



UNIVERSIDAD SURCOLOMBIANA GESTIÓN DE BIBLIOTECAS

CARTA DE AUTORIZACIÓN



CÓDIGO	AP-BIB-FO-06	VERSIÓN	1	VIGENCIA	2014	PÁGINA	1 de 2
--------	--------------	---------	---	----------	------	--------	--------

Neiva, 17 de marzo de 2022

Señores

CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD SURCOLOMBIANA

Ciudad

El (Los) suscrito(s):

Fernando Andrés Andrade López _____, con C.C. No. 7.731.804 _____,
_____, con C.C. No. _____,
_____, con C.C. No. _____,
_____, con C.C. No. _____,

Autor(es) de la tesis y/o trabajo de grado o _____

titulado **Portal web dinámico para el programa de Ingeniería de Software de la Universidad**

Surcolombiana _____

presentado y aprobado en el año **2022** como requisito para optar al título de

Ingeniero de Software _____;

Autorizo (amos) al CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN de la Universidad Surcolombiana para que, con fines académicos, muestre al país y el exterior la producción intelectual de la Universidad Surcolombiana, a través de la visibilidad de su contenido de la siguiente manera:

- Los usuarios puedan consultar el contenido de este trabajo de grado en los sitios web que administra la Universidad, en bases de datos, repositorio digital, catálogos y en otros sitios web, redes y sistemas de información nacionales e internacionales “open access” y en las redes de información con las cuales tenga convenio la Institución.
- Permite la consulta, la reproducción y préstamo a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato Cd-Rom o digital desde internet, intranet, etc., y en general para cualquier formato conocido o por conocer, dentro de los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia.
- Continúo conservando los correspondientes derechos sin modificación o restricción alguna; puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación del derecho de autor y sus conexos.

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



UNIVERSIDAD SURCOLOMBIANA
GESTIÓN DE BIBLIOTECAS

CARTA DE AUTORIZACIÓN



CÓDIGO	AP-BIB-FO-06	VERSIÓN	1	VIGENCIA	2014	PÁGINA	2 de 2
--------	--------------	---------	---	----------	------	--------	--------

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, "Los derechos morales sobre el trabajo son propiedad de los autores" , los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

EL AUTOR/ESTUDIANTE:

Firma: 

EL AUTOR/ESTUDIANTE:

Firma: _____

EL AUTOR/ESTUDIANTE:

Firma: _____

EL AUTOR/ESTUDIANTE:

Firma: _____

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO



CÓDIGO AP-BIB-FO-07 VERSIÓN 1 VIGENCIA 2014 PÁGINA 1 de 3

TÍTULO COMPLETO DEL TRABAJO: Portal web dinámico para el programa de Ingeniería de Software de la Universidad Surcolombiana

AUTOR O AUTORES:

Primero y Segundo Apellido	Primero y Segundo Nombre
Andrade López	Fernando Andrés

DIRECTOR Y CODIRECTOR TESIS:

Primero y Segundo Apellido	Primero y Segundo Nombre
Rojas Rojas	Fernando

ASESOR (ES):

Primero y Segundo Apellido	Primero y Segundo Nombre
Rojas Rojas	Fernando

PARA OPTAR AL TÍTULO DE: Ingeniero de Software

FACULTAD: Ingeniería

PROGRAMA O POSGRADO: Software

CIUDAD: Neiva

AÑO DE PRESENTACIÓN: 2022

NÚMERO DE PÁGINAS: 83

TIPO DE ILUSTRACIONES (Marcar con una X):

Diagramas Fotografías Grabaciones en discos Ilustraciones en general Grabados Láminas
Litografías Mapas Música impresa Planos Retratos Sin ilustraciones Tablas o Cuadros



DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO

CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	2 de 3
--------	--------------	---------	---	----------	------	--------	--------



SOFTWARE requerido y/o especializado para la lectura del documento:

MATERIAL ANEXO:

PREMIO O DISTINCIÓN (En caso de ser LAUREADAS o Meritoria):

PALABRAS CLAVES EN ESPAÑOL E INGLÉS:

<u>Español</u>	<u>Inglés</u>
1. Spring Boot	1. Spring Boot
2. Facade	2. Facade
3. MVC	3. MVC
4. Diseño	4. Design
5. Marca	5. Brand

RESUMEN DEL CONTENIDO: (Máximo 250 palabras)

El propósito de este proyecto es realizar la página web del programa de Ingeniería de Software mediante la aplicación de diseños vanguardistas que logren estar a la par con las tendencias de diseño actuales.

Para la consecución de este proyecto se estudió la teoría de diseño que ha marcado tendencia en los últimos tres años para lograr partir de una base sólida de estilo. Luego de analizar los conceptos presentes se eligió utilizar una plantilla HTML 5 con Bootstrap 5 para reducir el tiempo de desarrollo y poder contar con un formato realizado por un diseñador experto.

Una vez establecido el diseño artístico fue igualmente importante seleccionar una tecnología adecuada que soporte todos los elementos visuales. Para esta tarea se eligió el marco de desarrollo Spring Boot, el cual es una variante simplificada de Spring Framework, la cual permite generar soluciones basadas en Java EE.

La selección de la tecnología Spring no solo se basó en su robustez garantizada tras años en el mercado, sino también, al hecho de que la Universidad Surcolombiana apoya su infraestructura en la misma tecnología, lo cual permite que este desarrollo pueda ser asimilado e implementado fácilmente por la universidad en el caso que se considere este aplicativo como un aporte valioso para la mejora de la institución.



DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO

ABSTRACT: (Máximo 250 palabras)

Español:

El propósito de este proyecto es realizar la página web del programa de Ingeniería de Software mediante la aplicación de diseños vanguardistas que logren estar a la par con las tendencias de diseño actuales en la internet, siendo todo esto apoyado tras una robusta solución de control que permita la sencilla edición de los contenidos esenciales del portal, con el fin de proyectar una imagen acorde a lo esperado de un programa académico de alta calidad.

Inglés:

The purpose of this project is to make the web page of the Software Engineering program through the application of avant-garde designs that manage to be on par with current design trends on the internet, all of this being supported by a robust control solution that allows the simple edition of the essential contents of the portal, to project an image in accordance with what is expected of a high-quality academic program.

APROBACION DE LA TESIS

Nombre Jurado: **Fernando Rojas Rojas**

Firma:

Nombre Jurado: **Yhon Jerson Robles Puentes**

Firma:

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA DE SOFTWARE

PROYECTO DE GRADO:

**PORTAL WEB DINÁMICO PARA EL PROGRAMA DE INGENIERÍA DE SOFTWARE DE
LA UNIVERSIDAD SURCOLOMBIANA**

Proyecto presentado por **Fernando Andrés Andrade López** para el título de Ingeniero de Software de la Universidad Surcolombiana.

Supervisado por el director del programa de Ingeniería de Software

Fernando Rojas Rojas

Tabla de Contenido

Índice de tablas.....	6
Índice de ilustraciones	7
1. Introducción.....	10
2. Justificación.....	11
3. Problema.....	12
4. Objetivo.....	13
4.1. Objetivo general	13
4.2. Objetivos específicos.....	13
5. Marco Teórico	14
5.1. Pénum	14
5.2. PostgreSQL.....	14
5.3. Java	14
5.4. SQL.....	14
5.5. Persistencia de datos	14
5.6. CRUD.....	14
5.7. JPA	14
5.8. Spring.....	15
5.9. Spring Boot	15
5.7 Spring Security.....	15
5.8. Thymeleaf	15
5.9. Web estática	16
5.10. Web dinámica	16
5.11. Programación del lado del cliente	16
5.12. Programación del lado del servidor	16
5.13. Combinación de la programación del lado del cliente y del lado del servidor .	17
5.14. Almacenamiento de contenido	17
5.15. Frontend.....	17
5.16. Backend	17
5.17 PHP.....	17
6. Estado del arte	18
6.1. Diseño USCO para portales de programa	18
6.2 Tecnologías utilizadas por el sitio www.usco.edu.co	19

6.3 Diseños de otras universidades	19
7. Metodología	21
7.1. Kanban	21
7.2 Feature Driven Development.....	21
8. Diseño	22
8.1. Planteamiento de soluciones	22
8.1.1. Desarrollo estático	22
8.1.2. Desarrollo dinámico.....	22
8.2. Requerimientos funcionales.....	23
8.3. Requerimientos no funcionales.....	23
8.4. UML.....	23
8.5. Base de datos.....	24
8.2 Diagrama de casos de uso	26
8.3 Diccionario de datos	26
8.4 Imagen corporativa.....	30
8.4.1 Colores	30
8.4.2 Fuente tipográfica para web	31
8.4.3 Logotipo.....	31
8.5 Plantilla HTML	31
8.5.1 Mirko	31
8.6 Créditos de propiedad intelectual de imágenes.....	33
8.6.1 Licencia Unsplash	34
8.6.2 Licencia simplificada de Pixabay	35
8.6.3 Licencia Pexels.....	35
9 Resultados	37
9.2 Mensajes y noticias	37
9.3 Página principal	37
9.3.1 Secciones de la página principal.....	38
9.4 Pénum	41
9.6 Personal.....	42
9.7 Investigación	43
9.8 Preguntas frecuentes	43
9.5 Contacto	44

9.6	Sistema de administración	44
9.7	Secciones del Dashboard	45
9.7.1	Mensajes de advertencia	46
9.7.2	Facultad	47
9.7.3	Programa	47
9.7.4	Curricular	48
9.7.5	Personal	50
9.7.6	Preguntas frecuentes	52
9.7.7	Infraestructuras	52
9.7.8	Grupos de investigación	53
9.7.9	Social	53
9.7.10	Proyectos	54
9.7.11	Publicaciones	54
9.7.12	Usuarios	55
9.8	Diseño responsivo	55
10	Implementación	57
10.1	Patrones de diseño	57
10.2	Paquetes	59
10.3	Controlador principal IndexController	59
10.4	Validación de datos	60
10.5	Repositorio	62
10.6	Spring Security	62
10.7	Thymeleaf	66
10.8	Estilo CSS y JS	68
11	Despliegue	69
11.1	Heroku	69
11.2	PostgreSQL	70
11.3	Despliegue alternativo	70
12	Implementación por PHP	71
12.2	Laravel	71
12.3	Despliegue	71
12.4	Dashboard	71
15.1	Portada principal	72

13	EDT	73
14	Cronograma	74
15	Presupuesto	76
15.1	Presupuesto para conversión hacia PHP	78
15.2	Presupuesto de hosting	78
15.2.1	Spring	79
15.2.2	PHP	79
16	Conclusiones	80
17	Bibliografía	81

