyecto para realizar cambios al proyecto.

2.2.2.2 Fase 2: Diseño

2.2.2.2.1 Diseño simple

Ludwig Mies van der Rohe, un arquitecto que además de ser conocido por sus obras es conocido por una frase: “menos es más” que ha sido aplicada en distintas actividades que el ser humano realiza y siempre es mencionada implantando ese pensamiento como sinónimo de simpleza y buen gusto. Esta frase aplica también para el diseño de *software* y más aún como principio de la metodología.

Desde el punto de vista de (Beck, 2000) los diseños simples son los mejores así propone cuatro condiciones para un diseño simple:

- El sistema en general debe comunicar todo lo que se desee
- El código no debe estar duplicado
- El sistema debe tener el menor número de clases posible
- El sistema debe tener el menor número de métodos posible

2.2.2.2.2 Tarjetas CRC

(Pressman, 2010) plantea que es lo único que se compone en esta fase, consiste en tarjetas con el detalle de clase, responsabilidad y colaborador (CRC), Es una forma de componer el *software* en un modelo orientado a objetos. En la parte de clase va el nombre de las clases se utilizan para el desarrollo, en responsabilidad se describe lo que la clase realiza y por último el colaborador que muestra las clases necesarias que brinden información a la clase principal y cumplir su fin. A continuación, se presenta un modelo de tarjeta CRC:

Clase	
Responsabilidad	Colaborador

Tabla 2.1: Modelo de tarjeta CRC

Fuente: (Pressman, 2010)

2.2.2.2.3 Prototipos

En lo que se refiere a creación de prototipos (Pressman, 2010) aconseja que se deben realizar cuando en el programador se encuentra con un problema

difícil, debe elaborar el prototipo de esa sección para probarlo y así evitar problemas a futuro durante la implementación.

2.2.2.3 Fase 3: Codificación

2.2.2.3.1 Programación

(Beck, 2000) declara que la mejor forma de aprender son los test o pruebas, luego de establecer las pruebas de unidad (Pressman, 2010) plantea que se debe empezar a programar en base a pasar dicha prueba, no se agrega nada diferente a lo que se haya planificado, se guarda la simpleza y cuando se termine la programación ahí se realizará la prueba de unidad formulada. Además, hace énfasis en la programación en parejas e integración continua destacando sus beneficios para el desarrollo como prevenir problemas de compatibilidad con otros módulos del programa realizados por otras parejas del equipo, tener a los desarrolladores centrados en lo que deben hacer y asegurar la calidad de cada avance que se realice.

2.2.2.3.2 Rediseño

(Pressman, 2010) alude al rediseño como el cambio que se realiza a la parte interna del programa (código) o su diseño, no se perturbe el comportamiento exterior o funcionamiento.

2.2.2.4 Fase 4: Prueba

2.2.2.4.1 Prueba unitaria

Como se mencionó en la fase de programación, las pruebas unitarias deben realizarse antes de la codificación en sí, de acuerdo a (Sommerville, 2005) estas pruebas se realizan a cada componente de forma individual para determinar que funcionen como se requiere, se las realiza constantemente y deben almacenarse en caso de que se realice cambios a la estructura de la parte sometida a evaluación.

2.2.2.4.2 Prueba de aceptación

(Sommerville, 2005) considera que estas pruebas corresponden a la etapa final de la realización de las pruebas, por lo cual se requiere que el cliente lo realice para que se manipule el sistema con los datos que vayan a utilizar en el mundo real diferentes a los datos utilizados de prueba y verificar que no se hayan pasado cosas por alto, como validaciones o requerimientos.

(Pressman, 2010) declara que estas pruebas provienen de las historias de usuario implementadas, que forman parte de cada liberación del programa.

CAPÍTULO III

RESULTADOS

3.1 *Software* y licencias

Producto	Licencia	Duración
Python y Django	Libre	-
JetBrains PyCharm	\$0	1 año
Bootstrap	Libre	-
Git	Libre	-

Tabla 3.1: Costos de licencias

Fuente: elaboración propia

Como se muestra en la Tabla 3.1. las licencias utilizadas para el desarrollo del presente proyecto son del tipo libre a excepción de PyCharm, el IDE de desarrollo posee un costo por su licencia, pero de acuerdo a su página web (JetBrains, 2016) sus productos, al cual pertenece el programa utilizado, son gratuitos por un año para estudiantes quienes posean un correo institucional, por lo que fue activado gracias al correo de la PUCESA.

3.2 Fase I: Planeación

Realización de entrevistas y *focus group* a futuros usuarios de la aplicación con el objetivo de definir los requerimientos funcionales de la aplicación y entregables a realizar.

3.2.1 Entrevista

Durante el desarrollo del presente proyecto, se implantó la solución por parte del área de informática de la institución, lamentablemente la aplicación de acuerdo a las entrevistas realizadas a las Señoras Secretarías de la Escuela de Sistemas y Psicología, y a varios estudiantes mediante el *focus group*, se obtuvieron varias particularidades a considerar.

1. ¿Conoce algún mecanismo en la universidad que permita realizar solicitudes académicas?

Sí existe un sistema que se encuentra alojado en la aplicación Academics, que el estudiante tiene que acceder al sistema para realizar cualquier solicitud que requiera, y es respondido con un email para que se acerque a ver su solicitud que se encuentra realizada.

2. ¿Qué se puede decir que le falta a dicho sistema?

Se indicó que el sistema le hace falta acoplarse a las necesidades del estudiante principalmente porque están las solicitudes más básicas, que se necesitan más formatos para realizar y que la aplicación debe ser más amigable. Asimismo, se manifestó que al momento de imprimir las solicitudes realizadas en dicho sistema no se imprime bien y aparece de distinta forma a la que se muestra en la pantalla.

3. ¿Es alto el número de solicitudes que los estudiantes entregan normalmente durante el semestre?

En la Escuela de Sistemas al ser pocos estudiantes se manifestó que el número de solicitudes que se realizan no es alto, pero cuando la entrevistada por parte de la misma escuela mencionó que cumplía su misma labor en maestrías, indicó que en dicha secretaría todo se maneja en base a solicitudes y se coincidió que en escuelas con mayor número de alumnos es alta la cantidad de solicitudes que se realizan.

4. ¿Los estudiantes se acercan con frecuencia a solicitar algún modelo de solicitud?

Si se acercan de acuerdo a las necesidades, principalmente a inicios del semestre o al finalizar el semestre, también se supo manifestar que en el

proceso de grado se debe realizar varias solicitudes y los estudiantes se acercan a solicitar los modelos.

5. ¿Cree que una aplicación web agilizaría el proceso de realizar solicitudes a los estudiantes?

Se declaró que una aplicación web si sería factible realizarla enfocándose a las necesidades del estudiante.

6. ¿Qué funcionalidades cree usted que debería tener la aplicación para satisfacer las necesidades del estudiante?

Se supo manifestar que el estudiante debería poder ingresar con su número de cédula o número de matrícula. También las solicitudes deben estar de acuerdo a las exigencias institucionales detallando correctamente varios campos de la solicitud y esté redactada de forma correcta.

7. ¿Qué solicitudes son las que más se realizan?

Entre las solicitudes que más se realizan por parte del estudiante se mencionaron las siguientes:

- Matrícula
- Asistencia a clases
- Culminación del semestre

- Proceso de grado
- Certificado de no ser sancionado
- No haber reprobado materias con tercera matrícula
- Contenidos programáticos
- Copia certificada de historiales académicos

3.2.2 Focus group

1. ¿Ha realizado alguna vez en su etapa universitaria solicitudes académicas?

Todos los estudiantes respondieron afirmativamente lo que motiva más al uso de la aplicación.

2. ¿Cuál ha sido el proceso que ha seguido para realizarlas?

Los estudiantes detallaron los siguientes pasos para realizar las solicitudes:

1. Ir al centro de cómputo o las copadoras que se encuentran en la institución.
2. Ver si existe el modelo que necesitan en el mismo lugar.
3. Dirigirse a secretaría a verificar si está la solicitud bien realizada.
4. Imprimir si es el caso en hoja membretada de la institución.
5. Regresar a secretaría a dejarla.

3. ¿Conoce sobre los modelos de solicitudes que ofrece la secretaría de la escuela?

En el caso de la Escuela de Sistemas ofrece un folleto en el que los estudiantes pueden solicitarlo y ver algún modelo que necesiten, pero ninguno de ellos lo conocía, de igual forma sucedió con los estudiantes de las otras carreras.

4. ¿Conoce algún mecanismo en la universidad que permita realizar con facilidad dichas solicitudes? ¿Cuéntanos la experiencia?

Si conocen que existe dicho sistema, pero nunca lo utilizan, para realizar sus solicitudes lo hacen de acuerdo a los pasos descritos anteriormente.

5. ¿Qué tan familiarizados están en utilizar en EVAE o Académicas desde el móvil?

Los estudiantes realizan procesos académicos con frecuencia desde su dispositivo móvil.

6. ¿Cree que una aplicación web facilitaría la creación de solicitudes y ahorraría tiempo a los estudiantes?

Se mencionó que es muy importante automatizar este proceso y sería de gran ayuda para los estudiantes al facilitarse el formato correcto sin perder tiempo

buscándolo. También se recalcó que el estudiante ahorraría dinero porque habido casos en los que les ha tocado repetir la solicitud volviendo a cancelar el valor de una hoja membretada.

7. ¿Qué diseño les gusta más al abrirlo desde el móvil? (EVAE - Academics)

Los estudiantes de la escuela de sistemas recalcaron que el entorno EVAE se adapta a los dispositivos, por lo que su gusto se inclina en esta dirección, también mencionaron que es molesto tener que ingresar dos veces el usuario y contraseña en la aplicación Academics para poder utilizarla además señalaron que el mismo no se adapta a todos los dispositivos. Cuando se realizó la misma pregunta a los alumnos de las otras carreras de igual forma destacaron que es mejor utilizar el EVAE en el móvil en comparación con la otra aplicación.

8. ¿Cree que la aplicación web debe visualizarse correctamente en teléfonos móviles?

Se indicó que se debe tener en cuenta todos los dispositivos en donde se puede utilizar la aplicación.

9. ¿Qué funcionalidades debería tener la aplicación para satisfacer las necesidades de los estudiantes?

Aparte de la funcionalidad principal que es generar las solicitudes se indicaron más requerimientos en cuestión de diseño como la forma ordenada que deben presentarse las solicitudes y el acceso mediante una cuenta donde se almacenen sus datos.

10. ¿Cree que la aplicación deba mostrar el proceso que sigue su solicitud al ser entregada?

Los estudiantes coincidieron que sería importante ser informados sobre los procesos que una solicitud sigue para ser aceptada luego de dejarla.

3.2.3 Historias de usuario

Las historias de usuario se las realizó mediante reuniones entre el desarrollador y varios estudiantes para determinar los requerimientos funcionales, en la Tabla 3.2. se muestra una tarjeta de historia de usuario y los detalles que contiene.

Historia de Usuario			
Número:	Identificador único	Usuario:	Persona o grupo a evaluar
Nombre:	Identificador textual de la acción a realizar		
Iteración Asignada:	Disposición para entregable	Puntos Estimados:	Duración
Prioridad en negocio:	Prioridad	Riesgo de desarrollo:	Cumplimiento de requerimientos
Descripción: Todo lo que va a realizarse.			
Observaciones: Énfasis en criterios de importancia para la aplicación.			

Tabla 3.2: Modelo de tarjeta de historia de usuario

Fuente: elaboración propia

Las historias de usuario acerca de las prioridades a cumplir de la aplicación se muestran posteriormente:

Historia de Usuario			
Número:	1	Usuario:	Todos
Nombre:	Base de datos		
Iteración Asignada:	1	Puntos Estimados:	1
Prioridad en negocio:	Alta	Riesgo de desarrollo:	Alto
Descripción: Se requiere que la aplicación gestione los datos mediante una base de datos			
Observaciones: La base de datos es independiente de la base del departamento de Tecnologías de la Información de la universidad.			

Tabla 3.3: Historia de usuario 1

Fuente: elaboración propia

Historia de Usuario			
Número:	2	Usuario:	Estudiantes
Nombre:	Formulario de ingreso/registro		
Iteración Asignada:	1	Puntos Estimados:	1
Prioridad en negocio:	Alta	Riesgo de desarrollo:	Bajo
Descripción: Todos los usuarios requieren acceder a la aplicación por medio de un usuario y contraseña de igual forma es importante que se puedan registrar.			
Observaciones:			

Tabla 3.4: Historia de usuario 2

Fuente: elaboración propia

Historia de Usuario			
Número:	3	Usuario:	Visitantes
Nombre:	Utilización del servicio		
Iteración Asignada:	1	Puntos Estimados:	1
Prioridad en negocio:	Alta	Riesgo de desarrollo:	Medio
Descripción: Existencia de ex alumnos de la universidad y no poseen cuenta para acceder, se desea crear una sección donde ellos puedan realizar solicitudes.			
Observaciones: Mostrar solo las solicitudes que pueden realizar los visitantes.			

Tabla 3.5: Historia de usuario 3

Fuente: elaboración propia

Historia de Usuario			
Número:	4	Usuario:	Administrador
Nombre:	Ingresar/modificar solicitudes		
Iteración Asignada:	2	Puntos Estimados:	0,5
Prioridad en negocio:	Alta	Riesgo de desarrollo:	Medio
Descripción: Se requiere que exista un usuario administrador que pueda realizar modificaciones sobre las solicitudes almacenadas.			
Observaciones:			

Tabla 3.6: Historia de usuario 4

Fuente: elaboración propia

Historia de Usuario			
Número:	5	Usuario:	Administrador
Nombre:	Modificar datos de carrera/escuela		
Iteración Asignada:	2	Puntos Estimados:	0,5
Prioridad en negocio:	Alta	Riesgo de desarrollo:	Bajo
Descripción: Se requiere que el usuario administrador pueda modificar datos de nombres de carrera, escuela y directores si es necesario.			
Observaciones:			

Tabla 3.7: Historia de usuario 5

Fuente: elaboración propia

Historia de Usuario			
Número:	6	Usuario:	Todos
Nombre:	Generar solicitud		
Iteración Asignada:	3	Puntos Estimados:	1,5
Prioridad en negocio:	Alta	Riesgo de desarrollo:	Medio
Descripción: Se desea que el usuario pueda seleccionar el tipo de solicitud que quiera y obtenga un documento para poder imprimirlo.			
Observaciones: El documento será de tipo PDF, tener en cuenta datos adicionales que completar en determinadas solicitudes.			

Tabla 3.8: Historia de usuario 6

Fuente: elaboración propia

Historia de Usuario			
Número:	7	Usuario:	Todos
Nombre:	Visualizar Solicitud		
Iteración Asignada:	3	Puntos Estimados:	1
Prioridad en negocio:	Media	Riesgo de desarrollo:	Medio
Descripción: El usuario debe ser capaz de poder visualizar su solicitud dentro de la aplicación antes de descargarla.			
Observaciones:			

Tabla 3.9: Historia de usuario 7

Fuente: elaboración propia

Historia de Usuario			
Número:	8	Usuario:	Estudiantes
Nombre:	Modificar contraseña		
Iteración Asignada:	4	Puntos Estimados:	0,2
Prioridad en negocio:	Media	Riesgo de desarrollo:	Bajo
Descripción: Todos los usuarios requieren cambiar la contraseña que poseen para una mayor seguridad.			
Observaciones: El usuario puede recuperar si olvidó su contraseña mediante un enlace enviado por correo.			

Tabla 3.10: Historia de usuario 8

Fuente: elaboración propia

Historia de Usuario			
Número:	9	Usuario:	Estudiantes
Nombre:	Modificar/eliminar usuario		
Iteración Asignada:	4	Puntos Estimados:	1
Prioridad en negocio:	Alta	Riesgo de desarrollo:	Medio
Descripción: Los usuarios son capaces de modificar su perfil donde incluye: nombre, apellido, matrícula, carrera a la que pertenece.			
Observaciones: Los usuarios adicionalmente son capaces de eliminar sus cuentas.			

Tabla 3.11: Historia de usuario 9

Fuente: elaboración propia

Historia de Usuario			
Número:	10	Usuario:	Estudiantes
Nombre:	Cierre de sesión		
Iteración Asignada:	4	Puntos Estimados:	0,2
Prioridad en negocio:	Alta	Riesgo de desarrollo:	Bajo
Descripción: Los usuarios registrados requieren que se pueda terminar su sesión luego de usar la aplicación.			
Observaciones:			

Tabla 3.12: Historia de usuario 10

Fuente: elaboración propia

3.2.3.1 Planificación de desarrollo de Historias de Usuario

De acuerdo a las reuniones realizadas para la planificación, se ejecutó la valoración de tiempo de las historias de usuario y lo podemos ver en la siguiente tabla:

Módulo	N° H.U.	Historia de usuario	Prioridad	Riesgo	Duración
Base de datos	1	Base de datos	Alta	Alto	5 días
Uso de aplicación	2	Formulario de ingreso/registro	Alta	Bajo	5 días
	3	Utilización del servicio	Alta	Medio	5 días
Administrador	4	Ingresar/Modificar solicitudes	Alta	Medio	3 días
	5	Modificar datos de carrera/escuela	Alta	Bajo	3 días
Solicitudes	6	Generar solicitud	Alta	Medio	8 días
	7	Visualizar solicitud	Media	Medio	5 días
Cuenta estudiante	8	Modificar contraseña	Media	Bajo	1 días
	9	Modificar/eliminar usuario	Alta	Medio	5 días
	10	Cierre de sesión	Alta	Bajo	1 días

Tabla 3.13: Estimación de historias de usuario

Fuente: elaboración propia

3.2.3.2 Planificación de desarrollo de Historias de Usuario

Las historias de usuario se agruparon en entregables los cuales fueron liberados en cada iteración, se determinó, el esfuerzo que demandó cumplir con lo solicitado, la duración, y los tiempos calendario empleados.