Índice de tablas

Tabla 1 Vínculos clave en los portales de tres programas evaluados	18
Tabla 2 Frameworks utilizados en www.usco.edu.co	19
Tabla 3 Herramientas de compatibilidad utilizados en www.usco.edu.co	19
Tabla 4 Comparativa de diseño entre algunas universidades de Colombia y Estados Unidos	19
Tabla 5 Diccionario de datos del programa	27
Tabla 6 Características de la plantilla Mirko basada en Bootstrap 5	32
Tabla 7 Lista de imágenes utilizadas con su respectiva fuente	33
Tabla 8 Elementos de la página principal	38
Tabla 9 Descripción de opciones en el Dashboard	45
Tabla 10 Capturas de diseño responsive en dispositivos móviles	55
Tabla 11 Cronograma de actividades del proyecto	74
Tabla 12 Cálculo del costo total de un empleado en Colombia, con un salario de 3M de pesos en 2021	76
Tabla 13 Listado de recursos utilizados y sus costos por periodo	76
Tabla 14 Distribución de costos por tarea	77
Tabla 15 Costo de trabajo por días para conversión a PHP	78
Tabla 16 Prestaciones sociales por parte del empleador	78
Tabla 17 Comparativa de precios para hosting con soporte para Tomcat	79
Tabla 18 Comparativa de precios para hosting con soporte para Tomcat	79

Índice de ilustraciones

Ilustración 1 Comparativa de páginas. Se respeta el diseño, pero no la manera de presentar la información	18
Ilustración 2 Modelo UML mediante JDL.....	24
Ilustración 3 Ejemplo de archivo de configuración " application.properties " para la conexión con una base de datos MariaDB.....	24
Ilustración 4 Modelo de la base de datos	25
Ilustración 5 Diagrama de casos de uso	26
Ilustración 6 Colores principales utilizados por la Universidad Surcolombiana.....	30
Ilustración 7 Logotipos verticales. Derecha, versión ocre. Izquierda, Versión blanco	31
Ilustración 8 Diseño original de la plantilla Mirko.....	32
Ilustración 10 Página principal del programa de Ingeniería de Software	37
Ilustración 11 Ejemplo de página principal del programa de Tecnología en Desarrollo de Software	38
Ilustración 12 Objetivos del programa	39
Ilustración 13 Definición del programa	39
Ilustración 14 Aplicación del programa	39
Ilustración 15 Retos del programa	40
Ilustración 16 Información general del programa	40
Ilustración 17 Perfil del profesional	40
Ilustración 18 Pénum del programa.....	41
Ilustración 19 Información de contacto, Extracto de la misión de la universidad, Créditos	41
Ilustración 20 Página propia del pénum del programa.....	41
Ilustración 21 Semáforo visual de materias y prerrequisitos.....	42
Ilustración 22 Pagina del personal académico del programa	42
Ilustración 23 Infraestructura del programa y Grupos de investigación del programa.....	43
Ilustración 24 Publicaciones científicas del programa y Proyectos del programa del programa	43
Ilustración 25 Preguntas frecuentes del programa	44
Ilustración 26 Formulario de contacto del programa	44
Ilustración 27 Página de ingreso al panel de administración del aplicativo.....	45
Ilustración 28 Página principal del portal de administración	45
Ilustración 29 Mensaje de advertencia por intento de borrado de la facultad de Ingeniería	46
Ilustración 30 Menú CRUD de Facultades	47
Ilustración 31 Menú CRUD de Programas	47
Ilustración 32 Menú superior desplegable para la sección Curricular	48
Ilustración 33 Formulario de creación de asignatura	48
Ilustración 34 Submenú de lista curricular por programa	48
Ilustración 35 CRUD del currículo de Ingeniería de Software.....	49
Ilustración 36 CRUD para Área de formación por asignatura.....	49
Ilustración 37 CRUD para NTP por asignatura.....	50
Ilustración 38 Menú superior desplegable para la sección Personal	50
Ilustración 39 Submenú de lista de personal por programa.....	50

Ilustración 40 CRUD del personal de Ingeniería de Software.....	51
Ilustración 41 Submenú de asignación de jefe por programa.....	51
Ilustración 42 CRUD para jefe de programa	51
Ilustración 43 CRUD para preguntas frecuentes	52
Ilustración 44 CRUD para infraestructuras.....	52
Ilustración 45 CRUD grupos de investigación	53
Ilustración 46 CRUD para redes sociales.....	53
Ilustración 47 CRUD para proyectos	54
Ilustración 48 CRUD para publicaciones.....	54
Ilustración 49 CRUD para usuarios	55
Ilustración 50 Ejemplo de clases de dominio	57
Ilustración 51 Ejemplo de archivos de visualización.....	57
Ilustración 52 Ejemplo de archivos controladores	57
Ilustración 53 Ejemplo de aislamiento de múltiples dependencias dentro de una única clase fachada	58
Ilustración 54 Ejemplo del paquete repository.....	59
Ilustración 55 Ejemplo del paquete service	59
Ilustración 56 Ejemplo del paquete service.impl.....	59
Ilustración 57 Estructura jerárquica de paquetes del programa.....	59
Ilustración 58 Clase inicial asignada como controladora mediante la etiqueta @Controller	60
Ilustración 59 Ejemplo de librerías javax.validation importadas para ejecutar la validación de campos de entrada	60
Ilustración 60 Anotaciones de JavaEE para la respectiva validación de datos de entrada	60
Ilustración 61 Ejemplo de implementación Try/Catch para capturar errores de excepción	61
Ilustración 62 Implementación de la etiqueta @Valid en el método de consulta de tipo POST.....	61
Ilustración 63 Advertencias de validación	61
Ilustración 64 Mensaje de error ya que se impide el borrado en cascada	62
Ilustración 65 Diagrama de flujo de verificación de seguridad mediante Spring Security (Sherwin & Calleja , 2016)	62
Ilustración 66 Clase de usuarios para Spring Security	63
Ilustración 67 Clase de roles para Spring Security	63
Ilustración 68 Repositorio de Usuario con herencia de JpaRepository	64
Ilustración 69 Interfaz UserService que hereda de UserDetailsService.....	64
Ilustración 70 Clase de tipo servicio para la implementación de UserService	64
Ilustración 71 Ejemplo de archivo de configuración de seguridad en Spring Security	65
Ilustración 72 Método de encriptado de usuarios mediante BCrypt.....	65
Ilustración 73 Ejemplo de bloqueo de URL mediante la regla de autorización "hasAuthority"	66
Ilustración 74 Etiquetas de autorización en la vista según el rol de usuario	66
Ilustración 75 Dependencia de Spring Security para Maven	67
Ilustración 76 Argumento modelo utilizado para entregar valores	67
Ilustración 77 Variables establecidas en el método model.addAttribute	67

Ilustración 78 Ejemplo de uso de variable en Thymeleaf para hacer el recorrido de un arreglo	68
Ilustración 79 Declaración del fragmento Footer	68
Ilustración 80 Implementación del fragmento Footer.....	68
Ilustración 81 Ejemplo de ingreso por terminal a través de Heroku CLI	69
Ilustración 82 Comandos de inicio de repositorio git	69
Ilustración 83 Ejemplo de proceso push hacia el master en Heroku	69
Ilustración 84 Declaración de dependencia PostgreSQL en Maven	70
Ilustración 85 Imagen de bienvenida del portal de Ingeniería de Software.....	72
Ilustración 86 Estructura de desglose de trabajo.....	73
Ilustración 87 Diagrama de Gantt del proyecto	75

1. Introducción

El propósito de este proyecto es realizar la página web del programa de Ingeniería de Software mediante la aplicación de diseños vanguardistas que logren estar a la par con las tendencias de diseño actuales en la internet, siendo todo esto apoyado tras una robusta solución de control que permita la sencilla edición de los contenidos esenciales de cada programa, con el fin de proyectar una imagen acorde a lo esperado de un programa académico de alta calidad.

Para la consecución de este proyecto se estudió la teoría de diseño que ha marcado tendencia en los últimos tres años para lograr partir de una base sólida de estilo. Luego de analizar los conceptos presentes se eligió utilizar una plantilla HTML 5 con Bootstrap 5 para reducir el tiempo de desarrollo y poder contar con un formato realizado por un diseñador experto. Es importante resaltar que aún con el uso de una plantilla prediseñada fue necesario modificarla acorde a la imagen corporativa de Universidad Surcolombiana.

Una vez establecido el diseño artístico fue igualmente importante seleccionar una tecnología adecuada que soporte todos los elementos visuales. Para esta tarea se eligió el marco de desarrollo Spring Boot, el cual es una variante simplificada de Spring Framework, pero igualmente robusta, la cual permite generar soluciones basadas en Java EE de forma rápida y fácil. Finalmente, para desplegar la parte dinámica se eligió el motor de plantillas Thymeleaf el cual es altamente apoyado por Spring.

La selección de la tecnología Spring no solo se basó en su robustez garantizada tras años en el mercado, sino también, al hecho de que la Universidad Surcolombiana apoya su infraestructura en la misma tecnología, lo cual permite que este desarrollo pueda ser asimilado e implementado fácilmente por la universidad en el caso que se considere este aplicativo como un aporte valioso para la mejora de la institución.

Desarrollar una página web simplemente informativa no requiere el uso de herramientas empresariales de tan alto nivel cuando el mensaje a exponer en línea es sencillo, pero al requerirse dos páginas con similares patrones y tras ver los problemas presentes en la Universidad Surcolombiana para definir un estándar de presentación de la información de sus programas académicos, resultó interesante realizar una aplicación capaz de proveer la visualización de la información de cualquier programa académico de ser necesario.

2. Justificación

Un sitio web es un reflejo de una marca y todo lo que representa. Por lo general, es una de las primeras cosas que las personas miran para conocer una marca y, por lo tanto, ayuda a formar la primera impresión (Commentator, 2020).

El diseño web es importante porque afecta la forma en que la audiencia percibe una marca. Si un sitio web parece poco atractivo o desactualizado, la audiencia inmediatamente tendrá una impresión negativa del negocio. La impresión que se dé puede hacer que permanezcan en la página y aprendan sobre el negocio o que abandonen la página y se dirijan a un competidor. Un buen diseño web ayuda a mantener clientes potenciales en la página (WebFX, 2021).

Hablando de primeras impresiones, los usuarios las forman casi instantáneamente. Según un estudio de Google (Google, 2012), solo toma 50 milisegundos (eso es 0.05 segundos). En esa fracción de segundo los usuarios no tienen la oportunidad de leer su sitio ni hacer clic en nada. Esa primera impresión rápida depende del diseño visual (EZ Marketing, 2018).

Con el crecimiento de los dispositivos móviles, el diseño responsivo es más importante que nunca. La audiencia accederá al sitio desde una multitud de dispositivos, incluidos teléfonos inteligentes, tabletas y computadoras de escritorio. El diseño responsivo garantiza que la audiencia tenga una buena experiencia en el sitio, independientemente del dispositivo que utilice. El sitio web se adaptará al dispositivo para garantizar que se tenga el tamaño adecuado para el dispositivo. Esto es valioso para mantener a los clientes potenciales comprometidos e interesados en su sitio (WebFX, 2021).

La Universidad Surcolombiana utiliza JavaEE (Build With, 2021) como arquitectura principal para su sitio web mediante la implementación de Spring, por lo que para una futura integración del código realizado en este proyecto con el código principal de la universidad se elige Spring Boot para acelerar la producción de código, y junto a esta tecnología se elige el motor de plantillas Thymeleaf, ya que es un motor sencillo de modificar, tiene completo soporte por parte de Spring y debido a su diseño simple de etiquetas es fácilmente modificable hacia otras soluciones de presentación tales como Angular.