3.2.3.3 Tiempo calendario planificado

Horas de desarrollo diario	Días por semana	Semanas del mes
6 horas	5 días	4 semanas

Tabla 3.14: Tiempos planificados

Fuente: elaboración propia

3.2.3.4 Esfuerzo de desarrollo en base a un programador

Por la razón que el proyecto es realizado de forma personal se establecieron los siguientes tiempos:

Número de desarrolladores	Horas de esfuerzo estimadas	Días dedicados al desarrollo	Semanas
1 persona	6 horas	5 días	1 semana

Tabla 3.15: Esfuerzo de desarrollo

Fuente: elaboración propia

3.2.3.5 Estimación de entregables

Módulo	N° H.U.	Historia de usuario	Prioridad	Riesgo	Duración	Esfuerzo de desarrollo			Iteración					
						Horas	Días	Semanas	1	2	3	4	5	
Base de datos	1	Base de datos	Alta	Alto	5 días	30	5	1	x					
Uso de aplicación	2	Formulario de ingreso/registro	Alta	Bajo	5 días	30	5	1	x					
	3	Utilización del servicio	Alta	Medio	5 días	30	5	1	x					
Administrador	4	Ingresar/Modificar solicitudes	Alta	Medio	3 días	18	3	0,5		x				
	5	Modificar datos de carrera/escuela	Alta	Bajo	3 días	18	3	0,5		x				
Solicitudes	6	Generar solicitud	Alta	Medio	8 días	48	8	1,5			x			
	7	Visualizar solicitud	Media	Medio	5 días	30	5	1			x			
Cuenta estudiante	8	Modificar contraseña	Media	Bajo	1 días	3	1	0,2					x	
	9	Modificar/eliminar usuario	Alta	Medio	5 días	30	5	1					x	
	10	Cierre de sesión	Alta	Bajo	1 días	3	1	0,2					x	

Tabla 3.16: Estimación de entregables

Fuente: elaboración propia

3.3 Fase II: Diseño

La metodología fomenta una participación constante por parte del usuario, por lo cual se han definido metáforas la aplicación posee para que la interacción entre el desarrollador y los clientes sea entendible y de forma clara. Lo que las metáforas realizan es detallar lo que concibe cada módulo de la aplicación.

3.3.1 Módulos de la aplicación

Los módulos propuestos para la elaboración de las metáforas son los siguientes:

- Uso de la aplicación
- Cuenta administrador
- Solicitudes
- Cuenta estudiante

3.3.1.1 Metáfora de uso de la aplicación

En el presente módulo se encarga del acceso y uso de la aplicación, aquí el estudiante puede ingresar su cédula como usuario y contraseña que la puede cambiar si desea. También existen usuarios son ex-alumnos de la universidad para lo cual existe una sección dentro de la aplicación para que ellos también las realicen.

3.3.1.2 Metáfora de administrador

En este módulo de cuenta de administración la aplicación realiza todos los cambios requeridos por el usuario administrador para la actualización de datos importantes que se requieran dentro de las solicitudes académicas, los cambios pueden ser de directores de escuela, nombres de carreras, agregar carreras, actualización de palabras dentro de solicitudes, o agregar nuevas solicitudes al sitio como también eliminar datos mencionados.

3.3.1.3 Metáfora de solicitudes

En las solicitudes, el estudiante quien posee acceso al sitio, únicamente debe elegir el tipo de solicitud que desea realizar, si es el caso que la solicitud requiere ingresar datos adicionales, lo podrá hacer y se generará la solicitud para visualizarla o para descargarla directamente. En el caso de usuarios que no poseen cuenta o son ex-alumnos de la universidad, tienen acceso a un menú con las solicitudes que pueden realizar, luego de elegir la solicitud que deseen se procede a un formulario donde ingresarán sus datos y de igual manera podrán visualizar o descargar la solicitud.

3.3.1.4 Metáfora de cuenta estudiante

Este módulo se encarga de la administración de la cuenta del estudiante, modificación de datos personales del usuario, también aquí se gestiona el cierre de sesión para que luego de que el estudiante utilice el sitio pueda salir

de su cuenta. Adicionalmente si es el caso de olvido de contraseña el estudiante puede recuperar su cuenta mediante un pequeño proceso. El estudiante poseerá su cuenta hasta que el mismo decida borrarla.

3.3.2 Diagrama de clases

La siguiente ilustración muestra como está conformado el programa detallando sus métodos, atributos y como se relacionan.

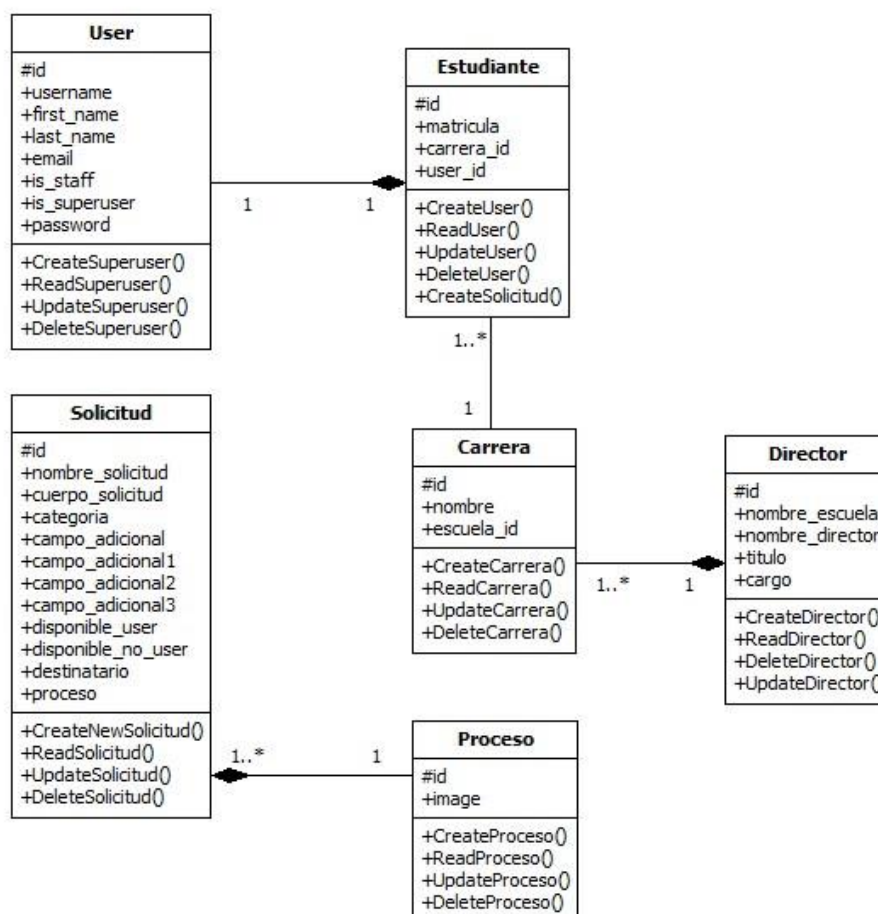


Ilustración 3.1: Diagrama de clases

Fuente: elaboración propia

3.3.3 Tarjetas Clase-Responsabilidad-Colaboración (CRC)

Las tarjetas presentadas a continuación ayudaron al desarrollador a entender de mejor manera los modelos utilizados mediante clases para poder implementarlos correctamente:

Tarjeta CRC: User	
Responsabilidad	Colaboración
Administración de la aplicación en general. Gestión de datos de los modelos.	Solicitud. Proceso. Carrera. Director.

Tabla 3.17: Tarjeta CRC: User

Fuente: elaboración propia

Tarjeta CRC: Estudiante	
Responsabilidad	Colaboración
Gestionar información de estudiante o usuario.	User. Carrera.

Tabla 3.18: Tarjeta CRC: Estudiante

Fuente: elaboración propia

Tarjeta CRC: Carrera	
Responsabilidad	Colaboración
Gestión de la información de las carreras de la universidad.	Director.

Tabla 3.19: Tarjeta CRC: Carrera

Fuente: elaboración propia

Tarjeta CRC: Director	
Responsabilidad	Colaboración
Gestión de las escuelas o facultades de la universidad.	Ninguna.

Tabla 3.20: Tarjeta CRC: Director

Fuente: elaboración propia

Tarjeta CRC: Solicitud	
Responsabilidad	Colaboración
Gestión de la información de las solicitudes académicas.	Proceso.

Tabla 3.21: Tarjeta CRC: Solicitud

Fuente: elaboración propia

Tarjeta CRC: Proceso	
Responsabilidad	Colaboración
Administración de la información de los procesos que puede tener las solicitudes para ser aprobadas.	Ninguna.

Tabla 3.22: Tarjeta CRC: Proceso

Fuente: elaboración propia

3.3.4 Diseño arquitectónico

El diseño arquitectónico en el que se basa la aplicación de solicitudes, es el diseño que propone el *framework* de desarrollo Django, y lo podemos observar en la Ilustración 3.2: Diseño arquitectónico de Django.

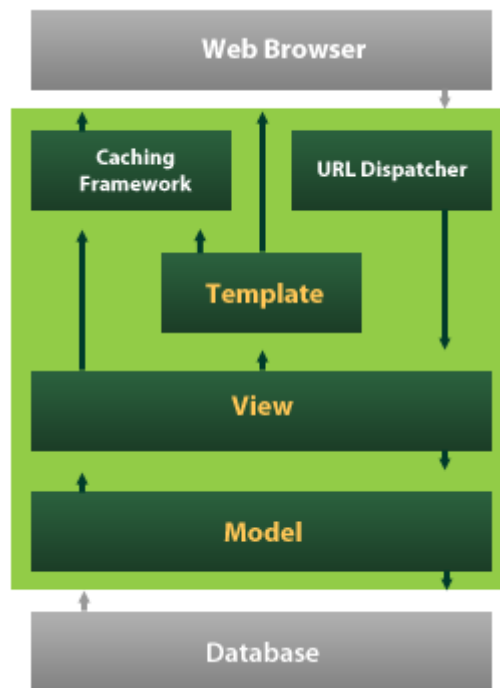


Ilustración 3.2: Diseño arquitectónico de Django

Fuente: (Androulakis, 2010)

El orden de funcionamiento de la arquitectura va de la siguiente manera de acuerdo a (Androulakis, 2010):

1. El *URL dispatcher* gestiona las urls y las direcciona a una vista, el archivo de urls es llamado `Urls.py`.
2. La vista cuyo archivo es `Views.py` se encarga de la funcionalidad del sitio.
3. El modelo que se encuentra en `Models.py` se encarga de definir el modelado de los datos que se van almacenar en la aplicación.
4. El *Caching Framework* se refiere al caché que almacena Django durante un período de tiempo.

5. Finalmente, el *Template* que, respetando las buenas prácticas de Django, los *templates* se almacenan en una carpeta con el mismo nombre, allí permanecen los archivos HTML encargados de la interfaz gráfica de la aplicación.

3.3.5 Diseño de interfaz

La ilustración 3.3 presenta la manera en que los componentes visuales del sitio se muestran, se tomó en cuenta varios lineamientos de usabilidad web de (Carvajal & Saab, 2010) del gobierno de Colombia, entre los que se mencionan:

- La ubicación del logotipo en la parte superior izquierda de igual manera en todas las páginas del sitio.
- El menú ubicarlo a la izquierda por la forma de leer en la región.
- Limpieza y orden en el diseño para no frustrar al usuario.
- Añadir una ruta de migas indicando el desplazamiento del usuario hacia un determinado punto de la página.
- Mostrar la ubicación del usuario.
- Motor de búsqueda dentro de la aplicación con el tamaño adecuado.

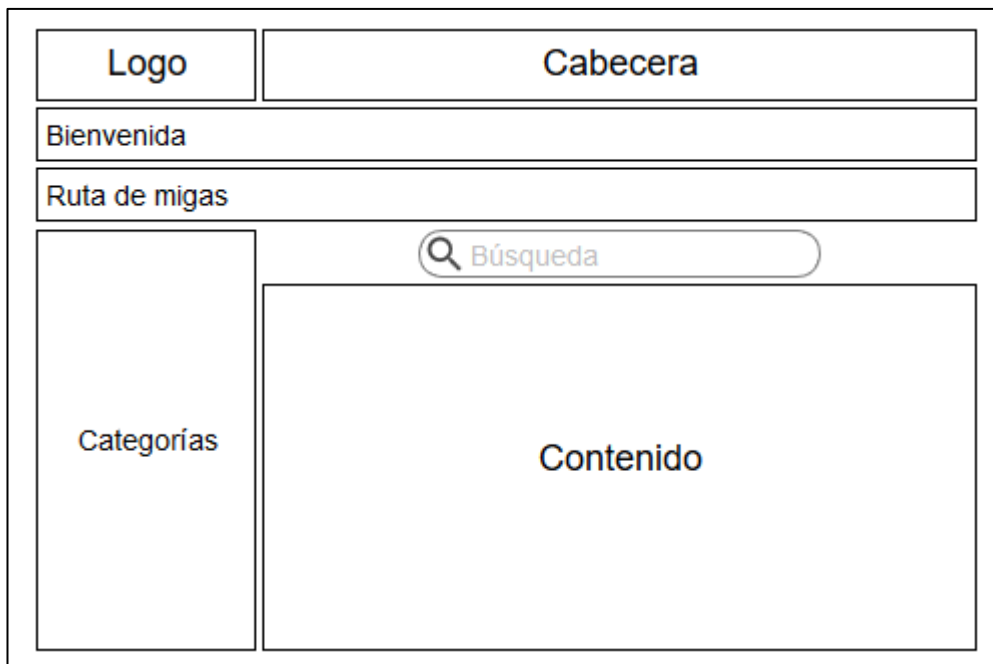


Ilustración 3.3: Diseño de interfaz

Fuente: elaboración propia

3.4 Fase III: Codificación

En esta fase consiste la elaboración de las historias de usuario mediante la escritura de código.

3.4.1 Estándares de codificación

Por parte de Django se programó siguiendo la filosofía que el *framework* de desarrollo motiva a practicar DRY (*Don't repeat yourself*) con su significado en español "No te repitas", esto quiere decir evitar la redundancia. Se ha adoptado de igual forma las buenas prácticas del lenguaje Python PEP8:

- Usar cuatro espacios para indentación.
- Líneas alineadas verticalmente con el carácter utilizado.
- Importaciones de forma correcta y al inicio del documento.
- Nombres de clases de acuerdo a “CapWords”.
- Nombres de funciones en minúscula y palabras separadas por un guion.

3.4.2 Cronograma de actividades

En la ilustración a continuación se detalla las fechas con las que se trabajó para el desarrollo del proyecto:

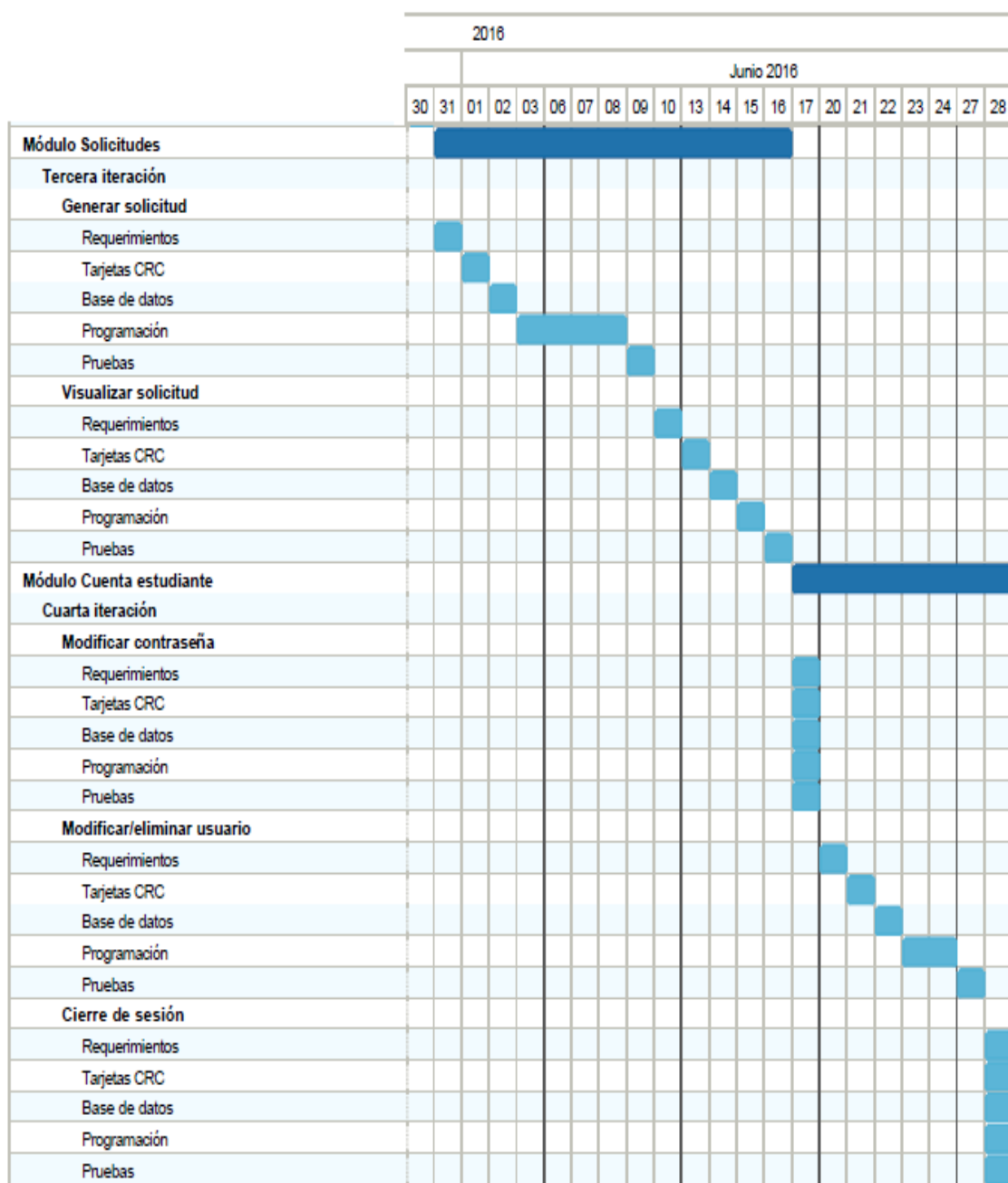


Ilustración 3.5: Cronograma de actividades 2

Fuente: elaboración propia

3.4.3 Codificación de historias de usuario

A continuación, se puede ver porciones de código pertenecientes a la aplicación divididas en historias de usuario planteadas anteriormente.