3. Problema

Actualmente la Universidad Surcolombiana no tiene definido un diseño único para la presentación de información de su oferta académica, por lo que en la página oficial de la universidad es fácil apreciar la desorganización de la información entre los programas académicos. Debido a esta situación, la portada oficial de cada programa se ha ido quedando desfasada y obsoleta con relación a las tendencias de diseño y facilidad de uso actuales.

La falta de diseño no solo afecta a la presentación de los datos y la calidad de la información, sino también, afecta la reputación que proyecta la universidad a toda persona interesada en conocer su oferta académica, ya que en la era actual de la internet un sitio web no solamente se encarga de dar a conocer un mensaje, sino, además, se encarga de representar virtualmente la confiabilidad de una entidad al generar una primera gran impresión.

Además de los problemas de diseño, el programa de Ingeniería de Software se enfrenta a una excesiva burocracia administrativa la cual limita y restringe la implementación de nuevas ideas o soluciones, por lo que el programa se ve obligado a desplegar cualquier solución propia en servidores externos a la universidad hasta el momento en que el Centro de Información, Tecnologías y Control Documental (CTIC) de el visto bueno para abrir un espacio en los servidores propios de la universidad.

El programa de Ingeniería de Software ante la falta de interés por parte de la universidad para mejorar el diseño web o aportar herramientas que permitan la estandarización de la información, se ve en la necesidad de crear un diseño web propio que permita resolver estos inconvenientes.

4. Objetivo

4.1. Objetivo general

- Desarrollar la página web oficial para el programa de Ingeniería de la Universidad Surcolombiana

4.2. Objetivos específicos

- Crear un sistema de plantillas de visualización que permita la presentación de uno o varios programas académicos
- Crear un sistema de administración de información independiente al sistema oficial de la Universidad Surcolombiana
- Desplegar el sistema fuera de los servidores oficiales de la Universidad Surcolombiana
- Elegir tecnologías que puedan ser posteriormente acopladas al sistema oficial de la Universidad Surcolombiana
- Reducir el exceso de información mostrada a los visitantes y centrar la atención en los puntos clave de cada programa

5. Marco Teórico

5.1. Pénum

Plan de estudios de una carrera universitaria (Rae.es, 2021).

5.2. PostgreSQL

PostgreSQL es una base de datos relacional de código abierto de clase empresarial avanzada que admite consultas tanto SQL (relacional) como JSON (no relacional). Es un sistema de administración de bases de datos altamente estable, respaldado por más de 20 años de desarrollo comunitario que ha contribuido a sus altos niveles de resiliencia, integridad y corrección. PostgreSQL se utiliza como almacén de datos principal o almacén de datos para muchas aplicaciones web, móviles, geoespaciales y de análisis (Amazon, 2021).

5.3. Java

Java es un tipo de lenguaje de programación y una plataforma informática, creada y comercializada por Sun Microsystems en el año 1995. Se constituye como un lenguaje orientado a objetos, su intención es permitir que los desarrolladores de aplicaciones escriban el programa una sola vez y lo ejecuten en cualquier dispositivo (Content, 2019).

5.4. SQL

SQL es un lenguaje de computación para trabajar con conjuntos de datos y las relaciones entre ellos (Microsoft, 2020).

5.5. Persistencia de datos

La persistencia de datos es un medio mediante el cual una aplicación puede recuperar información desde un sistema de almacenamiento no volátil y hacer que esta persista (IBM, 2021).

5.6. CRUD

CRUD (Create, Read, Update, Delete) es un acrónimo para las maneras en las que se puede operar sobre información almacenada. Es un nemónico para las cuatro funciones del almacenamiento persistente. CRUD usualmente se refiere a operaciones llevadas a cabo en una base de datos o un almacén de datos, pero también puede aplicar a funciones de un nivel superior de una aplicación como soft deletes (marcar como borrado un registro sin en realidad eliminarlo) donde la información no es realmente eliminada, sino es marcada como eliminada a través de un estatus (Mozilla, 2021).

5.7. JPA

La API de persistencia de Java (JPA) es el estándar de Java para mapear objetos de Java a una base de datos relacional. JPA es un posible enfoque de ORM (Object Relational Mapping). A través de JPA, el desarrollador puede mapear, almacenar, actualizar y recuperar datos de bases de datos relacionales a objetos Java y viceversa. JPA se puede utilizar en aplicaciones Java-EE y Java-SE. JPA es una especificación y hay varias implementaciones disponibles. (Yadanar, 2020)

5.8. Spring

Spring es un marco ligero. Se puede pensar como un framework de frameworks porque da soporte a varios frameworks como Struts, Hibernate, Tapestry, EJB, JSF, etc. El framework, en un sentido más amplio, se puede definir como una estructura donde encontramos solución de varios problemas técnicos.

El marco de Spring comprende varios módulos como IOC, AOP, DAO, Context, ORM, WEB MVC, etc. Aprenderemos estos módulos en la página siguiente. Primero entendamos el COI y la inyección de dependencia. (javaTpoint, 2021)

5.9. Spring Boot

Spring Boot es un proyecto que se basa en Spring Framework. Proporciona una forma más fácil y rápida de instalar, configurar y ejecutar aplicaciones simples y basadas en la web.

Es un módulo Spring que proporciona la función RAD (Desarrollo rápido de aplicaciones) al Spring Framework. Se utiliza para crear una aplicación independiente basada en Spring que puede ejecutar porque necesita una configuración mínima de Spring. En resumen, Spring Boot es la combinación de Spring Framework y Embedded Servers.

En Spring Boot, no hay ningún requisito para la configuración XML (descriptor de implementación). Utiliza la convención sobre el paradigma de diseño de software de configuración, lo que significa que disminuye el esfuerzo del desarrollador. (javaTpoint, 2021)

5.7 Spring Security

Spring Security es un marco de control de acceso y autenticación potente y altamente personalizable. Es el estándar de facto para proteger las aplicaciones basadas en Spring. Spring Security es un marco que se centra en proporcionar autenticación y autorización a aplicaciones Java. Como todos los proyectos de Spring, el verdadero poder de Spring Security radica en la facilidad con la que se puede ampliar para cumplir con los requisitos personalizados (Spring.io, 2021).

5.8. Thymeleaf

Thymeleaf es una biblioteca Java de código abierto con licencia Apache License 2.0. Es un motor de plantillas HTML5 / XHTML / XML. Es un motor de plantillas Java del lado del servidor para entornos web (basados en servlets) y no web (fuera de línea). Es perfecto para el desarrollo web HTML5 JVM de hoy en día. Proporciona una integración completa con Spring Framework.

Aplica un conjunto de transformaciones a los archivos de plantilla para mostrar datos o texto producido por la aplicación. Es apropiado para servir XHTML / HTML5 en aplicaciones web.

El objetivo de Thymeleaf es proporcionar una forma elegante y bien formada de crear plantillas. Se basa en etiquetas y atributos XML. Estas etiquetas XML definen la ejecución

de la lógica predefinida en el DOM (Modelo de objeto de documento) en lugar de escribir explícitamente esa lógica como código dentro de la plantilla. Es un sustituto de JSP.

La arquitectura de Thymeleaf permite el procesamiento rápido de plantillas que depende del almacenamiento en caché de los archivos analizados. Utiliza la menor cantidad posible de operaciones de E / S durante la ejecución (javaTpoint, 2021).

5.9. Web estática

Las páginas estáticas contienen información que no cambia hasta que el diseñador o programador la modifica manualmente. En los primeros días de internet, simplemente subirse al carro de internet era importante para las corporaciones. Y no era tan importante lo que las empresas publicaban en la web, siempre y cuando tuvieran una presencia en la red donde los clientes pudieran obtener información básica sobre la entidad y sus productos.

Las páginas web estáticas, compuestas únicamente de HTML y CSS se realizan fácilmente, pero una de las grandes limitaciones de las páginas web estáticas es el esfuerzo que se requiere para actualizarlas. Cambiar un solo elemento en una página web estática requiere reconstruir y recargar en el servidor toda la página, o a veces incluso un grupo de páginas web.

Este proceso es demasiado engorroso para una organización que con frecuencia necesita publicar información en tiempo real, tal como eventos. Además, durante este proceso, un desarrollador puede cambiar accidentalmente otros artículos en la página, arruinando seriamente la información de la web, o incluso el diseño completo del sitio (Universidad de Murcia, 2021).

5.10. Web dinámica

Las páginas web dinámicas permiten cambiar fácilmente su contenido en tiempo real sin siquiera tocar el código de la página. Sin hacer manualmente cualquier cambio en la página en sí, la información en la página puede variar. Esto hace posible mantener el contenido de la página actualizado para que lo que un visitante ve allí pueda ser actualizado o substituido en un día, una hora, o un minuto. El diseño central de la página web puede seguir siendo el mismo, pero los datos presentan cambios constantes (Universidad de Murcia, 2021).

5.11. Programación del lado del cliente

En la programación del lado del cliente, se inserta el código del programa dentro del código HTML que el servidor envía al navegador del cliente. El navegador debe poder detectar el código del programa incorporado y ejecutarlo, ya sea dentro del navegador o como un programa separado fuera del navegador. La figura muestra este proceso (Universidad de Murcia, 2021).

5.12. Programación del lado del servidor

El otro aspecto de la programación web es la programación del lado del servidor. Los lenguajes de programación del lado del servidor resuelven el problema de diferentes

intérpretes de código de cliente ejecutando el código en el servidor. En la programación del servidor, el servidor web interpreta el código de programación insertado antes de enviar la página web al navegador del cliente.

Luego, el servidor toma cualquier código HTML que el código de programación genere y lo inserta directamente en la página web antes de enviarlo al cliente. El servidor hace todo el trabajo ejecutando el código de script, por lo que tiene la garantía de que cada página web se ejecutará correctamente (Universidad de Murcia, 2021).

5.13. Combinación de la programación del lado del cliente y del lado del servidor

La programación del lado del cliente y del lado del servidor tienen pros y contras. En lugar de tratar de elegir un método para crear páginas web dinámicas, se pueden utilizar ambos al mismo tiempo. Se puede insertar fácilmente el código de programación del lado del cliente y del lado del servidor en la misma página web para ejecutar en el servidor, como se muestra a continuación (Universidad de Murcia, 2021).

5.14. Almacenamiento de contenido

Con las páginas web estáticas, el contenido ya está integrado en el código de la página web. Para cambiar la información en una página web estática, se debe recodificar la página.

Desafortunadamente, la mayoría de las veces, cuando se actualiza una página web, se pierde la versión anterior. Las bases de datos son una forma fácil de almacenar y recuperar datos. Son más rápidas que el almacenamiento de datos con archivos estándar y proporcionan un nivel de seguridad para proteger sus datos. Al almacenar contenido en una base de datos, también se puede archivar y referenciar fácilmente el contenido anterior y reemplazarlo con contenido nuevo según sea necesario (Universidad de Murcia, 2021).

5.15. Frontend

Frontend es la parte de un programa o dispositivo a la que un usuario puede acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios ("maleadora", 2019).

5.16. Backend

Backend es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos. El Backend también accede al servidor, que es una aplicación especializada que entiende la forma como el navegador solicita cosas ("maleadora", 2019).

5.17 PHP

PHP es un lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web, favoreciendo la conexión entre los servidores y la interfaz de usuario.

6. Estado del arte

6.1. Diseño USCO para portales de programa

La Universidad Surcolombiana no ha establecido un sistema único de estilo para su sitio web y programas asociados, por lo que es común ver múltiples estilos que impiden una estandarización de la información.

A manera de ejemplo se tomaron tres programas: Ingeniería de Software, Tecnología en Desarrollo de Software e Ingeniería Electrónica. De estos se extrajeron los vínculos clave y se observó como la información básica no es presentada de forma común en cada uno de los programas académicos, incluso se puede apreciar que el único vínculo común entre los tres es la sección de perfil.

Tabla 1 Vínculos clave en los portales de tres programas evaluados

Ingeniería de Software	Tecnología en Desarrollo de S.	Ingeniería Electrónica
		Dirigido a
		Objetivos
Institucional	Información	
Perfiles	Perfil	Perfil
Microdiseños		
Organización		
Rasgos		
	Plan de estudio	Plan de estudios
		Funcionarios



The image displays three side-by-side screenshots of university program websites. The left screenshot is for 'Ingeniería de Software', the middle for 'Tecnología en Desarrollo de Software', and the right for 'Ingeniería Electrónica'. Each screenshot shows the university's header with the logo and '50 AÑOS' anniversary, followed by a main content area with program-specific information like 'Perfiles' (Profiles), 'Microdiseños' (Microdesigns), 'Organización' (Organization), and 'Rasgos' (Features). The rightmost screenshot also includes a 'Justificación - Introducción' (Justification - Introduction) section and a 'Denominación Académica del Programa' (Academic Name of the Program) section. The footer of each page includes standard links like 'Home', 'About the University', 'Study in USCO', 'Sedes', 'Gobernación', 'Ministry of Education', 'Ministry of Science and Innovation', and 'Universidad de Antioquia'.

Ilustración 1 Comparativa de páginas. Se respeta el diseño, pero no la manera de presentar la información

6.2 Tecnologías utilizadas por el sitio www.usco.edu.co

La Universidad Surcolombiana ha utilizado un gran número de tecnologías con el paso de los años, muchas de ellas obsoletas y descartadas; pero como motor principal de desarrollo se destaca el uso de Java EE (Build With, 2021), el cual es implementado mediante Spring.

Tabla 2 Frameworks utilizados en www.usco.edu.co

Frameworks	Primera vez detectado	Última vez detectado
Java EE	Jul-12	Sep-21
Adobe ColdFusion	Jun-16	Aug-20
PHP	Jan-16	Nov-19
Java Servlet	Nov-11	May-12
jQuery	Jul-12	Sep-21
Modernizr	Jun-20	Aug-21

Tabla 3 Herramientas de compatibilidad utilizados en www.usco.edu.co

Compatibilidad	Primera vez detectado	Última vez detectado
jQuery UI	Jun-20	Aug-21
OWL Carousel	Jun-20	Aug-21
JCarousel	Aug-12	Jul-15
jQuery Easing	Jul-12	Jul-15
PNG Fix	Nov-11	Jul-15
Ext JS	Nov-11	Jul-15
SWFObject	Jan-14	Feb-15

6.3 Diseños de otras universidades

Estos son algunos de los diseños que presentan otras universidades en Colombia y Estados Unidos en programas similares a la ingeniería de software. Se logra apreciar un mayor nivel estético junto a una navegación mejorada en la mayoría de las páginas comparadas. El diseño actual de la Universidad Surcolombiana la hace parecer una universidad estancada en diseños de mediados de los años noventa.

Tabla 4 Comparativa de diseño entre algunas universidades de Colombia y Estados Unidos

	Universidad de Los Andes
---	--------------------------

	<p>Universidad EAFIT</p>
	<p>Universidad Javeriana</p>
	<p>Universidad Sergio Arboleda</p>
	
	<p>Stanford University</p>

7. Metodología

7.1. Kanban

Durante la etapa de investigación, “un enfoque de proyecto Kanban a menudo se adapta mejor para rastrear y exponer el progreso del proyecto. La exploración de datos, la construcción inicial del modelo y el análisis generalmente requieren grandes bloques de tiempo de enfoque ininterrumpido. Incluso las reuniones breves pueden romper la concentración y provocar una pérdida de impulso y contexto, que son clave para un análisis complejo y, por lo tanto, ralentizan el progreso. Los enfoques Kanban admiten la duración difícil de predecir de gran parte de la exploración y experimentación de datos durante esta fase, lo que permite bloques de tiempo más prolongados y productivos, al tiempo que refleja el progreso, las dependencias y los bloqueos. (Megler, 2019)”

El desarrollo de software Kanban es un modelo fluido que le permite agregar y cambiar las prioridades de las historias sobre la marcha. Minimiza la tensión entre el negocio y la ingeniería, y evita que el equipo de desarrollo tenga que lidiar con cambios de última hora o en la mitad del proceso. El poder de Kanban proviene de su enfoque continuo en la calidad del software, reduciendo cualquier actividad derrochadora o cambio de contexto, y creando el máximo valor para el cliente. La técnica Kanban combate cualquier tentación de tomar atajos y omitir algunos pasos importantes en el proceso (Skelia, 2017).

Debido a que es el desarrollo de una sola persona, no hay forma de establecer cargos administrativos como ocurre en Scrum, por lo que Kanban es una mejor opción a la hora de trabajar en un proyecto solitario, ya que solo se requiere listar las necesidades a cubrir, resolverlas y por último dejarlas en la categoría de tareas cumplidas.

7.2 Feature Driven Development

Feature Driven Development, o FDD es una metodología de desarrollo de software basada en Agile. Su objetivo es simple, evitar confusiones que conduzcan a costosos reprocesos. A veces, FDD se confunde con centrarse en cada una de las funciones del software. No lo es.

Lo que hace el desarrollo basado en características es dividir las actividades de desarrollo en una lista de características en el modelo general. Para cada una de las características, los desarrolladores pasan por una iteración de planificación, diseño y construcción. Normalmente, una función no debería tardar más de dos semanas en completarse.

El resultado de FDD son resultados rápidos e impactantes para cada una de las actividades enumeradas como características. Este enfoque está destinado a equipos grandes y la información se comunica a través de documentación detallada (Nikolaieva, 2021).

8. Diseño

8.1. Planteamiento de soluciones

8.1.1. Desarrollo estático

Pros

- Se puede crear un sitio web rápido
- El desarrollo es sencillo
- No requiere base de datos
- Solo hace una consulta para descargar el archivo HTML
- Puede utilizar bases de datos de terceros (ej.: Facebook) para gestionar la creación y almacenamiento de mensajes o noticias a la comunidad, los cuales pueden ser mostrados en el sitio web estático mediante plugin

Contra

- Solo puede ser actualizado mediante el cambio en su código original
- Requiere una persona con conocimientos suficientes para entender la tecnología HTML utilizada
- Debe depender de bases de datos de terceros (ej.: Facebook) para almacenar sus noticias o mensajes a la comunidad
- No puede realizar ningún tipo de cambio sencillo en el contenido de la página
- Deben crearse archivos para cada programa

8.1.2. Desarrollo dinámico

Pros

- La vista es dinámica en su contenido
- Se puede crear un backend fijo y personalizado con todos los campos necesarios para el sitio web
- Solo se requiere un conocimiento básico de computadores para poder modificar la información del backend

Contra

- El backend requiere un servidor más especializado dependiendo de la tecnología
- Requiere el uso de bases de datos
- Se incrementa el número de pasos internos para entregar la información al usuario ya que se agrega la consulta a la base de datos
- Se requiere de alguien muy especializado en el caso de necesitar crear nuevos campos en el backend o diseños del frontend

8.2. Requerimientos funcionales

- Función CRUD para todos los elementos modelables del Frontend
- Sistema de correo de contacto independiente al servidor de alojamiento
- La vista ser compatible con dispositivos móviles
- El sistema de contraseñas debe ser salteado (Salt hashing)
- Los campos opcionales no completados en la base de datos deben ser correctamente ocultados en la vista
- Los listados de datos en el dashboard deben ser creados de acuerdo con el programa
- Los campos de número de contacto del programa deben ser de texto
- Las imágenes deben ser alojadas en servidores externos para reducir la carga del servidor de alojamiento
- Los correos de contacto deben declarar el tipo de asunto administrativo a tratar
- Se debe agregar un certificado de seguridad SSL para habilitar el protocolo HTTPS

8.3. Requerimientos no funcionales

- La aplicación web deberá ser fácil de usar para todos los empleados.
- Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.
- El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 4 horas.
- La página debe ser visible de manera completa en una resolución mínima de 720p

8.4. UML

Mediante la herramienta en línea JDL Studio de JHipster se desarrolló la visualización del modelo UML. “El JDL es un lenguaje de dominio específico de JHipster donde puede describir todas sus aplicaciones, implementaciones, entidades y sus relaciones en un solo archivo (o más de uno) con una sintaxis fácil de usar” (JHipster, 2021).

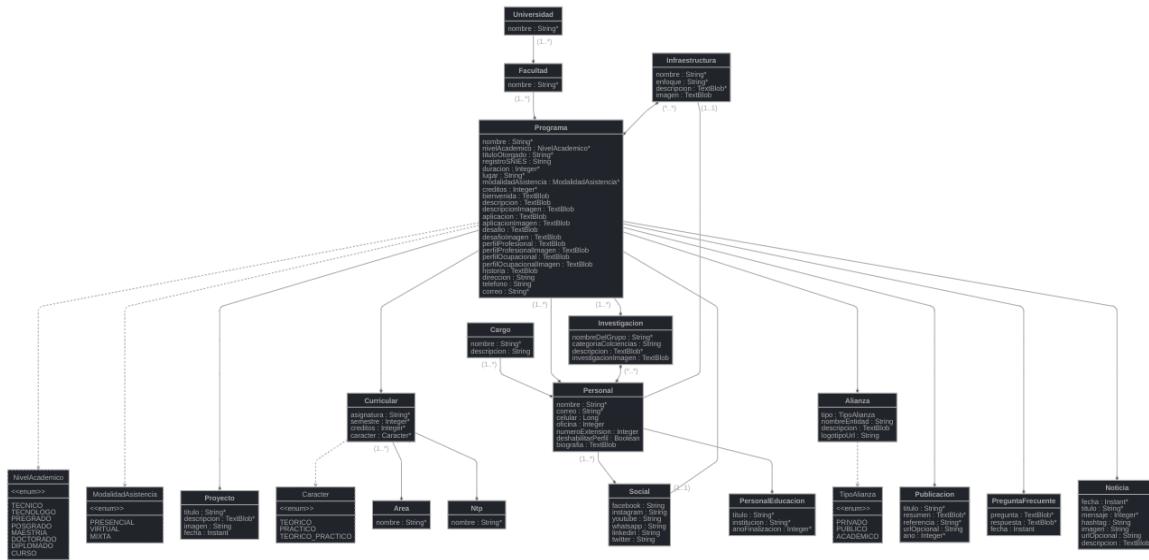


Ilustración 2 Modelo UML mediante JDL

8.5. Base de datos

La aplicación al estar desarrollada en Spring Boot y utilizando JPA permite que pueda ser desplegada en la mayoría de los motores SQL sin necesidad de precargar el modelo de la base de datos ya que JPA se encarga de la tarea de configuración y despliegue en la base de datos objetivo sin que el usuario tenga que hacer mayor trabajo de configuración, tan solo se necesita establecer los datos de seguridad en el archivo "application.properties" para el ingreso al servidor de datos.

```

1  # =====
2  # = DATA SOURCE
3  # =====
4  # Set here configurations for the database connection
5  spring.datasource.url=jdbc:mariadb://localhost:3306/portal
6  spring.datasource.username=root
7  spring.datasource.password=1234
8  spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
9  # Keep the connection alive if idle for a long time (needed in production)
10 spring.datasource.testWhileIdle=true
11 spring.datasource.validationQuery=SELECT 1