3.4.3.1 Módulo: Base de datos

3.4.3.1.1 Historia de usuario: Base de datos.

Al utilizar Django para crear una base de datos se crean modelos, cabe mencionar que el *framework* contiene varias tablas definidas en su estructura como la de *User*, los modelos se crean de la siguiente forma:

```
class Estudiante(models.Model):  
    user = models.OneToOneField(User, on_delete=models.CASCADE)  
    matricula = models.CharField(max_length=10, unique=True)  
    carrera = models.ForeignKey(Carrera, on_delete=models.CASCADE,  
null=True, blank=False)
```

Cada modelo creado provee de un campo "id" que a su vez pasa a ser su clave primaria. Para que la base de datos se actualice y entre en funcionamiento con el motor que se haya seleccionado, luego de declarar dicho motor en el archivo de configuración de Django (settings.py):

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql_psycopg2',  
        'NAME': 'SolicitudesApp',  
        'USER': 'admin',  
        'PASSWORD': '****',  
        'HOST': 'localhost',  
        'PORT': '5432',  
    }  
}
```

Se debe ingresar en la terminal los siguientes comandos posicionándose en la carpeta de la aplicación:

- `python manage.py makemigrations`
- `python manage.py migrate`

3.4.3.2 Módulo: Uso de aplicación

3.4.3.2.1 Historia de usuario: Formulario de ingreso/registro

Formulario de ingreso

Para el validar los datos del usuario y confirmar su acceso a la aplicación se utiliza el siguiente código:

```
def auth_user(request):  
    if request.user.is_authenticated():  
        return redirect('/menu-user/?categoria=')  
  
    if request.method == 'POST':  
        username = request.POST.get('username', None)  
        password = request.POST.get('password', None)  
        user = authenticate(username=username, password=password)  
  
        if user is not None:  
            login(request, user)  
            return redirect('/menu-user/?categoria=')  
  
        else:  
            mensaje = 'Cédula o contraseña no válida'  
            context = {  
                'mensaje': mensaje,  
            }  
            return render(request, 'accounts/login.html', context)  
    return render(request, 'accounts/login.html')
```

Formulario de registro

Django facilita la creación de formularios mediante la creación de un archivo llamado forms.py donde se puede crear formularios que aparecerán en la vista del usuario de acuerdo al modelo de datos que se lo requiera, de igual manera en este archivo se pueden realizar las validaciones de los datos correspondientes. La creación del formulario fue de la siguiente manera.

```
class UserForm(forms.ModelForm):  
    password2 = forms.CharField()  
  
    class Meta:  
        model = User  
        fields = ('username', 'first_name', 'last_name', 'email', 'password')
```

De cada campo del formulario se customizaron varios atributos como los errores, la clase de HTML5 a la que pertenece, su *label* o etiqueta y un máximo número de valores que puede ingresar de la manera que se muestra a continuación:

```
username = forms.CharField(  
    required=True,  
    label='Cédula:',  
    max_length=10,  
    widget=forms.TextInput(  
        attrs={'class': 'form-control',  
              'placeholder': 'Ej. 1846378597',  
              'autofocus': 'autofocus'}),  
    error_messages={'unique': 'Cédula anteriormente registrada.'},  
    )
```

Finalmente, el siguiente código indica como se muestra el formulario al usuario:

```

<div class="col-xs-12 col-sm-6">
    {{ form1.username.label_tag }}
    {{ form1.username }}
    {{ form1.username.errors }}
</div>

```

3.4.3.2.2 Historia de usuario: Utilización del servicio

En la interfaz de usuario existe un link en el que las personas que no poseen cuenta puedan acceder al servicio.

```

<h5>¿Estás de visita?</h5>
<a href="{% url 'visitantes:menunouser' %}?categoria=" class="btn btn-info
btn-lg">Soy visitante</a>

```

Para mostrar las solicitudes a las que tienen acceso se utiliza la siguiente línea de código:

```

f = SolicitudesFilter(request.GET, queryset=Solicitud.objects.all()).
    filter(disponible_no_user='Si').
    order_by('nombre_solicitud')

```

3.4.3.3 Módulo: Administrador

3.4.3.3.1 Historia de usuario: Ingresar/modificar solicitudes

El ingreso y modificación de solicitudes no es necesario crear un código nuevo de parte del programador por la razón de que, al crear los modelos de Django, el mismo proporciona la interfaz de administración con las funcionalidades básicas de crear, leer, actualizar y borrar datos también permite filtrar, ordenar, buscar y otros. Lo único que se debe hacer es registrar el modelo en el archivo admin.py para que se pueda mostrar en la sección administrativa de la siguiente forma:

```
@admin.register(Solicitud)

class AdminSolicitud(admin.ModelAdmin):

    list_display = ('nombre_solicitud', 'disponible_user', 'disponible_no_user',)

    list_filter = ('categoria', 'disponible_user', 'disponible_no_user',)

    search_fields = ('nombre_solicitud',)

    form = SolicitudForm
```

3.4.3.3.2 Historia de usuario: Modificar datos carrera/escuela

Como se mencionó anteriormente, únicamente se debe registrar los modelos de Carrera y Escuela para obtener toda funcionalidad administrativa.

3.4.3.4 Módulo: Solicitudes

3.4.3.4.1 Generar solicitud

A continuación, se muestra como generar la solicitud, luego de recibir los datos enviados por el usuario visitante u obtener los datos del usuario registrado y las distintas consultas necesarias para realizar la solicitud. Cabe señalar que para una buena presentación se debe declarar una función con los distintos estilos que llevará el documento como: margen, tipo de letra, tamaño, salto entre párrafos alineación y otros.

```
response = HttpResponse(content_type='application/pdf')
response['Content-Disposition'] = 'attachment; filename="Solicitud.pdf"'
def build_flowables(stylesheet):
a = [
    Paragraph("", ParagraphStyle('default', spaceAfter=40)),
    Paragraph("Ambato, " + dia + " de " + mes + " de " + anio,
stylesheet['default']),
    Paragraph(query_director.get('titulo'), stylesheet['linea_simple']),
    Paragraph(query_director.get('nombre_director'),
stylesheet['linea_simple']),
    Paragraph('%s DE %s' % (query_director.get('cargo'),
query_director.get('nombre_escuela')), stylesheet['linea_simple']),
    Paragraph('Presente', stylesheet['default']),
    Paragraph('De mi consideración:', stylesheet['salto_intermedio']),
```

```

Paragraph('Agradeciéndole por su favorable atención, me suscribo',
stylesheet['default']),
Paragraph('Atentamente,', stylesheet['default']),
Paragraph('%s %s' % (nombre, apellido), stylesheet['linea_simple']),
Paragraph(cedula, stylesheet['default']),
]

```

```

"""Insertar el cuerpo"""
a.insert(7, cuerpo)
return a

```

3.4.3.4.2 Visualizar solicitud

Para la visualización de la solicitud únicamente se cambió la variable response de *“attachment”* a *“inline”* para así no descargarla.

```
response['Content-Disposition'] = 'inline; filename="Solicitud.pdf"'
```

Una condición dentro de la función determina si el usuario pide visualizar o descargar la solicitud.

3.4.3.5 Módulo: Cuenta estudiante

3.4.3.5.1 Modificar contraseña/recuperar cuenta

Modificar contraseña

Lo que tiene que ver la modificación de la contraseña, se trabajó con el Sistema de autenticación de Django que viene por defecto instalado y el uso del formulario llamado PasswordChangeForm. La funcionalidad se detalla en la siguiente función.

```
def change_password(request):  
    if request.method == 'POST':  
        form = PasswordChangeForm(request.user, request.POST)  
        if form.is_valid():  
            user = form.save()  
            update_session_auth_hash(request, user) # Important!  
            messages.success(request, 'Tu contraseña fue actualizada con  
éxito.')  
            return redirect('/change_password/')  
        else:  
            form = PasswordChangeForm(request.user)  
  
    context = {  
        'form': form
```

```
}  
  
return render(request, 'user/change_password.html', context)
```

Recuperar cuenta

Al igual que para modificar contraseña, en la recuperación de cuenta por correo también se utiliza el Sistema de autenticación de Django, y lo más importante de la recuperación es la utilización de 4 vistas incluidas.

1. Password_reset
2. Password_reset_done
3. Password_reset_confirm
4. Password_reset_complete

3.4.3.5.2 Modificar/eliminar usuario

Modificar usuario

Para modificar usuario se crearon dos formularios que contenían los datos del usuario y donde se pueden ingresar nuevos datos, la información es procesada en la siguiente función:

```
def profile_update(request):  
  
    if not request.POST:  
  
        user_form = UpdateUserForm(instance=request.user)
```

```
    estudiante_form =
UpdateEstudianteForm(instance=request.user.estudiante)

    else:

        user_form = UpdateUserForm(data=request.POST,
instance=request.user)

        estudiante_form = UpdateEstudianteForm(data=request.POST,
instance=request.user.estudiante)

        if user_form.is_valid() and estudiante_form.is_valid():

            user = user_form.save(commit=False)

            estudiante = estudiante_form.save(commit=False)

            user.save()

            matricula = request.POST.get('matricula', None)

            estudiante.matricula = matricula.upper()

            estudiante.save()

            messages.success(request, 'Tu perfil fue actualizado.')

template = loader.get_template('user/profile.html')

context = {

    'user_form': user_form,

    'estudiante_form': estudiante_form

}

return HttpResponse(template.render(context, request))
```

Eliminar usuario

Para eliminar el usuario únicamente se selecciona el usuario actual y se lo elimina.

```
def delete_user(request):
    User.objects.get(id=request.user.id).delete()
    messages.info(request, 'Tu usuario ha sido borrado exitosamente.')
    return HttpResponseRedirect('/logout/')
```

3.4.3.5.3 Cierre de sesión

Para todo el funcionamiento de la aplicación se debe crear las direcciones o enlaces que va a tener en un archivo llamado urls.py. Para el cierre de sesión se es importante escribir la url de *logout*.

```
url(r'^logout/$', auth_views.logout, {'next_page': '/'}, name='logout'),
```

Adicionalmente se añade el enlace en la vista del usuario.

```
<li><a href="{% url 'estudiantes:logout' %}"><i class="fa fa-sign-out"></i>
Cerrar sesión</a></li>
```

3.5 Fase IV: Prueba

Como se describió en el capítulo II del presente proyecto, la última fase son las pruebas unitarias y pruebas de aceptación importantes para alcanzar un correcto funcionamiento de la aplicación.

3.5.1 Pruebas unitarias

Estas pruebas fueron importantes para resolver rápidamente ciertas dificultades que se plantearon. Django indica que no se debe realizar un test de los modelos por la razón de que, al crearlos y registrarlos, el *framework* de desarrollo se encarga de su correcto funcionamiento, así que el test se realizó del código escrito por el programador, esto quiere decir que se ha dejado la parte administrativa fuera de los test.

3.5.1.1 Prueba unitaria: Registro de usuario estudiante

Esta prueba simula la creación de un usuario con todos los atributos que el mismo posee para ser utilizado en posteriores pruebas.

```
class EstudianteTest(TestCase):  
    def new_user_test(self):  
        user = User.objects.create(  
            id = 1,  
            username = '1804657362',
```

```
first_name = 'Juan Pablo',  
last_name = 'Pérez Bernard',  
email = 'jp@pucesa.edu.ec',  
password = '1234')
```

```
estudiante = Estudiante.objects.create(  
    id = 1,  
    matricula = 'EAE-1010',  
    carrera_id = 6,  
    user_id = 1)
```

Las validaciones que contiene la aplicación al crear usuario son validar la cédula, que el usuario no esté creado anteriormente mediante el almacenamiento de cédula, correo y matrícula irrepetibles entre estudiantes registrados

3.5.1.2 Prueba unitaria: *Login* de usuario estudiante

En la prueba de autenticación toma de datos la cédula y contraseña, si ambas corresponden se dirigirá a la siguiente url, también se confirma que el usuario se encuentra autenticado verificando que los correos sean iguales como se muestra a continuación:

```
def login_test(self):  
    self.client.login(username='1804657362', password='1234')
```

```
response = self.client.get('/index/', follow=True)
user = User.objects.get(username='1804657362')
self.assertEqual(response.context['email'], 'jp@pucesa.edu.ec')
```

Las validaciones al momento de realizar el ingreso, es la verificación de la existencia del usuario obteniéndose un ingreso exitoso caso contrario mostrará un mensaje de error.

3.5.1.3 Prueba unitaria: Modificación de perfil de usuario estudiante

Para modificaciones de la cuenta funciona de la misma manera que al crear un usuario.

```
def update_user_test(self):
    user = User.objects.filter(pk=1).update(
        username = '1804657362',
        first_name = 'Juan Pablo',
        last_name = 'Pérez Bernard',
        email = 'jp@pucesa.edu.ec')

    estudiante = Estudiante.objects.filter(pk=user_id).update(
        matricula = 'EAE-1010',
        carrera_id = 6)
```

3.5.1.4 Prueba unitaria: Cambio de contraseña de usuario estudiante

Para cambiar la contraseña el test que se realizó comprende la selección del usuario al que se va a cambiar la contraseña, en este caso será un usuario que ha abierto su sesión en cambio para recuperar la cuenta en necesario ingresar un correo y con el que se ha registrado el usuario y de esta forma llegar a él para establecer la nueva contraseña.

```
from django.contrib.auth.models import User
users = User.objects.get(id=1)
user.set_password('12345')
user.save()
```

3.5.1.5 Prueba unitaria: Eliminar cuenta de usuario estudiante

Al igual que la anterior prueba se selecciona el usuario a borrar y se lo elimina.

```
from django.contrib.auth.models import User
users = User.objects.get(id=1)
users.delete()
```

Las pruebas también se lo realizan dentro de la aplicación para verificar que los enlaces o botones funcionen correctamente.

3.5.1.6 Prueba unitaria: Generación de solicitud

En la creación de las solicitudes, la prueba es la misma entre el usuario y visitante con la única diferencia que en lugar de matrícula, se mostrará la cédula en la solicitud. En esta prueba se revisó únicamente que la vista responda en la aplicación, el funcionamiento final se lo probó durante el desarrollo por la cantidad de datos y consultas que requería.