```

Ilustración 3 Ejemplo de archivo de configuración " application.properties " para la conexión con una base de datos MariaDB

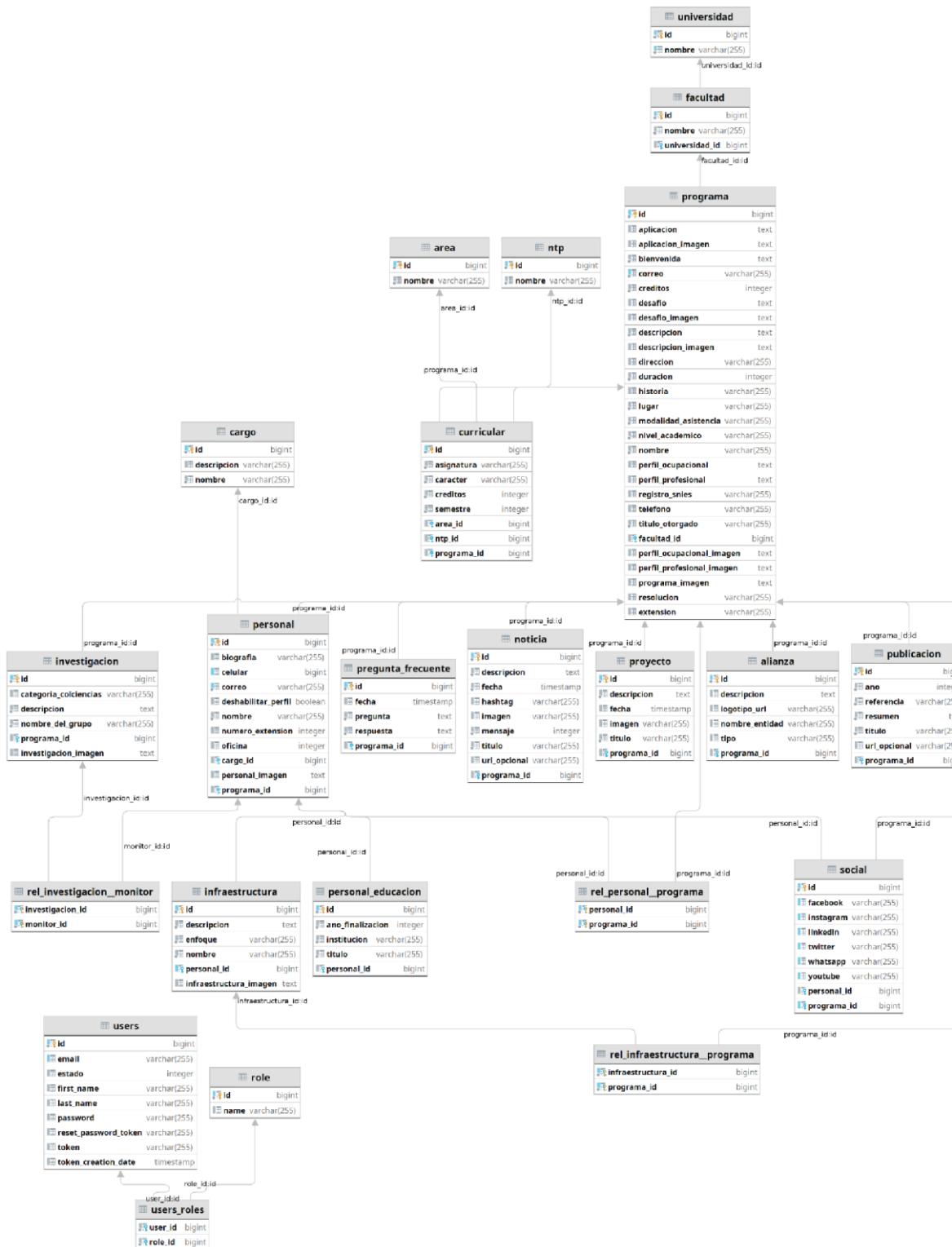


Ilustración 4 Modelo de la base de datos

8.2 Diagrama de casos de uso

Las funcionalidades generales se han establecido en un solo caso de uso en donde se establece la relación de uso con sus actores.

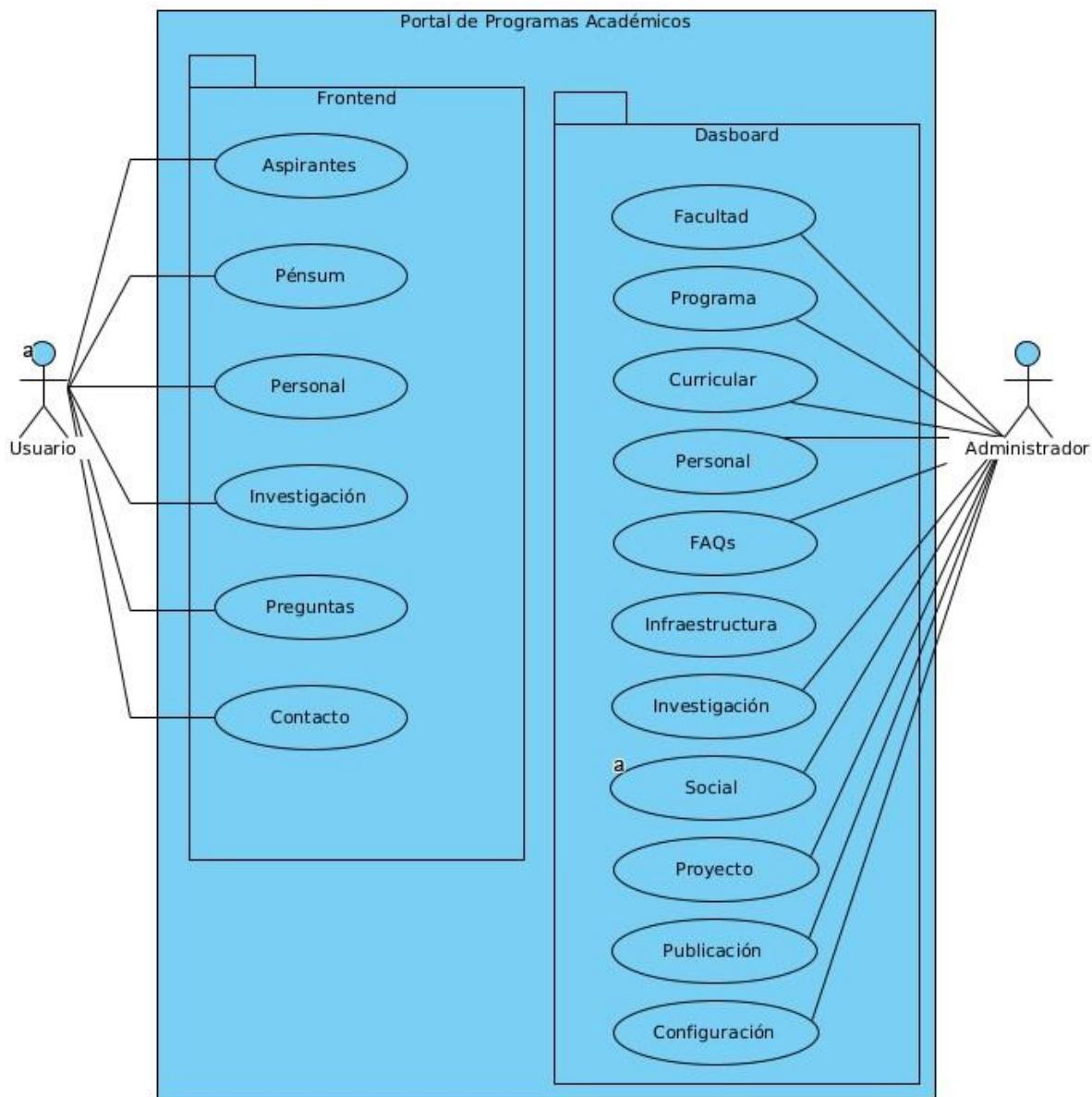


Ilustración 5 Diagrama de casos de uso

8.3 Diccionario de datos

Un diccionario de datos es una colección de nombres, definiciones y atributos sobre elementos de datos que se utilizan o capturan en una base de datos, sistema de información o parte de un proyecto de investigación. Describe los significados y propósitos de los elementos de datos dentro del contexto de un proyecto y brinda orientación sobre la interpretación, los significados aceptados y la representación (UC Merced, 2021).

Tabla 5 Diccionario de datos del programa

Nombre Entidad		Tipo de datos	Longitud	PK	Nulo	Único
	Nombre columna					
area	id	int8	0	true	false	false
	nombre	varchar	255	false	false	false
cargo	descripcion	varchar	255	false	true	false
	id	int8	0	true	false	false
curricular	nombre	varchar	255	false	false	false
	area_id	int8	0	false	true	false
facultad	asignatura	varchar	255	false	false	false
	caracter	varchar	255	false	false	false
infraestructura	creditos	int4	0	false	false	false
	id	int8	0	true	false	false
investigacion	ntp_id	int8	0	false	true	false
	programa_id	int8	0	false	true	false
noticia	semestre	int4	0	false	false	false
	descripcion	text	0	false	false	false
noticia	enfoque	varchar	255	false	false	false
	id	int8	0	true	false	false
noticia	infraestructura_imagen	text	0	false	true	false
	nombre	varchar	255	false	false	false
noticia	personal_id	int8	0	false	true	false
	categoria_colciencias	varchar	255	false	true	false
noticia	descripcion	text	0	false	false	false
	id	int8	0	true	false	false
noticia	investigacion_imagen	text	0	false	true	false
	nombre_del_grupo	varchar	255	false	false	false
noticia	programa_id	int8	0	false	true	false
	descripcion	text	0	false	false	false
noticia	fecha	timestamp	6	false	false	false
	hashtag	varchar	255	false	true	false
noticia	id	int8	0	true	false	false
	imagen	varchar	255	false	true	false

	mensaje	int4	0	false	false	false
	programa_id	int8	0	false	true	false
	titulo	varchar	255	false	false	false
	url_opcional	varchar	255	false	true	false
📁 ntp						
	id	int8	0	true	false	false
	nombre	varchar	255	false	false	false
📁 personal						
	biografia	varchar	255	false	true	false
	cargo_id	int8	0	false	true	false
	celular	int8	0	false	true	false
	correo	varchar	255	false	false	false
	deshabilitar_perfil	bool	0	false	true	false
	id	int8	0	true	false	false
	nombre	varchar	255	false	false	false
	numero_extension	int4	0	false	true	false
	oficina	int4	0	false	true	false
	personal_imagen	text	0	false	true	false
	programa_id	int8	0	false	true	false
📁 personal_educacion						
	ano_finalizacion	int4	0	false	false	false
	id	int8	0	true	false	false
	institucion	varchar	255	false	false	false
	personal_id	int8	0	false	true	false
	titulo	varchar	255	false	false	false
📁 pregunta_frecuente						
	fecha	timestamp	6	false	true	false
	id	int8	0	true	false	false
	pregunta	text	0	false	false	false
	programa_id	int8	0	false	true	false
	respuesta	text	0	false	false	false
📁 programa						
	aplicacion	text	0	false	true	false
	aplicacion_imagen	text	0	false	true	false
	bienvenida	text	0	false	false	false
	correo	varchar	255	false	false	false
	creditos	int4	0	false	false	false
	desafio	text	0	false	true	false
	desafio_imagen	text	0	false	true	false
	descripcion	text	0	false	false	false
	descripcion_imagen	text	0	false	true	false
	direccion	varchar	255	false	true	false
	duracion	int4	0	false	false	false
	extension	varchar	255	false	true	false
	facultad_id	int8	0	false	true	false

	historia	varchar	255	false	true	false
	id	int8	0	true	false	false
	lugar	varchar	255	false	false	false
	modalidad_asistencia	varchar	255	false	false	false
	nivel_academico	varchar	255	false	false	false
	nombre	varchar	255	false	false	false
	perfil_ocupacional	text	0	false	true	false
	perfil_ocupacional_imagen	text	0	false	true	false
	perfil_profesional	text	0	false	true	false
	perfil_profesional_imagen	text	0	false	true	false
	programa_imagen	text	0	false	true	false
	registro_snies	varchar	255	false	true	false
	resolucion	varchar	255	false	true	false
	telefono	varchar	255	false	true	false
	titulo_otorgado	varchar	255	false	false	false
 proyecto						
	descripcion	text	0	false	false	false
	fecha	timestamp	6	false	true	false
	id	int8	0	true	false	false
	imagen	varchar	255	false	true	false
	programa_id	int8	0	false	true	false
	titulo	varchar	255	false	false	false
 publicacion						
	ano	int4	0	false	false	false
	id	int8	0	true	false	false
	programa_id	int8	0	false	true	false
	referencia	varchar	255	false	false	false
	resumen	text	0	false	false	false
	titulo	varchar	255	false	false	false
	url_opcional	varchar	255	false	true	false
 rel_infraestructura_programa						
	infraestructura_id	int8	0	true	false	false
	programa_id	int8	0	true	false	false
 rel_investigacion_monitor						
	investigacion_id	int8	0	true	false	false
	monitor_id	int8	0	true	false	false
 rel_personal_programa						
	personal_id	int8	0	true	false	false
	programa_id	int8	0	true	false	false
 role						
	id	int8	0	true	false	false
	name	varchar	255	false	true	false
 social						
	facebook	varchar	255	false	true	false
	id	int8	0	true	false	false

	instagram	varchar	255	false	true	false
	linkedin	varchar	255	false	true	false
	personal_id	int8	0	false	true	false
	programa_id	int8	0	false	true	false
	twitter	varchar	255	false	true	false
	whatsapp	varchar	255	false	true	false
	youtube	varchar	255	false	true	false
📁 universida d						
	id	int8	0	true	false	false
	nombre	varchar	255	false	false	false
📁 users						
	email	varchar	255	false	true	false
	estado	int4	0	false	true	false
	first_name	varchar	255	false	true	false
	id	int8	0	true	false	false
	last_name	varchar	255	false	true	false
	password	varchar	255	false	true	false
	reset_password_token	varchar	255	false	true	false
	token	varchar	255	false	true	false
	token_creation_date	timestamp	6	false	true	false
📁 users_roles						
	role_id	int8	0	false	false	false
	user_id	int8	0	false	false	false

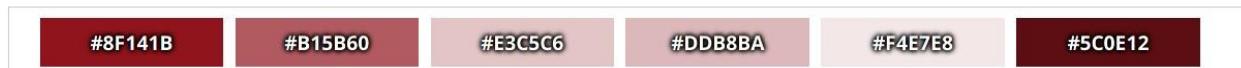
8.4 Imagen corporativa

Para la elaboración de la página web se siguieron los lineamientos establecidos por la Universidad Surcolombiana en su Guía de Imagen Corporativa (USCO, 2021).

8.4.1 Colores

Principales

Vino tinto: #8F141B



Gris: #4D626C



Ocre: #DFD4A6

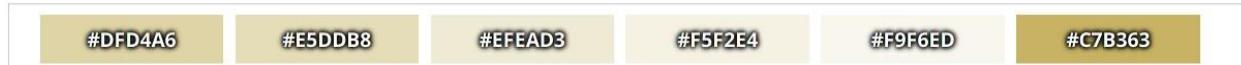


Ilustración 6 Colores principales utilizados por la Universidad Surcolombiana

8.4.2 Fuente tipográfica para web

- Open Sans
- Invocación de fuente desde Google Font:
 - <link href="https://fonts.googleapis.com/css?family=Open+Sans:400,400i,700,700i&rel=stylesheet">
- CSS:
 - body{ font-family: 'Open Sans', Arial, Helvetica, sans-serif; }

8.4.3 Logotipo



8.5 Plantilla HTML

En la ingeniería de software se invierte tiempo aprendiendo, desarrollando e implementando código computacional, por lo que el arte del diseño estético queda en un plano mayormente olvidado, por esta razón es buena práctica contar con diseñadores que realicen modelos y plantillas visualmente atractivas para el cliente y los usuarios. Debido a esto se decidió utilizar una plantilla prediseñada con licencia libre para uso comercial.

Se eligió la plantilla Mirko basada en Bootstrap 5 y HTML 5, la cual brinda un aspecto muy elegante y llamativo, que evoca confianza y modernidad ya que no sobrecarga al lector con demasiada información.

8.5.1 Mirko

Mirko es una plantilla Bootstrap 5 gratuita y prometedora para sitios web de servicios empresariales y corporativos. Es completamente funcional como plantilla de sitio web de servicios de banda ancha, pero puede adaptar fácilmente su diseño web a las necesidades de su propia empresa. Esta plantilla comercial es un diseño de tema oscuro moderno con colores de llamada a la acción contrastantes y elementos gráficos atractivos que se suman a la experiencia general. Para cualquier negocio nuevo, este puede ayudarlo a presentar toda la información requerida de una manera agradable y efectiva sobre sus servicios. (Themewagon, 2021)

Tabla 6 Características de la plantilla Mirko basada en Bootstrap 5

<ul style="list-style-type: none"> • Bootstrap 5 • HTML5 y CSS • 100% receptivo • Encabezado de héroe • Compatibilidad con varios navegadores • Diseño de varias páginas • Sobre los efectos de desplazamiento • Menú desplegable • Páginas extra • Barra de navegación fija 	<ul style="list-style-type: none"> • Sección de servicio • Tabla de precios • Contadores de datos divertidos • Carrusel de testimonios • Suscripciones a los boletines • Formulario de contacto • Fuente impresionante • Fuentes de Google • Navegación de pie de página
--	---

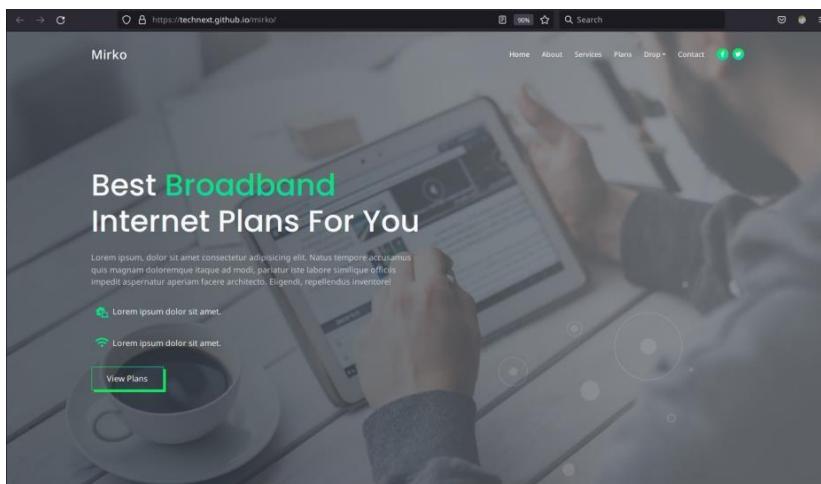


Ilustración 8 Diseño original de la plantilla Mirko

8.5.1.1 *Licencia de la plantilla Mirko*

Licencia simple

Las plantillas de Inovatik se ofrecen bajo la licencia MIT de código abierto, lo que significa que puede usarlas para cualquier tipo y cantidad de proyectos, incluidos los comerciales.

Permisos

- Utilice las plantillas para cualquier proyecto personal y comercial tantas veces como desee
- Utilice las plantillas para cualquier tipo de sitio web, aplicación, tema de WordPress, tema de CMS, creador de sitios web, producto digital
- Puede eliminar los derechos de autor de Inovatik y realizar cualquier otra modificación que necesite

- Básicamente, puede usar las plantillas para cualquier cosa además de las restricciones que se indican a continuación.

Restricciones

- Tiene estrictamente prohibido alojar las plantillas en cualquier sitio web y ofrecerlas de forma gratuita o de pago, de forma similar a como se ofrecen en Inovatik.com.
- No se le permite usar las plantillas para proyectos que estén destinados a dañar a personas u organizaciones de ninguna manera, incluidos spam, piratería, phishing y cualquier otra actividad maliciosa. (Inovtik, 2021)

8.6 Créditos de propiedad intelectual de imágenes

Las imágenes utilizadas en el desarrollo de la plantilla son de una gran calidad y fueron seleccionadas de repositorios de imágenes libres para uso comercial.

Tabla 7 Lista de imágenes utilizadas con su respectiva fuente

Frontal TDS	https://unsplash.com/photos/5tBqnBl7tKs		
Frontal IS	https://pixabay.com/es/photos/mujerordendores-oficina-laboral-5653501/		
Retos	https://images.unsplash.com/photo-1549692520acc6669e2f0c?ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&ixlib=rb1.2.1&auto=format&fit=crop&w=687&q=80		

Perfil Profesional	https://i.ibb.co/9YhMyC2/pexels-photo3758105.jpg	
Perfil Ocupacional	https://images.unsplash.com/photo1600880292089-90a7e086ee0c?ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&ixlib=rb1.2.1&auto=format&fit=crop&w=687&q=80	
Aplicaciones	Collage propio (fuentes: Pexels.com): <ul style="list-style-type: none"> • pexels-photo-2468399 • pexels-photo-3184423 • pexels-photo-4508751 • pexels-photo-4960630 • plus-and-minus-daily-profit-stockexchange-numbers-2210x1473 • pexels-photo-245252 • pexels-photo-3861969 	

8.6.1 Licencia Unsplash

Las fotos de Unsplash están hechas para ser utilizadas libremente. Nuestra licencia refleja eso.