```
def test_details(self):  
    request = self.factory.get('/user/generar-user.html')  
    request.user = self.user  
    response = my_view(request)  
    self.assertEqual(response.status_code, 200)
```

3.5.2 Pruebas de aceptación

En esta fase que es la última de la metodología de desarrollo, se revisó el correcto funcionamiento de la aplicación mediante un planteamiento de escenarios posibles por los que pueda pasar el *software*. En el presente proyecto las pruebas se resumen a continuación de acuerdo a cada historia de usuario que se ha creado.

Número de prueba:	1
Módulo:	Base de datos
Nombre de prueba:	Base de datos
Descripción:	
Correcta visualización de los datos en la sección administrativa.	
Pasos a ejecutar:	
- Abrir el administrador	
Resultado esperado: La aplicación responde correctamente.	

Tabla 3.23: Prueba de aceptación N° 1

Fuente: elaboración propia

Número de prueba:	2
Módulo:	Uso de aplicación
Nombre de prueba:	Formulario de ingreso/registro
Descripción:	
Registro de usuario con datos erróneos.	
Pasos a ejecutar:	
<ul style="list-style-type: none"> - Ingresar letras en lugar de números en la cédula - Dejar campos vacíos - Ingresar datos de un usuario ya registrado 	
Resultado esperado: La aplicación responde como debe hacerlo e indica que no es válido ingresar ciertos valores.	

Tabla 3.24: Prueba de aceptación N° 2

Fuente: elaboración propia

Número de prueba:	3
Módulo:	Uso de aplicación
Nombre de prueba:	Utilización del servicio
Descripción:	
Verificar la sección visitante se muestre correctamente	
Pasos a ejecutar:	
- Verificar los enlaces dentro de la vista	
Resultado esperado: El test realizado fue correcto.	

Tabla 3.25: Prueba de aceptación N° 3

Fuente: elaboración propia

Número de prueba:	4
Módulo:	Administrador
Nombre de prueba:	Ingresar/modificar solicitudes
Descripción:	
Administración de solicitudes.	
Pasos a ejecutar:	
<ul style="list-style-type: none"> - Ingresar solicitudes con datos incompletos - Modificar solicitudes - Eliminar solicitudes - Buscar solicitudes - Filtrar solicitudes 	
Resultado esperado: Las solicitudes en la sección administrativa funciona correctamente.	

Tabla 3.26: Prueba de aceptación N° 4

Fuente: elaboración propia

Número de prueba:	5
Módulo:	Administrador
Nombre de prueba:	Modificar datos de carrera/escuela
Descripción:	
Administración de carrera y escuelas de la universidad.	
Pasos a ejecutar:	
<ul style="list-style-type: none"> - Ingresar carreras y escuelas con datos incompletos - Modificar carreras y escuelas - Eliminar carreras y escuelas - Buscar carreras y escuelas 	
Resultado esperado: Las carreras y escuelas en la sección administrativa funciona correctamente.	

Tabla 3.27: Prueba de aceptación N° 5

Fuente: elaboración propia

Número de prueba:	6
Módulo:	Solicitudes
Nombre de prueba:	Generar solicitud
Descripción:	
Generar solicitud de usuarios visitantes.	
Pasos a ejecutar:	
<ul style="list-style-type: none"> - Seleccionar la solicitud a realizar - Ingresar datos personales - Ingresar datos opcionales de acuerdo a cada solicitud - Descargar solicitud 	
Resultado esperado: Todos los datos son requeridos para poder realizar la solicitud, al momento de ingresarlos se descarga la solicitud.	

Tabla 3.28: Prueba de aceptación N° 6

Fuente: elaboración propia

Número de prueba:	7
Módulo:	Solicitudes
Nombre de prueba:	Generar solicitud
Descripción:	
Generar solicitud de usuarios registrados.	
Pasos a ejecutar:	
<ul style="list-style-type: none"> - Seleccionar la solicitud a realizar - Ingresar datos opcionales de acuerdo a cada solicitud - Descargar solicitud 	
Resultado esperado: La generación de solicitudes en ambos casos de usuarios registrados o no cumple con lo establecido.	

Tabla 3.29: Prueba de aceptación N° 7

Fuente: elaboración propia

Número de prueba:	8
Módulo:	Solicitudes
Nombre de prueba:	Visualizar solicitud
Descripción:	
Visualizar solicitud.	
Pasos a ejecutar:	
<ul style="list-style-type: none"> - Se cumplen los pasos de las pruebas de aceptación 6 y 7 - Se presiona el botón correspondiente para visualizar la solicitud 	
Resultado esperado: Se visualiza la solicitud con todos los datos correspondientes.	

Tabla 3.30: Prueba de aceptación N° 8

Fuente: elaboración propia

Número de prueba:	9
Módulo:	Cuenta estudiante
Nombre de prueba:	Modificar contraseña
Descripción:	
Modificar la contraseña del usuario.	
Pasos a ejecutar:	
<ul style="list-style-type: none"> - Ir al apartado de cambiar contraseña - Cambiar a una nueva contraseña 	
Resultado esperado: El usuario actualiza la contraseña sin problema.	

Tabla 3.31: Prueba de aceptación N° 9

Fuente: elaboración propia

Número de prueba:	10
Módulo:	Cuenta estudiante
Nombre de prueba:	Recuperar contraseña
Descripción:	
Recuperar cuenta de usuario por olvido de contraseña	
Pasos a ejecutar:	
<ul style="list-style-type: none"> - Seleccionar el link de olvido de contraseña - Ingresar el correo perteneciente a la cuenta - Cambiar a una nueva contraseña 	
Resultado esperado: El usuario recupera su cuenta sin problema.	

Tabla 3.32: Prueba de aceptación N° 10

Fuente: elaboración propia

Número de prueba:	11
Módulo:	Cuenta estudiante
Nombre de prueba:	Modificar/eliminar usuario
Descripción:	
Actualizar datos del usuario o eliminarlo.	
Pasos a ejecutar:	
<ul style="list-style-type: none"> - Ir al perfil del usuario - Modificar datos - Guardar datos - Verificar datos actualizados - Eliminar usuario 	
Resultado esperado: El usuario es capaz de actualizar sus datos y borrar su cuenta si lo requiere.	

Tabla 3.33: Prueba de aceptación N° 11

Fuente: elaboración propia

Número de prueba:	12
Módulo:	Cuenta estudiante
Nombre de prueba:	Cierre de sesión
Descripción:	
Cerrar sesión.	
Pasos a ejecutar:	
<ul style="list-style-type: none"> - Seleccionar el enlace de cierre de sesión 	
Resultado esperado: La sesión se cierra sin problema.	

Tabla 3.34: Prueba de aceptación N° 12

Fuente: elaboración propia

3.6 Acceso a la aplicación

Para que el usuario pueda utilizar la aplicación no es necesario que la misma se distribuya en tiendas online de los distintos dispositivos móviles, únicamente el usuario debe tener acceso a internet y abrir el enlace de la aplicación. Esta acción la puede realizar desde cualquier dispositivo sin perder ningún detalle menos aún funcionalidad por la razón de que la aplicación posee un diseño responsivo a cada *gadget* tecnológico.

CAPÍTULO IV

ANÁLISIS Y VALIDACIÓN DE RESULTADOS

4.1 Análisis de resultados

Para finalizar el proyecto investigativo, se analizó los resultados mediante ilustraciones que se muestran en los siguientes puntos.

4.1.1 Aplicación web con diseño responsivo

La aplicación funciona correctamente en los distintos tamaños de pantalla de los dispositivos móviles que se usan actualmente, por lo que las pruebas que se muestran a continuación serán de los dispositivos básicos para analizar una página web adaptativa como son: móvil, tableta y laptop.

Es importante mencionar que el diseño responsivo solo afecta a la vista del usuario, la vista administrativa al ser utilizada para ese fin y pertenecer a la vista proporcionada por el *framework* de desarrollo no cumple con dicha característica.

4.1.2 SolicitudesApp

Módulo de uso de aplicación

En las siguientes ilustraciones se muestra lo que el usuario verá al abrir el enlace de la aplicación desde cualquier dispositivo.



Ilustración 4.1: Pantalla de inicio vista móvil

Fuente: elaboración propia

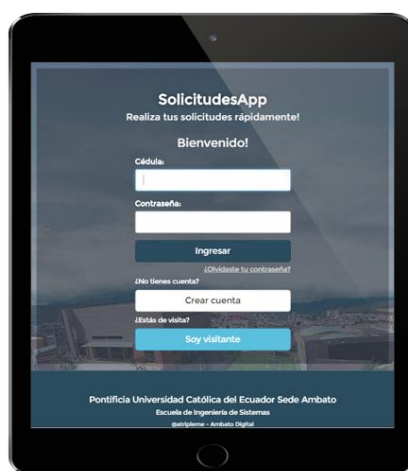


Ilustración 4.2: Pantalla de inicio vista de tableta

Fuente: elaboración propia



Ilustración 4.3: Pantalla de inicio vista de laptop

Fuente: elaboración propia

Formulario de registro

Para que el usuario pueda tener una cuenta es necesario su registro, en la ilustración 4.4, 4.5 y 4.6 se puede ver cómo la misma se presenta en cada dispositivo, este registro es únicamente válido para el estudiante mas no para el usuario administrador. En este registro existen campos únicos entre usuarios como son: cédula, correo y matrícula.



Ilustración 4.4: Registro de usuario vista móvil

Fuente: elaboración propia

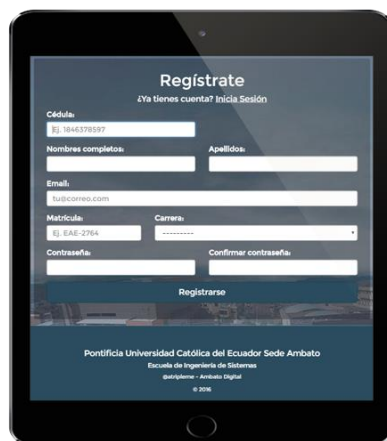


Ilustración 4.5: Registro de usuario vista tableta

Fuente: elaboración propia

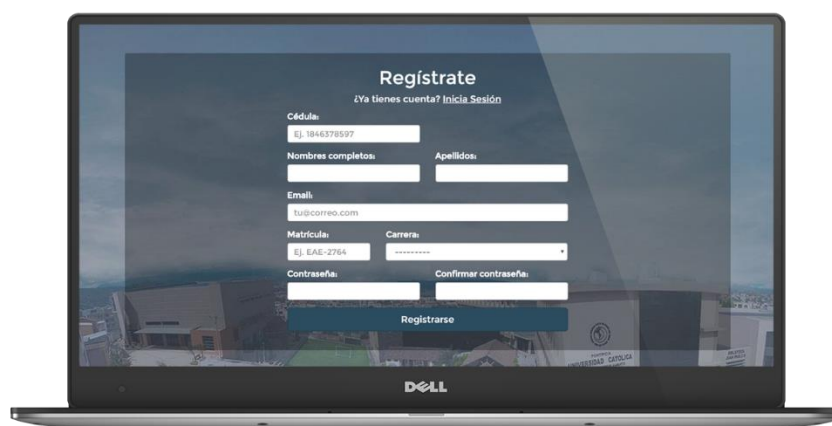


Ilustración 4.6: Registro de usuario vista laptop

Fuente: elaboración propia

Acceso al menú de solicitudes - usuarios visitantes

Como se puede observar en las ilustraciones 4.1, 4.2 y 4.3, existe un botón que indica que el usuario es visitante, en las siguientes ilustraciones se podrá observar el menú de solicitudes habilitadas para usuarios visitantes. Se puede observar en la parte superior un menú que sirve para enlazar al visitante a crear una cuenta y se encuentra desplegado en la ilustración 4.9.



Ilustración 4.7: Menú de visitantes vista móvil

Fuente: elaboración propia

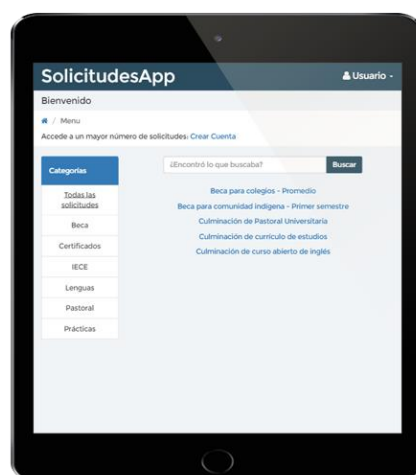


Ilustración 4.8: Menú de visitantes vista tableta

Fuente: elaboración propia



Ilustración 4.9: Menú de visitantes vista laptop

Fuente: elaboración propia

Acceso al menú de solicitudes - usuarios registrados

Para acceder a una mayor cantidad de solicitudes, el usuario deberá iniciar sesión. En la ilustración 4.10 se puede observar el mensaje de error que da la aplicación al ingresar datos incorrectos durante el inicio de sesión. Si sucediera que el usuario ha olvidado su contraseña existe un enlace para recuperar la cuenta y su funcionamiento se detallará más adelante.

Una captura de pantalla de una interfaz de usuario para el inicio de sesión. El encabezado dice "Bienvenido!". Hay dos campos de entrada: "Cédula:" y "Contraseña:". Debajo de los campos, hay un mensaje de error que dice "Cédula o contraseña no válida" con un icono de advertencia. Debajo del mensaje de error hay un botón que dice "Ingresar". En la parte inferior derecha, hay un enlace que dice "¿Olvidaste tu contraseña?".

Ilustración 4.10: Inicio de sesión incorrecto

Fuente: elaboración propia

Datos correctos al iniciar sesión

Cuando los datos estén correctos al iniciar sesión, la aplicación mostrará el siguiente menú de solicitudes con una bienvenida personalizada con el nombre del usuario, adicionalmente el menú superior presentará otras funciones como se puede observar en la ilustración 4.13.



Ilustración 4.11: Menú de solicitudes de usuario vista móvil

Fuente: elaboración propia



Ilustración 4.12: Menú de solicitudes de usuario vista tableta

Fuente: elaboración propia



Ilustración 4.13: Menú de solicitudes de usuario vista laptop

Fuente: elaboración propia

Módulo de solicitudes

Generación de solicitud básica - usuarios visitantes

En las ilustraciones a continuación se muestra el formulario de una solicitud básica luego de que el usuario visitante haya elegido una en el menú, por solicitud básica se entiende que solo pide datos de usuario sin datos adicionales.

A smartphone displaying the 'SolicitudesApp' interface. The screen shows a 'Bienvenido' (Welcome) message and a navigation menu with 'Menu' and 'Generar'. Below this is the title 'Datos para la solicitud de:' followed by the subtitle 'Culminación de currículo de estudios'. A note states '* Todos los datos son obligatorios'. The form includes fields for 'Cédula:' (with a sample ID 'EJ: 1846378997'), 'Nombres:', 'Apellidos:', and 'Carrera:' (with a dropdown menu showing 'Diseño Industrial'). At the bottom, there are two buttons: 'Visualizar' (blue) and 'Descargar' (red). A 'Proceso' section at the very bottom indicates 'Tu solicitud seguirá el siguiente flujo para ser aprobada:'.

Ilustración 4.14: Formulario de solicitud básica visitante vista móvil

Fuente: elaboración propia

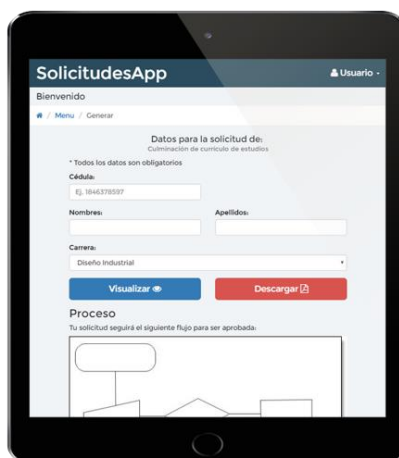
A tablet displaying the 'SolicitudesApp' interface. The layout is similar to the mobile view but scaled for a larger screen. It features the same 'Bienvenido' message, navigation menu, and form fields for 'Cédula:', 'Nombres:', 'Apellidos:', and 'Carrera:'. The 'Visualizar' and 'Descargar' buttons are also present. The 'Proceso' section at the bottom shows a flowchart diagram.

Ilustración 4.15: Formulario de solicitud básica visitante vista tableta

Fuente: elaboración propia

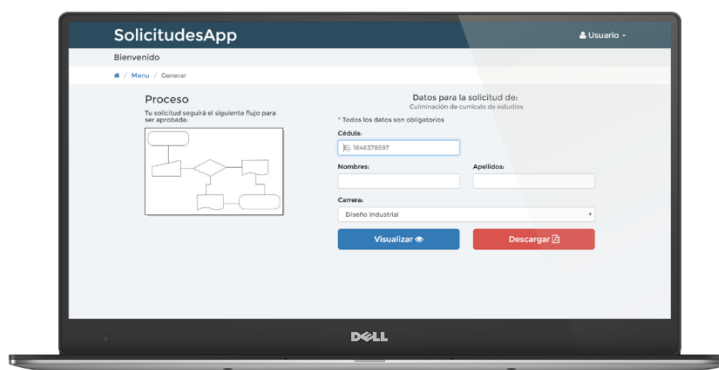


Ilustración 4.16: Formulario de solicitud básica visitante vista laptop

Fuente: elaboración propia

Generación de solicitud con campos adicionales - usuarios visitantes

Las siguientes vistas corresponden a solicitudes que requieren el ingreso de campos adicionales, se muestra el ejemplo de beca para colegio, un campo necesario y único para esta solicitud es el período al que va a ingresar el estudiante y la nota promedio de grado obtenida en la unidad educativa, estos campos adicionales varían entre solicitudes.



Ilustración 4.17: Formulario de solicitud con campos adicionales visitante vista móvil

Fuente: elaboración propia

The image shows a tablet displaying the 'SolicitudesApp' interface. At the top, it says 'Bienvenido' and 'Usuario'. Below that is a navigation bar with 'Menu' and 'Generar'. The main content area is titled 'Datos para la solicitud de:' and includes a sub-header 'Forma para colegios - Promoción'. A note states '* Todos los datos son obligatorios'. The form contains several input fields: 'Cédula' (with an example 'Ej. 1846378597'), 'Nombres', 'Apellidos', 'Carrera' (a dropdown menu with 'Diseño Industrial' selected), 'Período (Ej. Septiembre - Febrero 2016)', and 'Nota promedio de grado'. At the bottom, there are two buttons: 'Visualizar' (blue) and 'Descargar' (red). Below the form, there is a section titled 'Proceso' with the text 'Tu solicitud seguirá el siguiente flujo para ser aprobada.' and a small flowchart diagram.

Ilustración 4.18: Formulario de solicitud con campos adicionales visitante vista tableta

Fuente: elaboración propia

The image shows a laptop displaying the 'SolicitudesApp' interface. The layout is similar to the tablet version but adapted for a larger screen. On the left side, there is a 'Proceso' section with a flowchart diagram. The main form area on the right contains the same fields as the tablet version: 'Cédula', 'Nombres', 'Apellidos', 'Carrera', 'Período (Ej. Septiembre - Febrero 2016)', and 'Nota promedio de grado'. The 'Visualizar' and 'Descargar' buttons are also present at the bottom.

Ilustración 4.19: Formulario de solicitud con campos adicionales visitante vista laptop

Fuente: elaboración propia

Generación de solicitud básica - usuarios registrados

Al igual que se mencionó anteriormente, luego seleccionar la solicitud que el usuario desea existirá un formulario, pero en este sus datos se encuentran visibles en los campos con el atributo *readonly*.

SolicitudesApp

Bienvenido Mario Alejandro

Menú / Generar

Datos para la solicitud de:
Culminación de currículo de estudios

* Puede modificar la información básica en su perfil

Cédula:

Matrícula:

Nombres:

Apellidos:

Carrera:

Ilustración 4.20: Formulario de solicitud básica usuario vista móvil

Fuente: elaboración propia

SolicitudesApp Usuario

Bienvenido Mario Alejandro

Menú / Generar

Datos para la solicitud de:
Culminación de currículo de estudios

* Puede modificar la información básica en su perfil

Cédula: Matrícula:

Nombres: Apellidos:

Carrera:

[Visualizar](#) [Descargar](#)

Proceso
Tu solicitud seguirá el siguiente flujo para ser aprobada:

```

graph TD
    A[Inicio] --> B[Registro]
    B --> C[Validación]
    C --> D[Revisión]
    D --> E[Finalización]
  
```

Ilustración 4.21: Formulario de solicitud básica usuario vista tableta

Fuente: elaboración propia

SolicitudesApp Usuario

Bienvenido Mario Alejandro

Menú / Generar

Proceso
Tu solicitud seguirá el siguiente flujo para ser aprobada:

```

graph TD
    A[Inicio] --> B[Registro]
    B --> C{Validación}
    C --> D[Revisión]
    D --> E[Finalización]
  
```

Datos para la solicitud de:
Culminación de currículo de estudios

* Puede modificar la información básica en su perfil

Cédula: Matrícula:

Nombres: Apellidos:

Carrera:

[Visualizar](#) [Descargar](#)

Ilustración 4.22: Formulario de solicitud básica usuario vista laptop

Fuente: elaboración propia

Generación de solicitud con campos adicionales - usuarios registrados

De manera que se mostró con las solicitudes con campos adicionales de usuarios visitantes, en el caso de los usuarios registrados, sus datos aparecerán en los campos ya ingresados y los campos adicionales de cada solicitud aparecerán en la parte inferior.

A smartphone displaying the 'SolicitudesApp' interface. The form includes fields for 'Matrícula' (EIS-1210), 'Nombres' (Mario Alejandro), 'Apellidos' (Moreno Montero), and 'Carrera' (Ingeniería en Sistemas). Below these fields, there is a note: '* Ingrese los siguientes datos adicionales. Semestre (Ej. Primer, segundo, etc.):'. At the bottom, there are two buttons: 'Visualizar' (blue) and 'Descargar' (red).

Ilustración 4.23: Formulario de solicitud con campos adicionales usuario vista móvil

Fuente: elaboración propia

A tablet displaying the 'SolicitudesApp' interface. The top bar shows 'Bienvenido Mario Alejandro' and 'Usuario'. Below the header, there is a 'Menu / Generar' link. The main section is titled 'Datos para la solicitud de:' and 'Beca para comunidad indígena'. It includes a note: '* Puede modificar la información básica en su perfil'. The form fields are arranged in a grid: 'Cédula' (with a masked input), 'Matrícula' (EIS-1210), 'Nombres' (Mario Alejandro), 'Apellidos' (Moreno Montero), and 'Carrera' (Ingeniería en Sistemas). Below these, there is a note: '* Ingrese los siguientes datos adicionales. Semestre (Ej. Primer, segundo, etc.):'. At the bottom, there are two buttons: 'Visualizar' (blue) and 'Descargar' (red). Below the buttons, there is a section titled 'Proceso' with the text: 'Tu solicitud seguirá el siguiente flujo para ser aprobada:'. At the very bottom, there is a progress indicator with four colored steps: green, red, blue, and red.

Ilustración 4.24: Formulario de solicitud con campos adicionales usuario vista tableta

Fuente: elaboración propia



Ilustración 4.25: Formulario de solicitud con campos adicionales usuario vista laptop

Fuente: elaboración propia

Es importante mencionar que, en el caso de las solicitudes de inglés o pastoral, se sigue el mismo proceso independiente de estar registrado en una carrera para los usuarios que poseen cuenta; para los que no, en el formulario de solicitud no aparecerá la opción de elegir una carrera.

* Todos los datos son obligatorios

Cédula:
Ej. 1846378597

Nombres:

Apellidos:

* Ingrese los siguientes datos adicionales:

Materias:

[Visualizar](#) [Descargar](#)

Ilustración 4.26: Formulario de solicitud con campo adicional de inglés/pastoral visitante

Fuente: elaboración propia

* Puede modificar la información básica en su perfil

Cédula: 1846378597

Matrícula: EIS-1210

Nombres: Mario Alejandro

Apellidos: Moreno Montero

[Visualizar](#) [Descargar](#)

Ilustración 4.27: Formulario de solicitud básica de inglés/pastoral usuario

Fuente: elaboración propia

Visualización de solicitud

Las ilustraciones 4.28, 4.29 y 4.30 corresponden a la vista de la solicitud realizada. Los tipos de solicitudes que puede realizar la aplicación con mayor detalle se muestran en los anexos, (ver anexo 4).



Ilustración 4.28: Solicitud vista móvil

Fuente: elaboración propia



Ilustración 4.29: Solicitud vista tableta

Fuente: elaboración propia

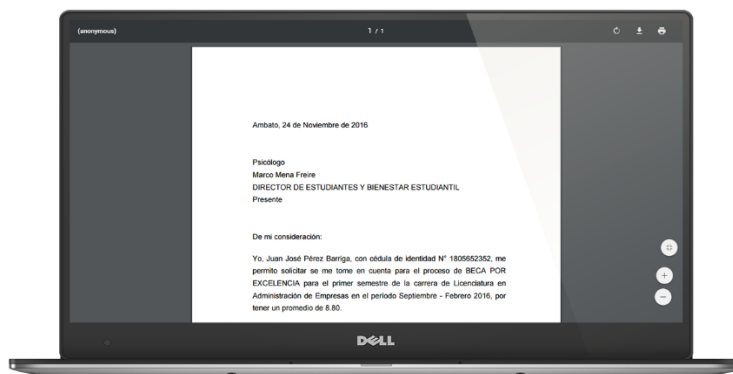


Ilustración 4.30: Solicitud vista laptop

Fuente: elaboración propia

Descarga de solicitud

Al igual que la sección anterior después de ingresar los datos que requiere la aplicación, al presionar el botón de descargar, la solicitud se descargará sin problema a excepción de los usuarios que estén utilizando la aplicación desde un iPhone o iPad, donde la solicitud se mostrará de nuevo y se debe poner la opción de abrir en iBooks y al hacerlo se descargará o si es el caso al presionar el botón “Más” se puede agregar la solicitud a iCloud que es un servicio de almacenamiento en la nube de Apple o compartirla entre distintas aplicaciones.



Ilustración 4.31: Opción de Abrir en iBooks en iPhone

Fuente: elaboración propia



Ilustración 4.32: Opción Más en iPhone

Fuente: elaboración propia

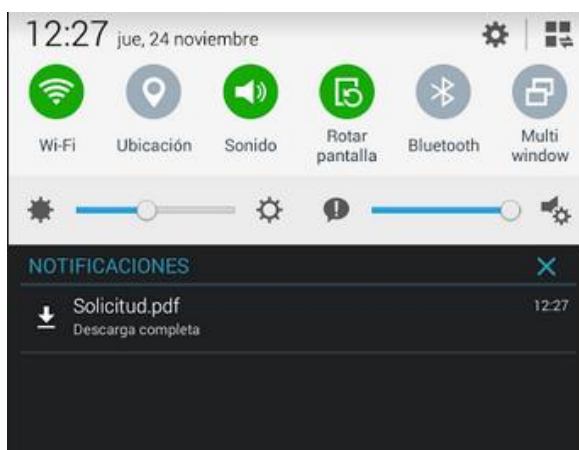


Ilustración 4.33: Solicitud descargada en dispositivo Android

Fuente: elaboración propia

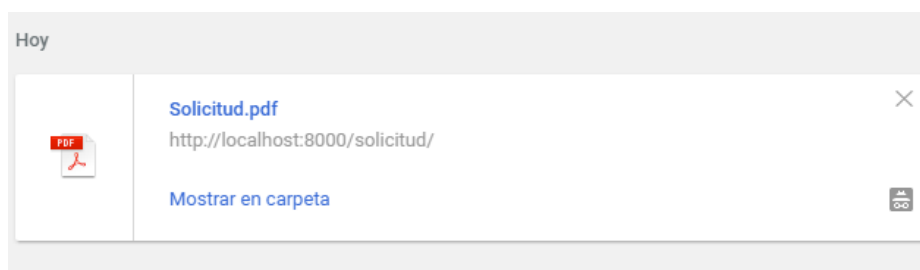


Ilustración 4.34: Solicitud descargada en Laptop

Fuente: elaboración propia

Módulo de cuenta estudiante

Modificar Usuario

En la modificación del usuario se trata de un formulario similar al de registro, adicionalmente se muestra información sobre el último ingreso del mismo.



Ilustración 4.35: Modificar usuario vista móvil

Fuente: elaboración propia



Ilustración 4.36: Modificar usuario vista tableta

Fuente: elaboración propia

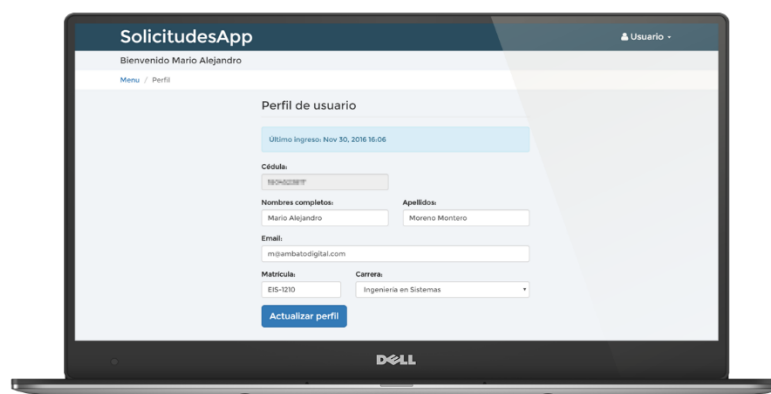


Ilustración 4.37: Modificar usuario vista laptop

Fuente: elaboración propia

Eliminar usuario

En la parte inferior del perfil de usuario se muestra una sección para eliminar la cuenta.

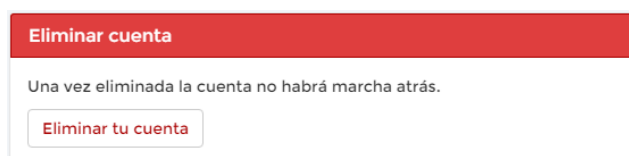


Ilustración 4.38: Sección de eliminar cuenta

Fuente: elaboración propia

Al eliminar cuenta se puede apreciar la siguiente vista de confirmación:



Ilustración 4.39: Confirmación eliminar usuario vista móvil

Fuente: elaboración propia

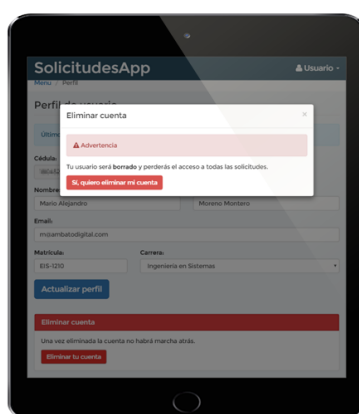


Ilustración 4.40: Confirmación eliminar usuario vista tableta

Fuente: elaboración propia

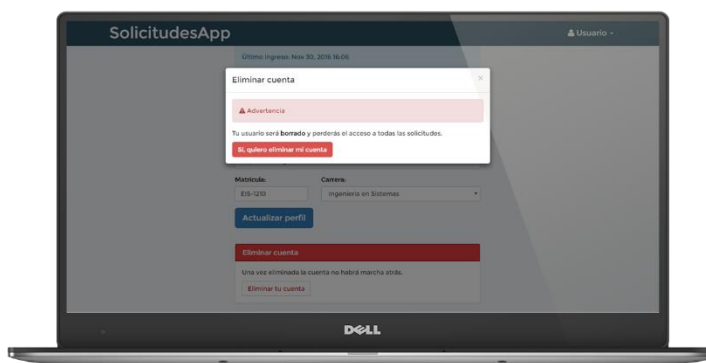


Ilustración 4.41: Confirmación eliminar usuario vista laptop

Fuente: elaboración propia

Modificar contraseña

La modificación de contraseña presenta un formulario donde el usuario debe ingresar su antigua contraseña, la contraseña nueva y su confirmación, adicionalmente se muestra varios consejos para tener una contraseña segura y que debe contener la nueva contraseña.



Ilustración 4.42: Cambio de contraseña vista móvil

Fuente: elaboración propia



Ilustración 4.43: Cambio de contraseña vista tableta

Fuente: elaboración propia



Ilustración 4.44: Cambio de contraseña vista laptop

Fuente: elaboración propia

Recuperar contraseña

La vista de recuperación de cuenta muestra únicamente el formulario de ingreso del correo del usuario.



Ilustración 4.45: Restablecer contraseña vista móvil

Fuente: elaboración propia



Ilustración 4.46: Restablecer contraseña vista tableta

Fuente: elaboración propia

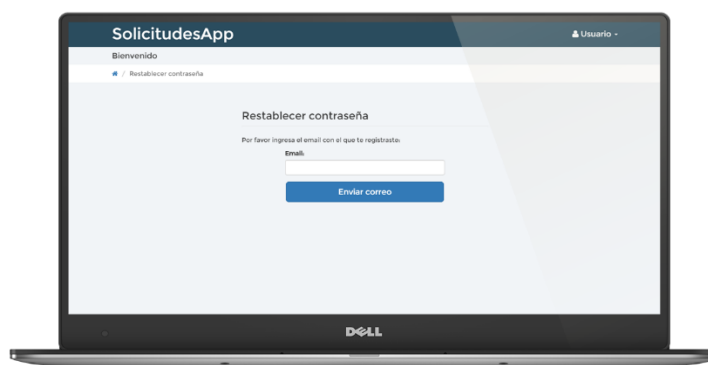


Ilustración 4.47: Restablecer contraseña vista laptop

Fuente: elaboración propia

En la ilustración 4.48 se puede observar el email de recuperación.



Ilustración 4.48: Email de recuperación

Fuente: elaboración propia

Cuando el link es válido muestra el formulario de cambio de contraseña.



Ilustración 4.49: Formulario de cambio de contraseña vista móvil

Fuente: elaboración propia

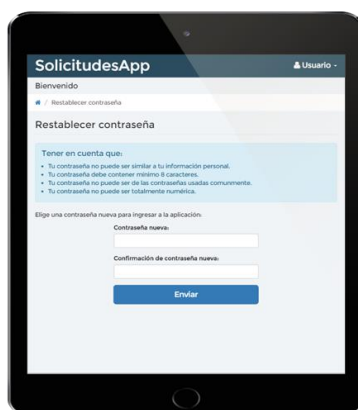


Ilustración 4.50: Formulario de cambio de contraseña vista tableta

Fuente: elaboración propia

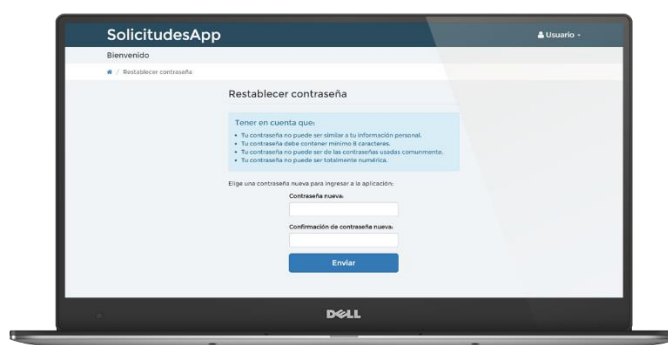


Ilustración 4.51: Formulario de cambio de contraseña vista laptop

Fuente: elaboración propia

Un mensaje de error es mostrado cuando se ingresa al enlace de recuperación de contraseña previamente utilizado.



Ilustración 4.52: Mensaje de error de enlace utilizado

Fuente: elaboración propia

Módulo de administrador

Como se mencionó anteriormente, el módulo de administración al ser parte del *framework* Django, no posee diseño responsivo y se lo observa adecuadamente en computadoras de escritorio y laptops.

Inicio de Sesión

La ilustración 4.53 corresponde al inicio de sesión de la parte administrativa de la aplicación.

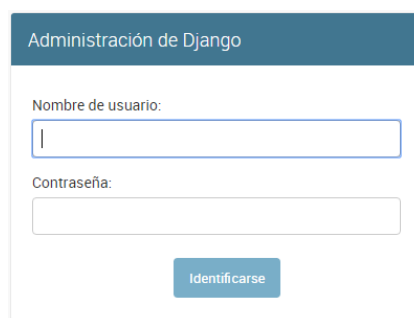
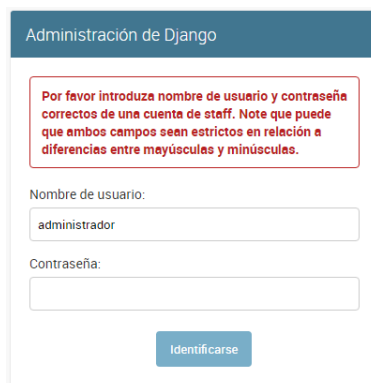


Ilustración 4.53: Ingreso a la administración de Django

Fuente: elaboración propia

Si el inicio de sesión es incorrecto entonces la se mostrará un mensaje de error.



Administración de Django

Por favor introduza nombre de usuario y contraseña correctos de una cuenta de staff. Note que puede que ambos campos sean estrictos en relación a diferencias entre mayúsculas y minúsculas.

Nombre de usuario:
administrador

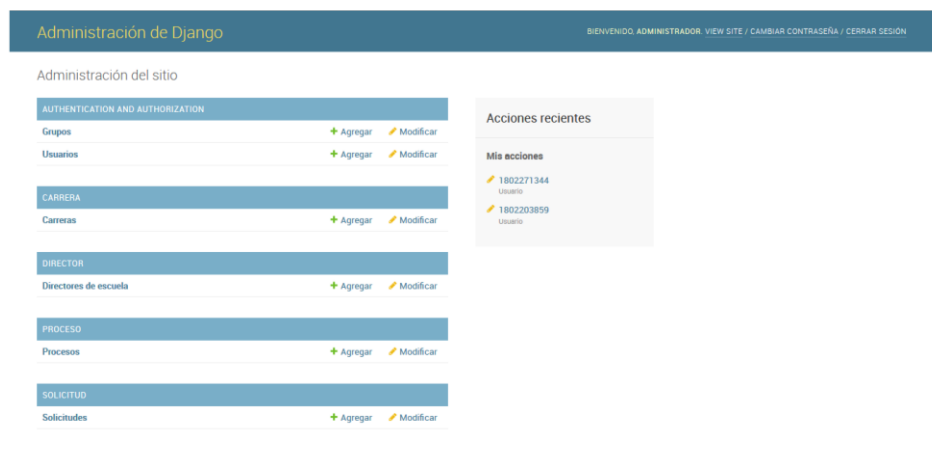
Contraseña:

Identificarse

Ilustración 4.54: Mensaje de error de ingreso

Fuente: elaboración propia

Cuando el inicio de sesión es correcto la sección administrativa de la aplicación, muestra los módulos creados: Usuarios, Carreras, Directores de Escuela, Procesos y Solicitudes y las acciones recientes efectuadas por el administrador actual.



Administración de Django BIENVENIDO, ADMINISTRADOR. VIEW SITE / CAMBIAR CONTRASEÑA / CERRAR SESIÓN

Administración del sitio

AUTHENTICATION AND AUTHORIZATION

Grupos	+ Agregar	Modificar
Usuarios	+ Agregar	Modificar

CARRERA

Carreras	+ Agregar	Modificar
----------	-----------	-----------

DIRECTOR

Directores de escuela	+ Agregar	Modificar
-----------------------	-----------	-----------

PROCESO

Procesos	+ Agregar	Modificar
----------	-----------	-----------

SOLICITUD

Solicitudes	+ Agregar	Modificar
-------------	-----------	-----------

Acciones recientes

Mis acciones

- 1802271344 Usuario
- 1802203859 Usuario

Ilustración 4.55: Vista administrativa

Fuente: elaboración propia

Módulo Usuarios

Como se puede ver en la ilustración 4.55, el administrador puede acceder a todos los módulos de la aplicación para crear, ver, modificar y eliminar datos. En el módulo de Usuarios el administrador puede filtrar datos con un menú en la parte derecha o también existe un formulario donde puede buscarlo.

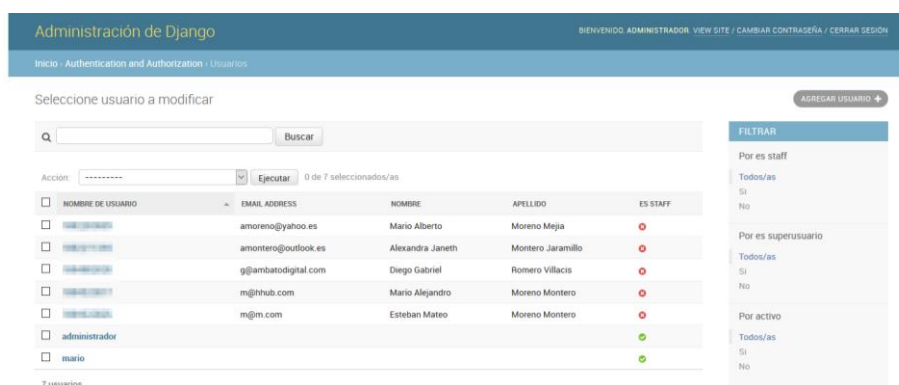


Ilustración 4.56: Vista administrativa módulo Usuarios

Fuente: elaboración propia

Agregar administrador

Para agregar un nuevo usuario administrador, la ilustración 4.57 muestra la vista de agregar usuario. Al crear un administrador, no será necesario agregar los campos de la sección Estudiante.

The screenshot shows the Django administration interface for adding a user. The page title is "Administración de Django" and the breadcrumb is "Inicio - Authentication and Authorization - Usuarios - Agregar usuario". The main heading is "Agregar usuario". Below this, there is a sub-heading "Primero introduzca un nombre de usuario y una contraseña. Luego podrá configurar opciones adicionales acerca del usuario." The form includes three input fields: "Nombre de usuario:" (with a note: "Obligatorio. Longitud máxima 30 caracteres alfanuméricos (letras, dígitos y @/./+/_/) solamente."), "Contraseña:" (with a note: "Obligatorio. Longitud máxima 30 caracteres alfanuméricos (letras, dígitos y @/./+/_/) solamente."), and "Confirmación de contraseña:" (with a note: "Enter the same password as before, for verification."). Below these fields is a section titled "ESTUDIANTES" with a sub-heading "Estudiante: #1". This section contains two input fields: "Matercula:" and "Carrera:" (with a dropdown menu and a plus sign). At the bottom right, there are three buttons: "Guardar y agregar otro", "Guardar y continuar editando", and "GUARDAR".

Ilustración 4.57: Vista administrativa agregar administrador

Fuente: elaboración propia

Modificar administrador

La ilustración 4.58 muestra los datos del usuario para modificar.

The screenshot shows the Django administration interface for modifying a user. The page title is "Administración de Django" and the breadcrumb is "Inicio - Authentication and Authorization - Usuarios - administrador". The main heading is "Modificar usuario" with a "HISTORIA" button. The form includes two input fields: "Nombre de usuario:" (with the value "administrador" and a note: "Obligatorio. Longitud máxima 30 caracteres alfanuméricos (letras, dígitos y @/./+/_/) solamente.") and "Contraseña:" (with the value "algoritmo: pbkdf2_sha256 repeticiones: 24000 salt: k1Fdkq**** hash: b1wgca*****" and a note: "New passwords are not stored, so there is no way to see this user's password, but you can change the password using this form."). Below these fields is a section titled "Información personal" with three input fields: "Nombre:", "Apellido:", and "Email address:". Below this section is a section titled "Permisos" with a checkbox labeled "Active" (checked) and a note: "Marca si el usuario debe ser tratado como un usuario activo. Desactive este campo en lugar de eliminar usuarios."

Ilustración 4.58: Vista administrativa modificar administrador

Fuente: elaboración propia

Eliminar administrador

A eliminar un administrador, se solicita confirmar la acción como muestra la ilustración 4.59.



Ilustración 4.59: Vista administrativa eliminar administrador

Fuente: elaboración propia

Módulo Director de Escuela

La ilustración 4.60 indica los datos que el módulo director de escuela posee.

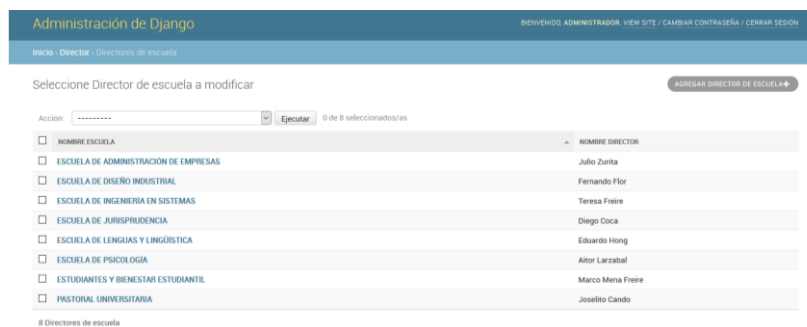


Ilustración 4.60: Vista administrativa módulo Directores de escuela

Fuente: elaboración propia

Agregar director de escuela

Agregar, modificar o eliminar datos de cada módulo funciona de la misma manera que se mostró con el módulo de usuario. En la ilustración 4.61 se puede observar el formulario para añadir un director de escuela.

The screenshot shows the Django administration interface for adding a school director. The page title is "Administración de Django" and the breadcrumb trail is "Inicio > Director > Directores de escuela > Agregar director de escuela". The form includes the following fields:

- Nombre escuela:
- Nombre director:
- Título:
- Cargo:

At the bottom of the form, there are three buttons: "Guardar y agregar otro", "Guardar y continuar editando", and "GUARDAR".

Ilustración 4.61: Vista administrativa agregar director de escuela

Fuente: elaboración propia

Modificar director de escuela

A continuación, se muestra el formulario de modificación de un director de escuela.

The screenshot shows the Django administration interface for modifying a school director. The page title is "Administración de Django" and the breadcrumb trail is "Inicio > Director > Directores de escuela > ESCUELA DE INGENIERIA EN SISTEMAS". The form includes the following fields:

- Nombre escuela: (with a "HISTORIA" button to its right)
- Nombre director:
- Título:
- Cargo:

At the bottom of the form, there are four buttons: "Eliminar" (in red), "Guardar y agregar otro", "Guardar y continuar editando", and "GUARDAR".

Ilustración 4.62: Vista administrativa modificar director de escuela

Fuente: elaboración propia

Eliminar escuela

La aplicación muestra una advertencia al eliminar escuela indicando que se debe eliminar primero las carreras que contiene.

No se puede eliminar Director de escuela

Para eliminar Director de escuela 'ESCUELA DE INGENIERÍA EN SISTEMAS' requiere eliminar los siguientes objetos relacionados protegidos:

- Carrera: Ingeniería en Sistemas

Ilustración 4.63: Vista administrativa eliminar director de escuela

Fuente: elaboración propia

Módulo Carrera

La ilustración 4.64, muestra los datos que el módulo carrera posee.

NOMBRE CARRERA	ESCUELA
<input type="checkbox"/> Diseño Industrial	ESCUELA DE DISEÑO INDUSTRIAL
<input type="checkbox"/> Ingeniería en Contabilidad y Auditoría	ESCUELA DE ADMINISTRACIÓN DE EMPRESAS
<input type="checkbox"/> Ingeniería en Sistemas	ESCUELA DE INGENIERÍA EN SISTEMAS
<input type="checkbox"/> Jurisprudencia	ESCUELA DE JURISPRUDENCIA
<input type="checkbox"/> Licenciatura en Administración de Empresas	ESCUELA DE ADMINISTRACIÓN DE EMPRESAS
<input type="checkbox"/> Psicología Clínica	ESCUELA DE PSICOLOGÍA
<input type="checkbox"/> Psicología Organizacional	ESCUELA DE PSICOLOGÍA

7 carreras

Ilustración 4.64: Vista administrativa módulo carrera

Fuente: elaboración propia

Agregar carrera

En la ilustración 4.65 muestra el formulario para agregar una nueva carrera.

Nombre carrera:

Escuela:

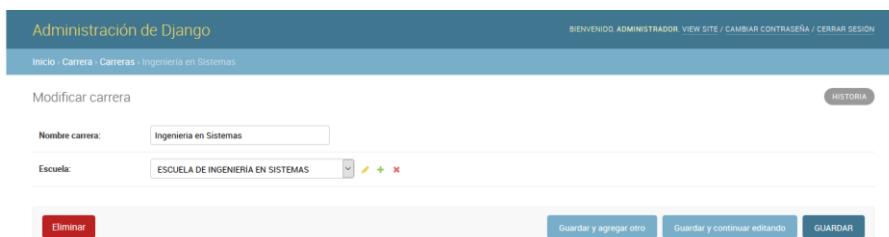
Guardar y agregar otro Guardar y continuar editando GUARDAR

Ilustración 4.65: Vista administrativa agregar carrera

Fuente: elaboración propia

Modificar carrera

La ilustración 4.66 muestra el formulario para modificar una carrera seleccionada.



The screenshot shows the Django administration interface for modifying a career. The page title is "Administración de Django" and the user is logged in as "BIENVENIDO ADMINISTRADOR". The breadcrumb trail is "Inicio > Carrera > Carreras > Ingeniería en Sistemas". The main heading is "Modificar carrera" with a "HISTORIA" button. The form contains two fields: "Nombre carrera:" with the value "Ingeniería en Sistemas" and "Escuela:" with a dropdown menu showing "ESCUELA DE INGENIERIA EN SISTEMAS". At the bottom, there are three buttons: "Eliminar" (red), "Guardar y agregar otro" (blue), "Guardar y continuar editando" (blue), and "GUARDAR" (blue).

Ilustración 4.66: Vista administrativa modificar carrera

Fuente: elaboración propia

Eliminar carrera

Al momento de eliminar una carrera muestra una advertencia con los objetos que se eliminarán.



The screenshot shows the Django administration interface for deleting a career. The page title is "Administración de Django" and the user is logged in as "BIENVENIDO ADMINISTRADOR". The breadcrumb trail is "Inicio > Carrera > Carreras > Ingeniería en Sistemas > Eliminar". The main heading is "¿Está seguro?". Below the heading, there is a warning message: "¿Está seguro de que quiere eliminar la carrera 'Ingeniería en Sistemas'? Se eliminarán los siguientes objetos relacionados:". Under "Summary", there are two items: "Estudiantes: 2" and "Carreras: 1". Under "Objects", there are three items: "Carrera: Ingeniería en Sistemas", "Estudiante: Mario Alejandro Moreno Montero", and "Estudiante: Diego Gabriel Romero Villacis". At the bottom, there are two buttons: "Si estoy seguro" (red) and "No, take me back" (grey).

Ilustración 4.67: Vista administrativa eliminar carrera

Fuente: elaboración propia

Módulo Proceso

Cada solicitud tiene un proceso a seguir y en este módulo es donde se administra ese proceso, en la siguiente ilustración 4.68 muestra los procesos agregados.



Ilustración 4.68: Vista administrativa módulo proceso

Fuente: elaboración propia

Agregar proceso de solicitud

La ilustración 4.69 se puede observar el formulario para añadir un proceso.

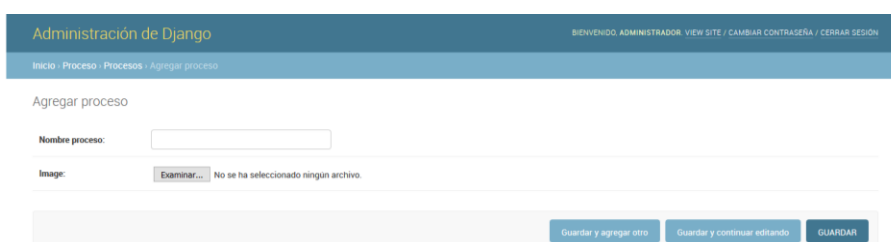


Ilustración 4.69: Vista administrativa agregar proceso

Fuente: elaboración propia

Modificar proceso de solicitud

La ilustración 4.70 indica el formulario de modificación de un proceso.

Ilustración 4.70: Vista administrativa modificar proceso

Fuente: elaboración propia

Eliminar proceso de solicitud

Al eliminar proceso muestra una advertencia indicando que los datos están protegidos, se recomienda modificar el proceso en lugar de eliminarlo.

Ilustración 4.71: Vista administrativa eliminar proceso

Fuente: elaboración propia

Módulo Solicitud

Las solicitudes se ingresan en este módulo, en la ilustración siguiente se muestran las solicitudes ingresadas, con la posibilidad de filtrarlas con un

menú en la parte derecha o buscar una solicitud en específico en el formulario ubicado en la parte superior del listado de solicitudes.

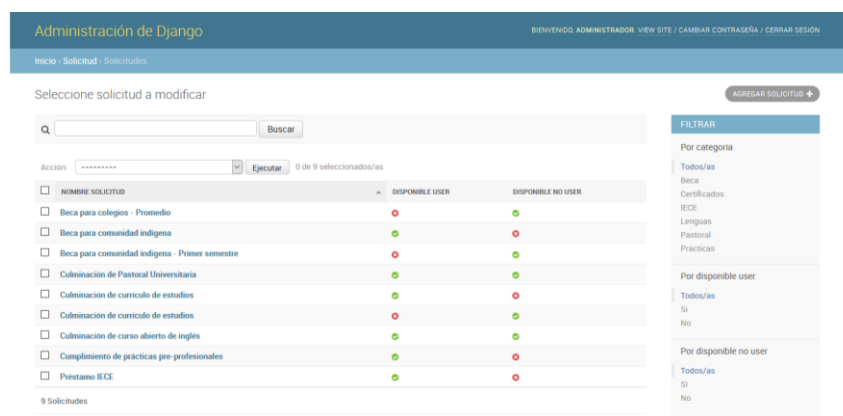


Ilustración 4.72: Vista administrativa módulo solicitudes

Fuente: elaboración propia

Agregar solicitud

La ilustración 4.73 muestra el formulario para agregar una solicitud.

Ilustración 4.73: Vista administrativa agregar solicitud

Fuente: elaboración propia

Modificar solicitud

Al seleccionar una solicitud a modificar muestra un formulario con los datos que contiene.

The screenshot shows the Django administration interface for modifying a request. The page title is "Administración de Django" and the breadcrumb trail is "Inicio > Solicitud > Solicitudes > Cumplimiento de prácticas pre-profesionales". The main heading is "Modificar solicitud" with a "HISTORIA" button. The form contains the following fields:

- Nombre solicitud:** "Cumplimiento de prácticas pre-profesionales"
- Cuerpo solicitud:** A text area containing the text: "Yo, %s %s, con matrícula N° %s, estudiante de la Escuela de %s, solicito se me certifique el cumplimiento de las horas de prácticas pre-profesionales, requisito previo a la obtención del título universitario en la presente carrera."
- Categoría:** "Prácticas"
- Campo adicional:** An empty text field.
- Campo adicional1:** An empty text field.
- Campo adicional2:** An empty text field.
- Campo adicional3:** An empty text field.

At the bottom left, there is a checkbox labeled "Desconectar usuario".

Ilustración 4.74: Vista administrativa modificar solicitud

Fuente: elaboración propia

Eliminar solicitud

La ilustración 4.75 muestra el mensaje de advertencia si se decide eliminar una solicitud.

The screenshot shows the Django administration interface for deleting a request. The page title is "Administración de Django" and the breadcrumb trail is "Inicio > Solicitud > Solicitudes > Cumplimiento de prácticas pre-profesionales > Eliminar". The main heading is "¿Está seguro?". Below the heading, there is a warning message: "¿Está seguro de que quiere eliminar los solicitud 'Cumplimiento de prácticas pre-profesionales'? Se eliminarán los siguientes objetos relacionados:". The "Summary" section shows "Solicitudes: 1". The "Objects" section shows "Solicitud: Cumplimiento de prácticas pre-profesionales". At the bottom, there are two buttons: "Sí, estoy seguro" (highlighted in red) and "No, tákeme back".

Ilustración 4.75: Vista administrativa eliminar solicitud

Fuente: elaboración propia

4.2 Validación de resultados

Para la validación de resultados, se realizó una encuesta de satisfacción a los estudiantes de la universidad, en su mayoría los mismos estudiantes que fueron de ayuda en el *focus group*, de ellos se obtuvo los requerimientos funcionales de la aplicación además se realizó una entrevista a la Ingeniera Teresa Freire directora de la Escuela de Sistemas para evaluar la aplicación desde el punto de vista administrativo.

4.2.1 Encuesta de satisfacción

Los resultados de las preguntas realizadas se muestran seguidos de sus gráficos.

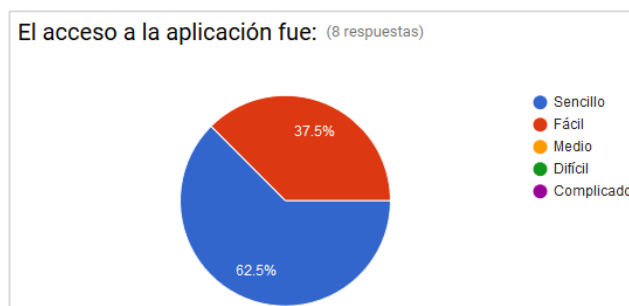


Ilustración 4.76: Resultados pregunta 1

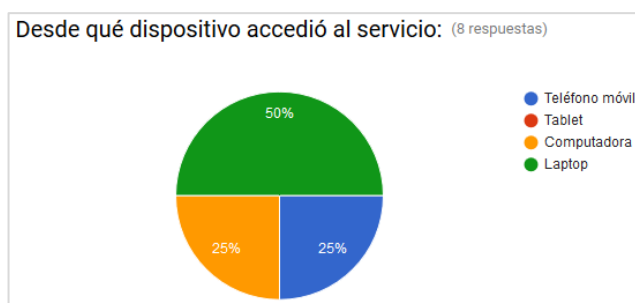


Ilustración 4.77: Resultados pregunta 2



Ilustración 4.78: Resultados pregunta 3



Ilustración 4.79: Resultados pregunta 4

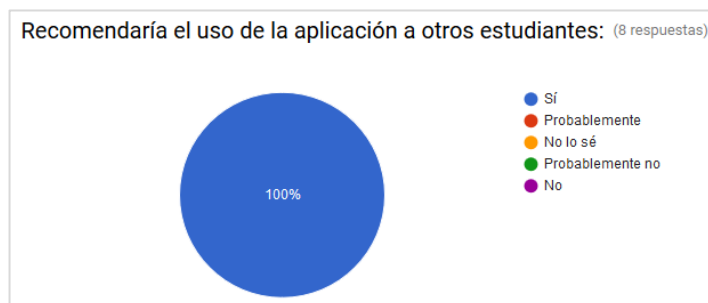


Ilustración 4.80: Resultados pregunta 5

En la última pregunta "¿Qué implementaría usted para mejorar la aplicación?" no se habló nada relacionado al diseño o funcionalidad, en cambio las personas que respondieron coincidieron en que se realice tutoriales de uso de la aplicación, uno de ellos mencionó específicamente un vídeo tutorial.

4.2.1 Entrevista a usuario administrador

1. ¿Cómo describiría el acceso a la aplicación?

La entrevistada describió como fácil al acceso a la parte administrativa recalcando que únicamente se necesita el nombre de usuario y contraseña.

2. ¿Recomendaría el uso de la aplicación a otros estudiantes?

La respuesta fue afirmativa mencionando que la aplicación es de mucha ayuda por los problemas que los estudiantes poseen, entre los más comunes como la redacción y el desconocimiento de parámetros que debe contener una solicitud, de esta manera con todo parametrizado no se desperdiciarían recursos en caso de errores.

3. ¿La modificación o ingreso de nuevas solicitudes se lo realiza de forma simple?

Se mencionó que la interfaz de la aplicación es fácil de usar, lo cual responde también la siguiente pregunta de la entrevista sobre la usabilidad, también se indica que el ingreso o modificación de nuevas solicitudes es sencillo, únicamente se debe establecer los parámetros que una solicitud contiene.

4. ¿La interfaz administrativa fue fácil de usar?

Se expresa la entrevistada sobre la facilidad de uso en la anterior pregunta indicando que fue fácil.

5. ¿Cree que el *framework* de desarrollo utilizado de Django, fue la mejor opción para el desarrollo del proyecto?

Se recalcaron las facilidades que posee, que minimiza el esfuerzo del desarrollador, la parte administrativa es robusta, las seguridades que contiene, los controles inmersos dentro de la aplicación donde el desarrollador se encarga de utilizarlos y optimizarlos y el uso de librerías instalables que ayudaron a resolver el problema planteado.

5. ¿Cree que las seguridades que incluye Django como protección contra XSS, CSRF, SQL injection son de aporte para el sitio?

Se destaca la herramienta utilizada comparando con otros frameworks de desarrollo donde las seguridades deberían ser implantadas por los mismos desarrolladores.

6. ¿La metodología utilizada para el desarrollo del sitio XP fue adecuada?

A ser una metodología ágil se resalta que fue la adecuada al ser una metodología utilizada en aplicaciones con cierto grado de complejidad donde

el tiempo de desarrollo se requiere que sea menor por lo cual se dice que fue correcta la elección de la misma.

7. ¿Qué tan satisfecho está con el funcionamiento de la aplicación?

Por parte de la aplicación se encuentra muy satisfecha la entrevistada mencionando que faltó un poco de colaboración del área de TI de la universidad, donde este tipo de iniciativas deben darse cabida para ponerse en funcionamiento a servicio de los usuarios independientemente que la tecnología utilizada sea la que el departamento de TI espera.

8. ¿Qué agregaría a la aplicación como una característica futura?

Como característica futura se indicó que sería de ayuda agregar tipos de trámites o informes realizados por docentes.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Como antecedente para el inicio del desarrollo, se fundamentó de manera teórica y metodológica sobre el desarrollo web lo cual permitió tener un mejor entendimiento de los recursos utilizados durante la construcción de la aplicación.
- Mediante la realización de entrevistas y *focus group* se logró tener un diagnóstico sobre la manera en las solicitudes son realizadas por estudiantes de la universidad y todos los problemas que tienen los mismos para realizarlas.
- El *framework* de desarrollo de aplicaciones Django encajó perfectamente con la metodología usada Programación Extrema, ya que desde su concepción brinda un tipo de arquitectura diferente que es Modelo, Plantilla y Vista (MTV, por sus siglas en inglés) y la configuración de las mismas se inclina en un desarrollo ágil.
- Django además de aportar con un desarrollo ágil y al ser código abierto permite a más usuarios a crear aplicaciones genéricas o librerías que pueden ser instaladas en cualquier proyecto de desarrollo hecho con este *framework*, es el caso de *django-filter* que se la puede encontrar en GitHub

(<https://github.com/carltongibson/django-filter>) y que fue instalado en esta aplicación para filtrar por categorías las solicitudes.

- Para el desarrollo y perfeccionamiento del presente proyecto, la metodología Programación Extrema expuso ser verdaderamente una metodología ágil, la misma permitió pulir la aplicación mediante pruebas en cada avance del proyecto.

RECOMENDACIONES

- Motivar a los estudiantes de la carrera de Sistemas a investigar sobre nuevas tecnologías de desarrollo que pueden ser soluciones factibles de acuerdo a distintos tipos de programas que se tengan que realizar.
- La documentación de cualquier tecnología es muy importante tenerla presente durante todo el tiempo de desarrollo, éstas suelen ser muy completas y ayudan a solucionar cualquier tipo de problema existente al elaborar cualquier aplicación.