- Todas las fotos se pueden descargar y utilizar de forma gratuita.
- Fines comerciales y no comerciales
- No se necesita permiso (¡aunque se agradece la atribución!)

Lo que no está permitido

- Las fotos no se pueden vender sin modificaciones significativas.
- Compilación de fotos de Unsplash para replicar un servicio similar o de la competencia.

Consejo: Cómo dar atribución

- Aunque no se requiere la atribución, los fotógrafos de Unsplash lo aprecian, ya que brinda exposición a su trabajo y los alienta a seguir compartiendo.

Forma larga

Unsplash le otorga una licencia de derechos de autor irrevocable, no exclusiva y mundial para descargar, copiar, modificar, distribuir, realizar y usar fotos de Unsplash de forma gratuita, incluso con fines comerciales, sin permiso del fotógrafo o de Unsplash o atribuyéndolo. Esta licencia no incluye el derecho de compilar fotos de Unsplash para replicar un servicio similar o de la competencia.

8.6.2 Licencia simplificada de Pixabay

Nuestra licencia empodera a los creadores y protege a nuestra comunidad. Queremos mantenerlo lo más simple posible. Aquí hay una descripción general de para qué se puede y no se puede usar el contenido de Pixabay.

¿Qué está permitido?

- Todo el contenido (por ejemplo, imágenes, videos, música) en Pixabay se puede usar de forma gratuita para uso comercial y no comercial en medios impresos y digitales, excepto en los casos mencionados en "Qué no está permitido".
- No se requiere atribución. No es necesario dar crédito al artista o a Pixabay, pero nuestra comunidad siempre lo agradece.
- Puedes hacer modificaciones al contenido de Pixabay.

¿Qué no está permitido?

Esta sección solo se aplica a los usuarios de medios y no a los artistas correspondientes.

- No redistribuya ni venda contenido de Pixabay en otras plataformas de acciones o fondos de pantalla.
- No venda copias inalteradas de contenido, p. no venda una copia exacta de una foto de stock como póster, impresión o en un producto físico.
- No retrate a personas identificables de mala manera o de una manera que sea ofensiva.
- No use contenido con personas o marcas identificables para crear una asociación engañosa con un producto o servicio.

8.6.3 Licencia Pexels

Simplicidad legal

Todas las fotos y videos de Pexels se pueden descargar y utilizar gratis.

¿Qué está permitido?

- Todas las fotos y videos en Pexels se pueden utilizar gratuitamente.
- No hace falta mencionar la fuente. No es obligatorio elogiar públicamente al fotógrafo o a Pexels, aunque es un lindo detalle.

- Puedes modificar las fotos y vídeos de Pexels. Utiliza tu creatividad y edítalos como quieras.

¿Qué no está permitido?

Nuestros colaboradores trabajan con dedicación y merecen respeto; por eso, es importante que recuerdes estas restricciones.

- Las personas identificables en la foto no deben aparecer en una situación donde resulten desfavorecidas o de una manera que pueda ser ofensiva.
- No vendas copias sin editar una foto o vídeo, por ejemplo, en formato de póster, impresión o en un producto físico sin haberla modificado antes.
- No puedes utilizar la imagen de forma que sugiera que las personas o marcas de la imagen están promocionando tu producto.
- No puedes redistribuir o vender las fotos y vídeos en otras plataformas de fotos de stock o de fondos de pantalla.

9 Resultados

El siguiente es el diseño final que se produjo a partir de la plantilla Mirko y que además fue modificada para seguir los lineamientos de colores establecidos por la imagen corporativa de la Universidad Surcolombiana.

Las fotografías usadas en el diseño son de uso libre para uso comercial de acuerdo con lo establecido en sus respectivas licencias de distribución. Las imágenes usadas no representan el diseño final, ya que la Universidad Surcolombiana generará sus propias imágenes para así poder remplazar las imágenes de uso libre.

9.2 Mensajes y noticias

Se decidió no presentar información de noticias dentro del nuevo sitio web ya que actualmente la comunidad estudiantil del programa de Ingeniería de ha elegido a la página oficial del programa en Facebook como centro único de información, y con ello se elimina el problema de publicar o actualizar la misma información en diferentes redes sociales o plataformas web.

9.3 Página principal

La portada principal se divide en tres aspectos: el menú superior de navegación, la información básica del programa (nombre, universidad, facultad, resolución, descripción corta), y finalmente la imagen principal representativa del programa.

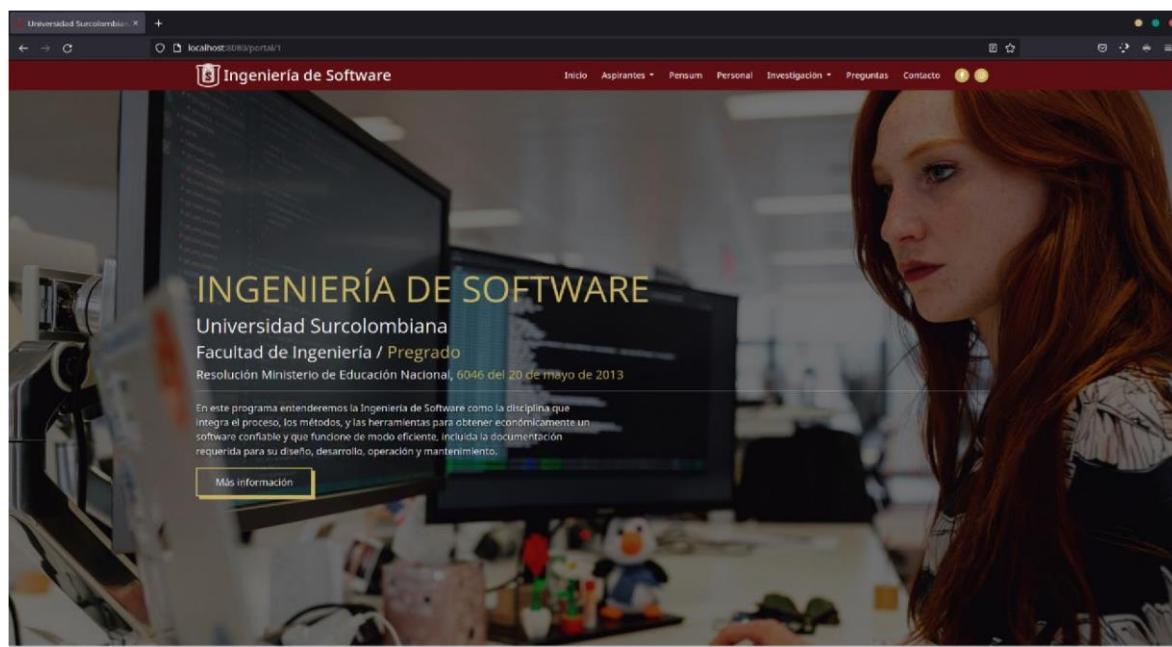


Ilustración 9 Página principal del programa de Ingeniería de Software



Ilustración 10 Ejemplo de página principal del programa de Tecnología en Desarrollo de Software

9.3.1 Secciones de la página principal

Las secciones de la página principal solo aparecen si hay contenido textual que ser presentado, de otro modo no se genera la vista.

Tabla 8 Elementos de la página principal

Tema	Descripción
Definición del programa	Descripción corta y clara acerca de lo que es el programa.
Objetivos	Metas propuestas por el programa de Ingeniería de Software
Aplicaciones	Ejemplos de posibles usos en los que hace parte el profesional en el área de estudio.
Retos	Desafíos a los que se puede enfrentar en la vida laboral.
Información general del programa	Descripción rápida de los puntos clave del programa.
Perfil	Rasgos particulares que se espera tendrá el graduado del programa.
Plan de estudio	Listado de asignaturas por semestre.
Datos de contacto	Información general de contacto del programa.
Descripción de la universidad	Extracto de la misión oficial de la Universidad Surcolombiana (Universidad Surcolombiana, 2021).
Créditos	Información sobre derechos de autor.

https://pregrado.insofuso.com/ +

pregrado.insofuso.com/#objetivo

Ingeniería de Software

Inicio Aspirantes Pensum Personal Investigación Preguntas Contacto

Objeto de Estudio

La Ingeniería de Software de la Universidad Surcolombiana tiene como objeto de estudio la construcción de modelos para la solución computarizada de problemas y la aplicación de técnicas para el análisis, diseño y desarrollo de componentes lógicos que hacen parte de los sistemas computacionales.

Objeto de Formación

El objeto de la formación profesional visiónado es diseñar, analizar, administrar, manejar y controlar, con una visión empresarial, diferentes entornos bajo el enfoque de la teoría general de sistemas y sus herramientas de soporte, actuando como agentes impulsadores de innovaciones tecnológicas con un alto criterio de responsabilidad social.

Objetivo General

Contribuir a la región y al país en la formación de profesionales integrales con capacidad de resolver problemas relevantes de la Ingeniería de software con alto rigor científico y tecnológico.

Objetivos Específicos

Atender y resolver los problemas del manejo de la información, así como la investigación y aplicación de tecnologías informáticas, asesoría, consultoría y outsourcing en: el desarrollo de software, ingeniería de software, ciencias de la computación, gestión y administración informática.

Gestionar y administrar sistemas de bases de datos y procesos relacionados con el desarrollo de software orientado a los servicios de TI, para asegurar la fiabilidad, disponibilidad, eficacia y la eficiencia de la información en las organizaciones.

Analizar y administrar sistemas de seguridad relacionados con la informática y la información para proponer soluciones a través de estrategias y modelos de seguridad que contribuyan a mitigar los riesgos y vulnerabilidad en los sistemas informáticos de las organizaciones.

Ilustración 11 Objetivos del programa

https://pregrado.insofuso.com/ +

pregrado.insofuso.com/#definicion

Ingeniería de Software

Inicio Aspirantes Pensum Personal Investigación Preguntas Contacto

DEFINICIÓN

¿Qué es la Ingeniería de Software?

La IEEE define la ingeniería de software como:

La aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software.

El pionero de la ciencia de computación, Friedrich L. Bauer, define la ingeniería de software como:

El establecimiento y uso de sólidos principios de ingeniería para obtener de manera económica un software que sea confiable y funcione en máquinas reales de manera eficiente.