- Promover el uso de Git y GitHub en los estudiantes, el primero para controlar las versiones en sus desarrollos y en el caso del segundo, conocer esta plataforma ya que además de ser un tipo de red social de programadores, se puede convertir en un portafolio y hacer crecer su marca personal. Es un recurso muy utilizado por grandes empresas y administradores de proyectos.
- “No reinventes la rueda” al comenzar a utilizar GitHub se tiene acceso a gran cantidad de código abierto que puede ser utilizado libremente en cualquier tipo de desarrollo.
- Para el uso de la aplicación es importante que el Administrador tenga actualizado cada módulo y al ingresar las solicitudes hacerlo

correctamente para que no se presenten errores al usuario final al momento de generarlas.

- Cuando el programador esté desarrollando alguna aplicación utilizando esta metodología, es recomendable hacerlo a manera de prototipos controlando la funcionalidad principal de cualquier sección así, de esta manera, evitar problemas posteriores.

BIBLIOGRAFÍA

- Androulakis, S. (2010). *Python hosted*. Obtenido de MyTARDIS: <https://pythonhosted.org/MyTARDIS/architecture.html>
- Aubry, C. (2014). *HTML5 y CSS3: para sitios con diseño web responsive*. Barcelona: Ediciones ENI.
- Barzana, R. (30 de 12 de 2006). *Apuntes. Ingeniería del software. Sistemas Informaticos. Nivel de madurez software. Informatica Aplicada a la Gestión Pública. 2005/06-2*. Obtenido de Universidad de Murcia: <http://www.um.es/docencia/barzana/LAGP/lagp2.html>
- Bassi, M. (2013). *Python en 8 clases: Aprendiendo a programar con Python*. Genes Digitales Publishing.
- Beck, K. (2000). *Extreme programming explained: embrace change*. Addison-Wesley Professional.
- Bootstrap Core Team. (s.f.). *Bootstrap*. Obtenido de <http://getbootstrap.com/>
- Caivano, R. M., & Villoria, L. N. (2009). *Aplicaciones Web 2.0*. Villa María: Eduvim.
- Canós, J., Letelier, P., & Penadés, M. (2003). *Metodologías ágiles en el desarrollo de software*. Valencia: Universidad Politécnica de Valencia.
- Carvajal, M., & Saab, J. (23 de Agosto de 2010). *Gobierno de Colombia*. Obtenido de Lineamientos y metodologías en usabilidad para Gobierno en línea: http://estrategia.gobiernoenlinea.gov.co/623/articles-8237_guia_usabilidad.pdf
- Coronel, C. (2011). *Bases de Datos, Diseño, Implementacion y Administracion*. México D. F.: Cengage Learning Editores.
- Cuello, J., & Vittone, J. (2013). *Diseñando apps para móviles*.
- De Luca, D. (2011). *HTML5: entienda el cambio, aproveche su potencial*. Buenos Aires: Fox Andina S. A.

Django Software Foundation. (s.f.). *Django*. Obtenido de Overview: <https://www.djangoproject.com/start/overview/>

Fantoni, Á. L. (2015). *Comunicación total 5a Ed.* Madrid: ESIC.

Forcier, J., Bissex, P., & Chun, W. (2008). *Python Web Development with Django*. Boston: Pearson Education.

Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript*. Barcelona: Marcombo.

Holovaty, A., & Kaplan-Moss, J. (2007). *El libro de Django*. Apress.

Honrubia López, F. J., Álvarez Garrote, H., Pérez Marín, S., & Bretón, A. (2013). *Programación de aplicaciones para Iphone y Ipad*. España: Ministerio de Educación, Cultura y Deporte.

Hurtado, I., & Toro, J. (2007). *Paradigmas y métodos de investigación en tiempos de cambio*. Caracas: Editorial CEC, SA.

Infante Montero, S. (30 de Abril de 2012). *Maestros del web*. Obtenido de <http://www.maestrosdelweb.com/curso-django-entendiendo-como-trabaja-django/>

JetBrains. (2016). Obtenido de Free for students: Professional developer tools from JetBrains: <https://www.jetbrains.com/student/>

JetBrains. (2016). Obtenido de PyCharm: <https://www.jetbrains.com/pycharm/>

Llopis, R. (2004). *Grupos de discusión*. Madrid: ESIC.

Lutz, M. (2013). *Learning Python*. Estados Unidos: O'Reilly Media, Inc.

Maymala, J. (2015). *PostgreSQL for Data Architects*. Mumbai: Packt Publishing Ltd.

Mendoza, M. (2004). *Metodologías De Desarrollo De Software*.

- Nixon, R. (2014). *Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites*. Estados Unidos: O'Reilly Media, Inc.
- Obe, R., & Hsu, L. S. (2012). *PostgreSQL: Up and Running*. California: O'Reilly Media, Inc.
- Oliver González, A., Moré López, J., & Climent Roca, S. (2011). *Traducción y tecnologías*. Barcelona: Editorial UOC.
- Ordax Cassá , J. M., & Ocaña Díaz Ufano, P. A. (2012). *Programación web en Java*. España: Ministerio de Educación.
- Pinkham, A. (2015). *Django Unleashed*. Estados Unidos: Sams Publishing.
- Pressman, R. (2010). *Ingeniería del software*. México D. F.: McGraw Hill .
- PUCESA. (2011). *Misión*. Obtenido de Pontificia Universidad Católica del Ecuador Sede Ambato: <http://pucesa.edu.ec>
- Real Academia Española. (2016). *Diccionario de la lengua española*. Obtenido de Real Academia Española: <http://dle.rae.es/>
- Rodríguez, E. (2005). *Metodología de la Investigación*. Univ. J. Autónoma de Tabasco.
- Sánchez Maza, M. Á. (2012). *JavaScript*. Málaga: Innovación Y Cualificación.
- Sawyer, D. (2012). *CSS: The Missing Manual 2a Ed*. Estados Unidos: O'Reilly Media, Inc.
- Sommerville, I. (2005). *Ingeniería del software*. Madrid: Pearson Educación.
- Villar Varela, A., & Fernández Pérez, C. (2010). *Base de Datos Access*. Vigo: Ideaspropias Editorial S.L.

ANEXOS

Anexo 1: Entrevista

1. ¿Conoce algún mecanismo en la universidad que permita realizar solicitudes académicas?
2. ¿Qué se puede decir que le falta a dicho sistema?
3. ¿Es alto el número de solicitudes que los estudiantes entregan normalmente durante el semestre?
4. ¿Los estudiantes se acercan con frecuencia a solicitar algún modelo de solicitud?
5. ¿Cree que una aplicación web agilizaría el proceso de realizar solicitudes a los estudiantes?
6. ¿Qué funcionalidades cree usted que debería tener la aplicación para satisfacer las necesidades del estudiante?
7. ¿Qué solicitudes son las que más se realizan?

Anexo 2: Focus group

1. ¿Ha realizado alguna vez en su etapa universitaria solicitudes académicas?
2. ¿Cuál ha sido el proceso que ha seguido para realizarlas?
3. ¿Conoce sobre los modelos de solicitudes que ofrece la secretaría de la escuela?
4. ¿Conoce algún mecanismo en la universidad que permita realizar con facilidad dichas solicitudes? ¿Cuéntanos la experiencia?
5. ¿Qué tan familiarizados están en utilizar en EVAE o Académicas desde el móvil?
6. ¿Cree que una aplicación web facilitaría la creación de solicitudes y ahorraría tiempo a los estudiantes?
7. ¿Qué diseño les gusta más al abrirlo desde el móvil? (EVAE - Academics)
8. ¿Cree que la aplicación web debe visualizarse correctamente en teléfonos móviles?
9. ¿Qué funcionalidades debería tener la aplicación para satisfacer las necesidades de los estudiantes?
10. ¿Cree que la aplicación deba mostrar el proceso que sigue su solicitud al ser entregada?

Anexo 3: Firmas de personas entrevistadas

Anexo 3: Firmas de personas entrevistadas

A continuación, se presenta la información perteneciente a personas han participado en la entrevista y focus - group la cual se la puede utilizar para validar su participación en el presente proyecto.

Entrevista

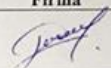
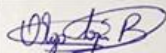
Entrevista: Aplicación web de solicitudes académicas	
Entrevistado	Firma
Hywem Hoyoga	
Olga López	

Tabla: Firmas de entrevista

Focus - group






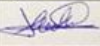
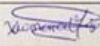

Focus-group: Aplicación web de solicitudes académicas			
Nombre	Escuela	Firma	Correo
Mario Ruiz	Sistemas		mario.ruiz.1995@hotmail.com
Mario Pérez	Sistemas		mfp1999@gmail.com
David Mora	Sistemas		davidmora@gmail.com
Gabriel Romero	Sistemas		g@contadigital.com
Andrés Robalino	Administración		andres.robalino@hotmail.com
Alex Mera	Contabilidad		damián.mera18@gmail.com
Xiomara Mejías	contabilidad		xiomara.mejias@hotmail.com
Katherine Briza	Administración		kathylep@gmail.com

Tabla: Firmas de focus-group

Ilustración: Firma de personas entrevistadas

Anexo 4: Encuesta de satisfacción

El presente formulario es una encuesta de satisfacción de la aplicación SolicitudesApp.

Nombre _____

Carrera _____

El acceso a la aplicación fue:

Sencillo	Fácil	Medio	Difícil	Complicado

Desde qué dispositivo accedió al servicio:

Teléfono móvil	Tablet	Computadora	Laptop

¿La interfaz de la aplicación fue fácil de usar?

Sí	De dificultad media	No

Qué tan satisfecho está con el funcionamiento de la aplicación:

Muy satisfecho	Satisfecho	Normal	Insatisfecho	Terriblemente insatisfecho

Recomendaría el uso de la aplicación a otros estudiantes:

Sí	Probablemente	No lo sé	Probablemente no	No

Qué implementaría usted para mejorar la aplicación:

Firma:

Anexo 5: Entrevista a usuario administrador

1. ¿Cómo describiría el acceso a la aplicación?
2. ¿Recomendaría el uso de la aplicación a otros estudiantes?
3. ¿La modificación o ingreso de nuevas solicitudes se lo realiza de forma simple?
4. ¿La interfaz administrativa fue fácil de usar?
5. ¿Cree que el framework de desarrollo utilizado de Django, fue la mejor opción para el desarrollo del proyecto?
6. ¿Cree que las seguridades que incluye Django como protección contra XSS, CSRF, SQL injection son de aporte para el sitio?
7. ¿La metodología utilizada para el desarrollo del sitio XP fue adecuada?
8. ¿Qué tan satisfecho está con el funcionamiento de la aplicación?
9. ¿Qué agregaría a la aplicación como una característica futura?

Anexo 6: Manual de despliegue

Crear el servidor

Dentro de Amazon Web Services (AWS):

1. Acceder a EC2
2. Launch Instance
3. Seleccionar Ubuntu
4. Elegir t2.micro y dejar la configuración por defecto en los dos pasos siguientes
5. Crear un grupo de seguridad con los puertos 22, 80 y 8000 abiertos desde cualquier IP.
6. Seleccionar Launch y crear y descargar las llaves ssh.

Conectarse al servidor

Se utiliza la llave descargada, en una terminal se abre su ubicación.

1. Poner la llave en modo lectura > `chmod 0400 NombreLlave.pem`
2. Conectarse al servidor mediante la IP > `sudo ssh -i Platzi.pem ubuntu@IP`

Configuración inicial del servidor

1. Como buena práctica se actualiza el servidor > `sudo apt-get update > sudo apt-get upgrade.`
2. Crear un nuevo usuario con capacidad de correr comandos de súper usuario > `sudo useradd nombreusuario -g sudo -m`
3. Asignar contraseña > `sudo passwd nombreusuario`
4. Iniciar sesión > `su nombreusuario`

Instalar dependencias

1. Python > `sudo apt-get install python-pip python-dev`
2. PostgreSQL > `sudo apt-get install postgresql postgresql-contrib libpq-dev`
3. Git > `sudo apt-get install git`
4. Nginx > `sudo apt-get install nginx`

Configuración de PostgreSQL

1. Iniciar sesión > `sudo su postgres`
2. Ingresar al shell > `psql`
3. Crear base de datos > `CREATE DATABASE nombrebase;`
4. Crear usuario para base de datos > `CREATE USER nombreusuario WITH PASSWORD 'contraseña';`
5. Asignar permisos sobre la base al usuario > `GRANT ALL PRIVILEGES ON DATABASE nombrebase TO nombreusuario;`

6. Salir de la shell > \q
7. Salir de postgres > exit

Configuración del proyecto

Para mayor facilidad se utilizó Github como repositorio.

1. Clonar el proyecto
> git clone <https://github.com/atripleme/SolicitudesApp-Django.git> app
2. Instalar virtualenv > pip install virtualenv
3. Crear el entorno virtual > virtualenv -p \$(which python) .venv
4. Activar entorno > source .venv/bin/actívate
5. Instalar dependencias de Pillow > sudo apt-get install libjpeg-dev
6. Ir a carpeta del proyecto > cd app
7. Instalar requerimientos > pip install -r requirements.txt
8. Editar settings.py DEBUG = False y ALLOWED_HOSTS = ['.eispucesa.com']

Primera revisión

El proyecto debe ser capaz de utilizar su base de datos y servirse usando el servidor de desarrollo gunicorn.

1. Hacer las migraciones > python manage.py makemigrations > python manage.py migrate
2. Crear súper usuario > python manage.py createsuperuser
3. Correr servidor de desarrollo > python manage.py runserver 0.0.0.0:8000
4. Correr gunicorn > gunicorn SolicitudesApp.wsgi -b 0.0.0.0:8000

Si todo funcionó sin problema el sitio se podrá ver en la IP o url en el puerto 8000.

Configurar Nginx

1. Iniciar sesión como súper usuario > sudo su -
2. Ir al directorio de Nginx > cd /etc/nginx/
3. Borrar los archivos de configuración > rm sites-*/default
4. Crear un script con el [contenido](#) > vim sites-available/app
5. Enlazar los archivos > ln -s /etc/nginx/sites-available/app /etc/nginx/sites-enabled/
6. Reiniciar Nginx > service nginx restart

Configurar gunicorn

1. Salir de la sesión de súper usuario > exit
2. Crear la carpeta de scripts y logs > mkdir deploy logs
3. Crear un script dentro de deploy con el [contenido](#) > vim deploy/gunicorn_start
4. Hacer ejecutable el script > chmod +x deploy/gunicorn_start
5. Probar script > ./deploy/gunicorn_start

Supervisor

1. Instalar supervisor > sudo aptitude install supervisor
2. Iniciar sesión como súper usuario > sudo su –
3. Ingresar a la carpeta de archivos de configuración > cd /etc/supervisor/conf.d
4. Crear un script con el [contenido](#) > nano solicitudesapp.conf
5. Crear el archivo de logs > touch /home/mario/logs/gunicorn_supervisor.log
6. Pedir a supervisor leer la configuración > sudo supervisorctl reread
7. Añadir al grupo la configuración > sudo supervisorctl update
8. Verificar status, también se puede parar, comenzar y resetear con stop, start y restart respectivamente > sudo supervisorctl status solicitudesapp

La aplicación iniciará automáticamente luego de un reinicio del sistema si tiene algún problema.

Archivos estáticos

Finalmente hay que situarse en la carpeta de la aplicación y ejecutar > python manage.py collectstatic

Anexo 7: Manual de usuario

Acceso a la aplicación

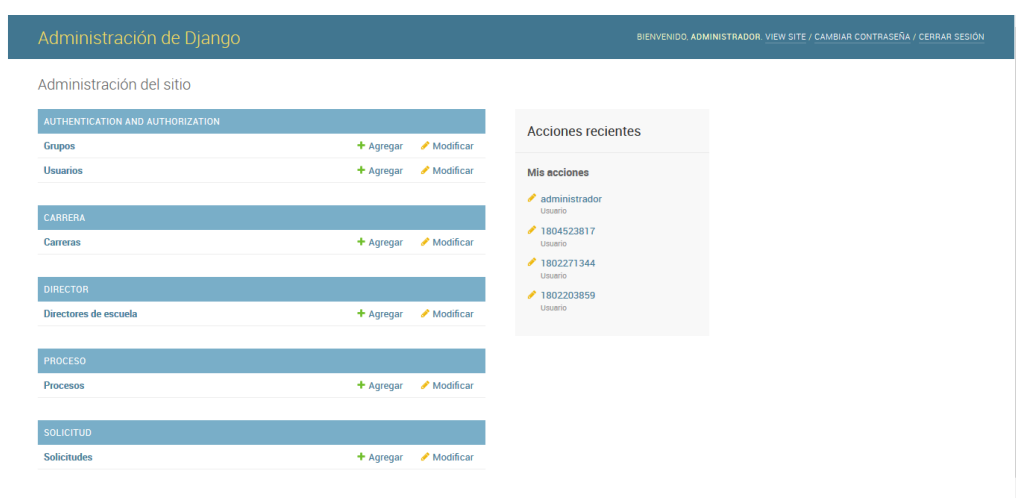
Al ingresar al panel de usuario administrador, muestra el inicio de sesión donde el usuario debe colocar sus credenciales.



The image shows a login form titled "Administración de Django". It contains two input fields: "Nombre de usuario:" and "Contraseña:". Below the fields is a blue button labeled "Identificarse".

Ilustración: Ingreso a la administración

Al iniciar sesión se mostrarán todos los módulos de la aplicación.



The image shows the Django administration dashboard. The header includes "Administración de Django" and "BIENVENIDO, ADMINISTRADOR VIEW SITE / CAMBIAR CONTRASEÑA / CERRAR SESIÓN". The main content is divided into several sections:

- Administración del sitio**
 - AUTHENTICATION AND AUTHORIZATION**
 - Grupos: + Agregar, ✎ Modificar
 - Usuarios: + Agregar, ✎ Modificar
 - CARRERA**
 - Carreras: + Agregar, ✎ Modificar
 - DIRECTOR**
 - Directores de escuela: + Agregar, ✎ Modificar
 - PROCESO**
 - Procesos: + Agregar, ✎ Modificar
 - SOLICITUD**
 - Solicitudes: + Agregar, ✎ Modificar
- Acciones recientes**
 - Mis acciones**
 - administrador Usuario
 - 1804523817 Usuario
 - 1802271344 Usuario
 - 1802203859 Usuario

Ilustración: Módulos de la aplicación

Cambio de contraseña

Para modificar la contraseña, se ingresa al módulo de Usuarios, y se selecciona el usuario administrador que deseamos modificar, la contraseña se muestra cifrada y para cambiarla se ingresa al enlace *“this form”*:

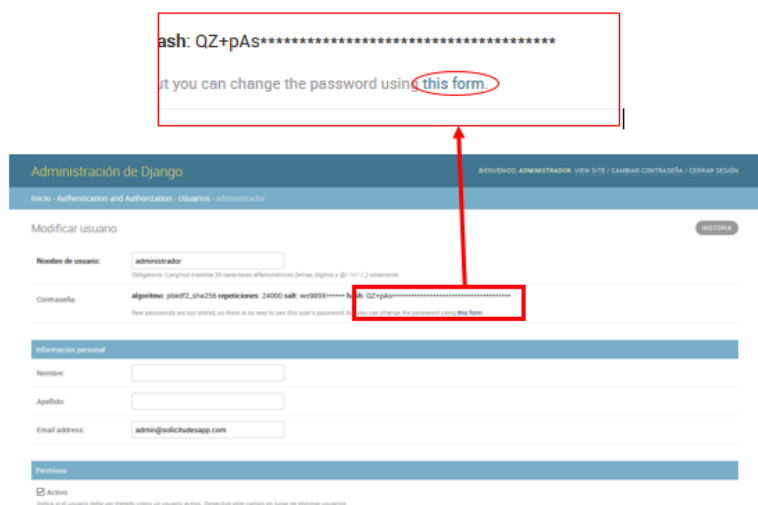


Ilustración: Enlace de cambio de contraseña

Al ingresar en el enlace se muestra el formulario para cambiar la contraseña.

The screenshot shows the Django Admin 'Change password' form for the 'administrador' user. The form includes fields for 'Contraseña:' and 'Contraseña (de nuevo):' with password strength requirements. A 'CAMBIAR CONTRASEÑA' button is at the bottom right.

Ilustración: Formulario de cambio de contraseña

Agregar administrador

Para ingresar un nuevo administrador, el proceso lo hace el administrador principal, se ingresa al módulo de Usuarios y se selecciona en la parte superior, agregar usuario.

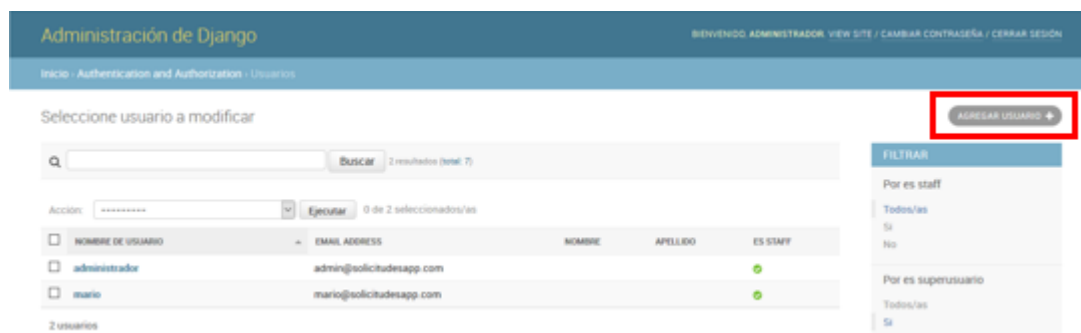


Ilustración: Módulo Usuarios

A continuación, se ingresan los datos en el formulario, no es necesario ingresar datos de la sección estudiante.

The screenshot shows the 'Agregar usuario' form in the Django Admin. The form includes fields for 'Nombre de usuario', 'Contraseña', and 'Confirmación de contraseña'. Below these is a section for 'ESTUDIANTES' with fields for 'Estudiante: #1', 'Matricula', and 'Carrera'. At the bottom, there are three buttons: 'Guardar y agregar otro', 'Guardar y continuar editando', and 'GUARDAR'.

Ilustración: Formulario de nuevo usuario administrador

Luego de ingresar los datos, en el listado de usuarios se selecciona el usuario recientemente creado, se abre un formulario donde se pueden ingresar datos adicionales como Información personal. Se recomienda ingresar un correo en caso del olvidar la contraseña.

The screenshot shows the Django administration interface for user management. The page title is 'Administración de Django' and the user is logged in as 'administrador'. The breadcrumb trail is 'Inicio > Authentication and Authorization > Usuarios > administrador'. The main heading is 'Modificar usuario' with a 'HISTORIA' button. The form contains the following sections:

- Nombre de usuario:** A text input field containing 'administrador'. Below it, a note states: 'Obligatorio. Longitud máxima 30 caracteres alfanuméricos (letras, dígitos y @/./+/-/_ solamente)'. The password algorithm is shown as 'algoritmo: pbkdf2_sha256 repeticiones: 24000 salt: wo989X***** hash: QZ+pAs*****'.
- Contraseña:** A text input field. A note below states: 'Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.'
- Información personal:** A section with three text input fields: 'Nombre:', 'Apellido:', and 'Email address:'. The email address field contains 'admin@solicitudesapp.com'.
- Permisos:** A section with a checked checkbox for 'Activo'. A note below states: 'Indica si el usuario debe ser tratado como un usuario activo. Desactive este campo en lugar de eliminar usuarios.'

Ilustración: Formulario de usuario administrador

Para que el usuario creado tenga permisos administrativos, siguiendo a Información personal se encuentra Permisos y se marcan las opciones de Es staff y Es superusuario.

The screenshot shows the 'Permisos' section of the Django user management interface. It contains three checked checkboxes with the following descriptions:

- Activo**
Indica si el usuario debe ser tratado como un usuario activo. Desactive este campo en lugar de eliminar usuarios.
- Es staff**
Indica si el usuario puede ingresar a este sitio de administración.
- Es superusuario**
Indica que este usuario posee todos los permisos sin que sea necesario asignarle los mismos en forma explícita.

Ilustración: Formulario de usuario administrador

Recuperación de cuenta

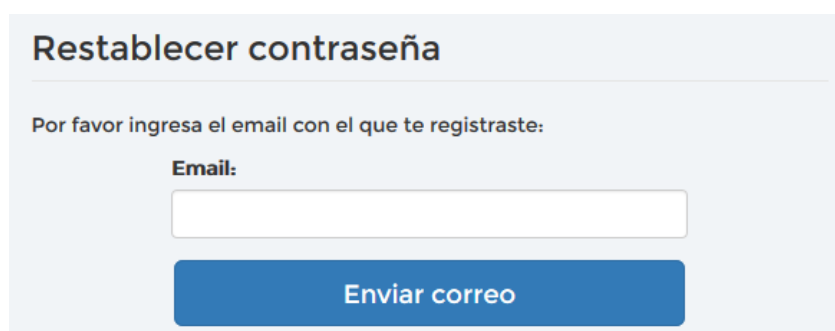
Para recuperar la cuenta del administrador, es necesario ingresar al enlace de olvidé mi contraseña en la vista de usuario normal.



The image shows the login interface for 'SolicitudesApp'. On the left, the app name and tagline 'Realiza tus solicitudes rápidamente!' are displayed. On the right, under the heading 'Bienvenido!', there are input fields for 'Cédula:' and 'Contraseña:'. Below these fields is an 'Ingresar' button. A red box highlights the link '¡Olvidaste tu contraseña?' located below the 'Ingresar' button. Other options include '¿No tienes cuenta?' with a 'Crear cuenta' button, and '¿Estás de visita?' with a 'Soy visitante' button.

Ilustración: Recuperación de contraseña

Es importante que, al crear un usuario administrador, se cree con un email asociado, para así tener un correo a cuál enviar el email de recuperación.



The image shows a form titled 'Restablecer contraseña'. Below the title, it says 'Por favor ingresa el email con el que te registraste:'. There is an input field labeled 'Email:'. Below the input field is a blue button labeled 'Enviar correo'.

Ilustración: Formulario de correo

Se enviará un correo, como muestra la siguiente ilustración:



Ilustración: Correo de recuperación

Se procede a abrir el enlace y establecer una nueva contraseña de acceso al sitio.

The screenshot shows a web form titled "Restablecer contraseña" with the following elements:

- Section:** Tener en cuenta que:
- Rules:**
 - Tu contraseña no puede ser similar a tu información personal.
 - Tu contraseña debe contener mínimo 8 caracteres.
 - Tu contraseña no puede ser de las contraseñas usadas comunmente.
 - Tu contraseña no puede ser totalmente numérica.
- Instruction:** Elige una contraseña nueva para ingresar a la aplicación:
- Fields:**
 - Contraseña nueva:
 - Confirmación de contraseña nueva:
- Button:** Enviar

Ilustración: Formulario de cambio de contraseña

Ingresar, modificar y eliminar datos de cualquier módulo

Todos los módulos funcionan de la misma manera para realizar las funciones principales. El ejemplo a continuación se utilizará el módulo Directores de escuela.

Ingreso de datos

Para ingresar nuevos datos al módulo se lo selecciona, y al hacerlo aparecen los datos que contiene, para agregar un nuevo dato, en la parte superior derecha se encuentra un botón de agregar elementos.

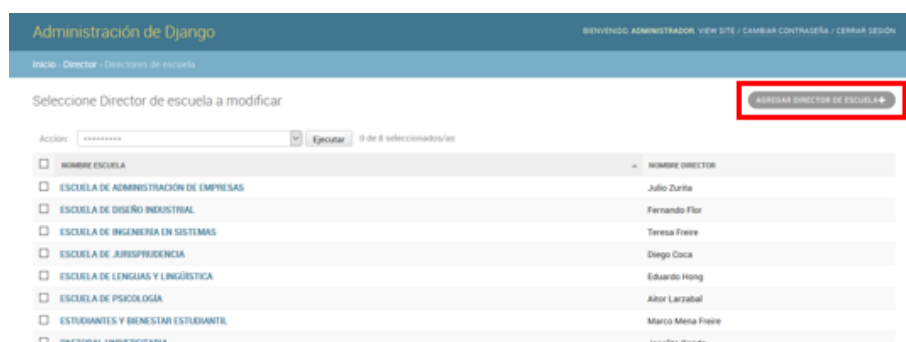


Ilustración: Botón de agregar datos

Cuando se decide en ingresar nuevos datos se abre un formulario, cada uno de acuerdo al módulo seleccionado y se procede a ingresar los elementos que correspondan, en el caso de Directores de escuela se ingresa el nombre de la escuela, nombre del director, el título que posee y se selecciona Director o Directora.

The screenshot shows the Django administration interface for 'Agregar Director de escuela'. The form has the following fields:

- Nombre escuela:
- Nombre director:
- Titulo:
- Cargo: (dropdown menu with options: DIRECTOR, DIRECTOR, DIRECTORA)

At the bottom right, there are three buttons: 'Guardar y agregar otro', 'Guardar y continuar editando', and 'GUARDAR'.

Ilustración: Formulario de ingreso de director

Para modificar o eliminar un elemento de cualquier módulo, se elige el mismo del listado de datos que existen y se muestra un formulario para modificarlo, o también el botón de eliminar. En el caso del módulo de proceso no es factible eliminar el proceso sino modificarlo.

The screenshot shows the Django administration interface for modifying school director data. The page title is "Administración de Django" and the user is logged in as "BIENVENIDO ADMINISTRADOR". The breadcrumb trail is "Inicio > Director > Directores de escuela > ESCUELA DE INGENIERIA EN SISTEMAS". The main heading is "Modificar Director de escuela" with a "HISTORIA" button. The form contains the following fields:

- Nombre escuela: ESCUELA DE INGENIERIA EN SISTEMAS
- Nombre director: Teresa Freire
- Título: Magister
- Cargo: DIRECTORA (dropdown menu)

At the bottom, there are three buttons: "Eliminar" (red), "Guardar y agregar otro", "Guardar y continuar editando", and "GUARDAR".

Ilustración: Modificar o eliminar datos

Ingreso de solicitudes

Para ingresar una solicitud se realiza de igual forma que todos los módulos, pero se debe cumplir ciertas indicaciones, el formulario de ingreso es el siguiente:

The screenshot shows the Django administration interface for adding a request. The page title is "Agregar solicitud". The form contains the following fields:

- Nombre solicitud: [text input]
- Cuerpo solicitud: [text area]
- Categoría: [text input]
- Campo adicional: [text input]
- Campo adicional1: [text input]
- Campo adicional2: [text input]
- Campo adicional3: [text input]
- Disponible user
- Disponible no user
- Destinatario: SECRETARÍA DE ESCUELA (dropdown menu)
- Proceso: [dropdown menu]

Ilustración: Ingreso de solicitud

Nombre de la solicitud

Se ingresa únicamente el nombre que tiene la solicitud y se indica si requiere hoja membretada para su presentación

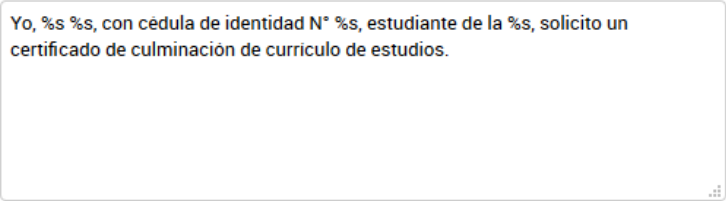
Cuerpo de la solicitud

En el cuerpo de la solicitud, donde debe ir obligatoriamente los nombres, apellidos, cédula o matrícula y escuela o carrera a la que pertenece el estudiante en ese orden, debe estar el símbolo de impresión (%s) que se sustituirá por cada dato mencionado, a continuación, se muestra varios ejemplos:

Cuerpo para usuarios visitantes

Se ubica cédula de identidad.

Cuerpo solicitud:



Yo, %s %s, con cédula de identidad N° %s, estudiante de la %s, solicito un certificado de culminación de currículo de estudios.

Ilustración: Ingreso de cuerpo de solicitud

Cuerpo para usuarios registrados

En lugar de cédula se ubica la palabra matrícula para no perder el contexto.

Cuerpo solicitud:

Yo, %s %s, con matricula N° %s, estudiante de la %s, solicito un certificado de culminación de currículo de estudios.

Ilustración: Ingreso de cuerpo de solicitud

Ingreso de categoría

Se selecciona una categoría a la que pertenece la solicitud, puede ser beca, certificaciones, etc. Cabe mencionar que si se agrega una solicitud de inglés o pastoral obligatoriamente debe tener de nombre Lenguas y Pastoral respectivamente.

Campos adicionales

Cada solicitud puede tener hasta cuatro campos adicionales, pero al ingresarlos quiere decir que por cada uno se aumentará un símbolo de impresión (%s) en el cuerpo de la solicitud donde corresponda. En el campo adicional se debe ingresar lo que el usuario que realiza la solicitud debe ubicar, ejemplo:

Cuerpo solicitud:

Yo, %s %s, con matricula N° %s, me permito solicitar se me tome en cuenta para la BECA por pertenecer a COMUNIDAD INDÍGENA para la carrera de %s, en el %s semestre.

Me permito informarle que he leído el Reglamento de Becas de la PUCESA y estoy de acuerdo con sus lineamientos

Categoría:

Beca

Campo adicional:

Semestre (Ej. Primer, segundo, etc.)

Ilustración: Solicitud con un campo adicional

En el ejemplo se puede observar que aparte de los cuatro elementos de impresión básicos de datos personales, se encuentra un campo adicional. También se muestra la etiqueta `
` que sirve para dar un salto de línea si se desea.

A continuación, se muestra otro tipo de solicitud con dos campos adicionales y la ubicación de los mismo con los símbolos de impresión.

Cuerpo solicitud:	Yo, %s %s, con cédula de identidad N° %s, me permito solicitar se me tome en cuenta para el proceso de BECA POR EXCELENCIA para el primer semestre de la carrera de %s en el periodo %s, por tener un promedio de %s. Habiendo leído el reglamento general de becas, estoy de acuerdo con lo que ahí expone para la carrera que estoy solicitando la beca.
Categoría:	Beca
Campo adicional:	Periodo (Ej. Septiembre - Febrero 2016)
Campo adicional1:	Nota promedio de grado

Ilustración: Solicitud con dos campos adicionales

Nombre de carrera o maestría en la solicitud

Para que una solicitud en lugar del nombre de la escuela, tenga el nombre de la carrera a la que pertenece el estudiante, en el cuerpo debe contener las palabras “carrera de” o “maestrante de la” seguido del símbolo de impresión. Esto hará que se ubique el nombre de la carrera o maestría en lugar de la escuela a la que pertenece el alumno, ejemplo:

Cuerpo solicitud:

Yo, %s %s, con cédula de identidad N° %s, me permito solicitar se me tome en cuenta para el proceso de BECA POR EXCELENCIA para el primer semestre de la carrera de %s en el periodo %s, por tener un promedio de %s.

 Habiendo leído el reglamento general de becas, estoy de acuerdo con lo que ahí expone para la carrera que estoy solicitando la beca.

Ilustración: Solicitud con la carrera del estudiante

Disponibilidad de solicitudes

Para identificar las solicitudes disponibles para los usuarios registrados y usuarios visitantes se marcan las casillas correspondientes dependiendo a quien se encuentre disponible la solicitud.

Disponible user

Disponible no user

Ilustración: Disponibilidad de solicitudes

Destinatario

Para identificar un destinatario se selecciona si va dirigida a la Dirección de escuela o a Dirección de estudiantes.

Destinatario:

Proceso:

Ilustración: Selección de destinatario

Proceso

Finalmente, de los procesos ingresados en el módulo procesos, se selecciona uno para la solicitud.

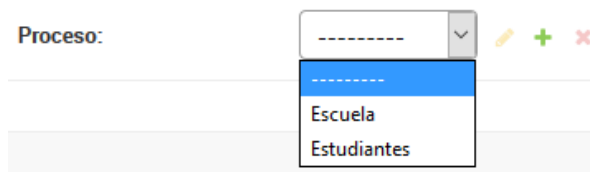


Ilustración: Selección de proceso