La ingeniería de software es la aplicación de un enfoque científico empírico para encontrar soluciones económicas y eficientes a problemas prácticos de software. [Modern Software Engineering: Doing What Works to Build Better Software Faster, David Farley]

Producir una aplicación de software es un concepto relativamente simple: tome una idea y conviértala en un programa. Desafortunadamente para los proyectos de cualquier alcance real, hay innumerables formas en que un concepto simple puede salir mal. [Beginning Software Engineering, Rod Stephens]

La ingeniería de software incluye técnicas para evitar las muchas tareas que, de otro modo, podrían hacer que su proyecto fracase. Garantiza que la aplicación final sea eficaz, utilizable y mantenible. Le ayuda a cumplir hitos a tiempo y producir un proyecto terminado a tiempo y dentro del presupuesto. Quizás lo más importante es que la ingeniería de software le brinda la flexibilidad de realizar cambios para satisfacer demandas inesperadas sin eliminar por completo sus restricciones de cronograma y presupuesto. [Beginning Software Engineering, Rod Stephens]

Ciclo de vida del Desarrollo de Software

Ilustración 12 Definición del programa

https://pregrado.insofuso.com/ +

pregrado.insofuso.com/#aplicaciones

Ingeniería de Software

Inicio Aspirantes Pensum Personal Investigación Preguntas Contacto

APLICACIONES

Amplia variedad de campos

Las aplicaciones de ingeniería de software son prácticamente ilimitadas. De hecho, la ingeniería de software ha crecido tan rápido en las últimas décadas que se ha conectado estrechamente con todas las demás áreas de la informática (por ejemplo, inteligencia artificial, procesamiento paralelo, gráficos por computadora, interacción persona-computadora, sistemas de administración de bases de datos y muchas más), así como con algunas áreas fuera del dominio informático tradicional (como las ciencias cognitivas y las artes visuales).

Algunos de los campos de especialización que existen:

- 1. Diseñador de videojuegos
- 2. Ingeniero de aseguramiento de calidad
- 3. Gerente de proyecto CRM
- 4. Ingeniero de integración de software
- 5. Ingeniero de Seguridad
- 6. Ingeniero de front-end
- 7. Ingeniero de Full-stack
- 8. Desarrollador de gráficos 3D
- 9. Científico de datos
- 10. Desarrollador móvil
- 11. Ingeniero DevOps
- 12. Ingeniero de Back-end
- 13. Ingeniero de aprendizaje automático

Ilustración 13 Aplicación del programa

RETOS

Principales desafíos

Los desafíos actuales en el desarrollo de software incluyen:

1. Volatilidad de los requisitos del software
2. Competencia extremadamente alta
3. Avances tecnológicos interminables
4. Numerosas iteraciones de software para pruebas de calidad y corrección de errores
5. Desconocimiento de la visión del cliente respecto al proyecto
6. Integración de múltiples tecnologías
7. Amenazas a la seguridad
8. Restricciones de tiempo de desarrollo
9. Comprender las mejores prácticas de desarrollo de código

El software es complejo y las aplicaciones de software suelen contener millones de líneas de código. El software se encuentra entre los sistemas más complejos jamás creados por la humanidad, incluyendo las grandes ciudades, los gobiernos de los países, los grandes ejércitos, las grandes corporaciones, etc. Cuando se trata de esta complejidad, los programadores están equipados con una memoria a corto plazo muy pequeña (Miller, 1956); por lo tanto, tienen que emplear varias estrategias para manejar esta complejidad. Existen varias estrategias bien conocidas que se ocupan de la complejidad y han sido adoptadas por los ingenieros de software. [Software Engineering, Vadav Rajlich]



Ilustración 14 Retos del programa

PROGRAMA

Información General

Título otorgado Ingeniero de Software	Duración en semestres 10	Créditos 165
Nivel académico PREGRADO	Modalidad de formación MIXTA	Registro SNIES 102526
Lugar de oferta Neiva		

Ilustración 15 Información general del programa

PERFIL PROFESIONAL



Emprendedor y líder

El Ingeniero de Software de la Universidad Surcolombiana es un profesional emprendedor con profundo conocimiento de las ciencias y las tecnologías de la información y comunicación, con capacidad para crear y liderar proyectos de desarrollo de software, con criterios de calidad internacional.

PERFIL OCUPACIONAL

Multidisciplinario

El perfil ocupacional del egresado del programa Ingeniería de Software de la Universidad Surcolombiana es:

- Desarrollador de Software
- Analista y Diseñador de Sistemas de Información
- Arquitecto de Software
- Administrador de proyectos de desarrollo de sistemas de información
- Coordinador de la implementación de grandes aplicaciones.

Ilustración 16 Perfil del profesional

Ilustración 17 Pénsum del programa

Ilustración 18 Información de contacto, Extracto de la misión de la universidad, Créditos

9.4 Pénsum

Muestra la información de créditos y carácter de cada asignatura.

Semestre 1		
Asignatura	Créditos	Carácter
Algorítmica de programación I	3	TEÓRICO
Arquitectura de Hardware	3	TEÓRICO, PRACTICO
Comunicación Lingüística	2	TEÓRICO
Dibujo de Ingeniería	2	TEÓRICO, PRACTICO
Fundamentos de Matemáticas	3	TEÓRICO
Inglés I	0	TEÓRICO
Introducción a la ingeniería de software	2	TEÓRICO
Medio Ambiente	1	TEÓRICO

Semestre 2		
Asignatura	Créditos	Carácter
Álgebra Lineal	3	TEÓRICO
Algorítmica para programación II	3	TEÓRICO
Cálculo Diferencial	4	TEÓRICO
Constitución Política	1	TEÓRICO
Deporte Formativo	0	PRACTICO
Física Mecánica	4	TEÓRICO, PRACTICO
Inglés II	0	TEÓRICO

Ilustración 19 Página propia del pénsum del programa

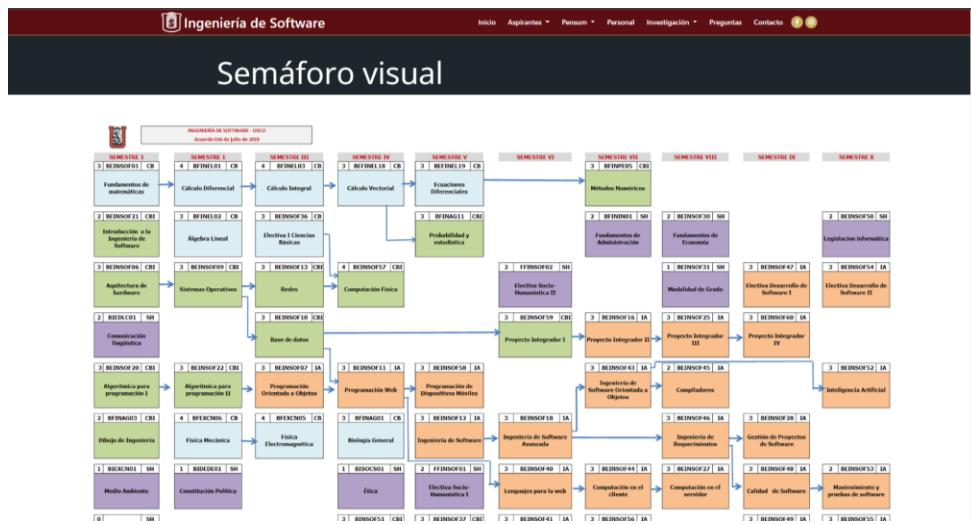


Ilustración 20 Semáforo visual de materias y prerequisitos

9.6 Personal

Información general del personal académico del programa. La información suministrada por persona:

- Nombre
- Correo electrónico
- Teléfono
- Educación

Ilustración 21 Pagina del personal académico del programa

9.7 Investigación

- Infraestructura
 - Nombre de la infraestructura (laboratorio)
 - Descripción
- Grupos
 - Nombre del grupo
 - Categoría Colciencias
 - Descripción
- Publicaciones
 - Título de la publicación
 - Referencia del autor
 - Año de publicación
 - URL opcional del archivo
 - Descripción corta
- Proyectos del programa
 - Título del proyecto
 - Descripción
 - Imagen del proyecto



Ilustración 22 Infraestructura del programa y Grupos de investigación del programa

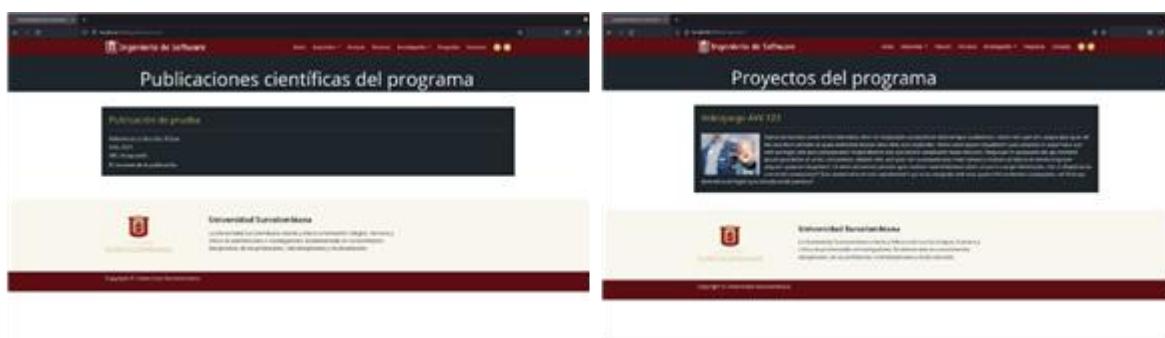


Ilustración 23 Publicaciones científicas del programa y Proyectos del programa del programa

9.8 Preguntas frecuentes

Preguntas frecuentes realizadas al programa

- Pregunta
- Respuesta

1. ¿Puedo adicionar y cancelar cursos?
Las dos (2) semanas previas al inicio de clase permiten adicionar o cancelar cursos según calendario colocado en la página de la USCO. Desde la semana uno (1) a la seis (6) el estudiante puede únicamente cancelar (No adicionar) cursos previo el pago de un valor en el Banco.

2. ¿Cuándo puedo cancelar el semestre?
Desde el primero hasta el último día del periodo académico el estudiante puede cancelar el semestre llenando el formato MI-FOR-FO-28 e incluyendo la firma de cada profesor que orienta los cursos. También debe explicar el motivo de la cancelación.

3. ¿Cómo ingreso al programa de Ingeniería de Software?
La Universidad Surcolombiana realiza dos (2) convocatorias de admisión en cada año. El Consejo Académico aprueba mediante Acuerdo el Instructivo de Admisiones, documento que se publica a través de la página Institucional www.usco.edu.co, el cual contiene la oferta académica, valor de la inscripción, las modalidades de inscripción, los cupos, fechas para tener en cuenta, documentos a presentar, ponderaciones, y en general las condiciones y requisitos para la inscripción, admisión e ingreso a la Universidad.

Ilustración 24 Preguntas frecuentes del programa

9.5 Contacto

Formulario de contacto directo con el programa el cual utiliza un correo intermedio para no vincular directamente el correo oficial. El correo destinatario es dinámico, depende del correo que tenga registrado el programa.

Contacto

Enviar mensaje

Nombre:

Correo:

Mensaje:

Enviar

DIRECCIÓN: Avenida Pastrana Borrero - Carrera 1 - Oficina 360 - Bloque Central - Neiva

TELÉFONO: 8754753 ext: 1039

CORREO: Ingenieria.software@usco.edu.co

HORARIO DE ATENCIÓN: 08:00 AM - 18:00 PM

Ilustración 25 Formulario de contacto del programa

9.6 Sistema de administración

La página principal (frontend) está administrada mediante un (backend) el cual tiene como ingreso la dirección “/login”.

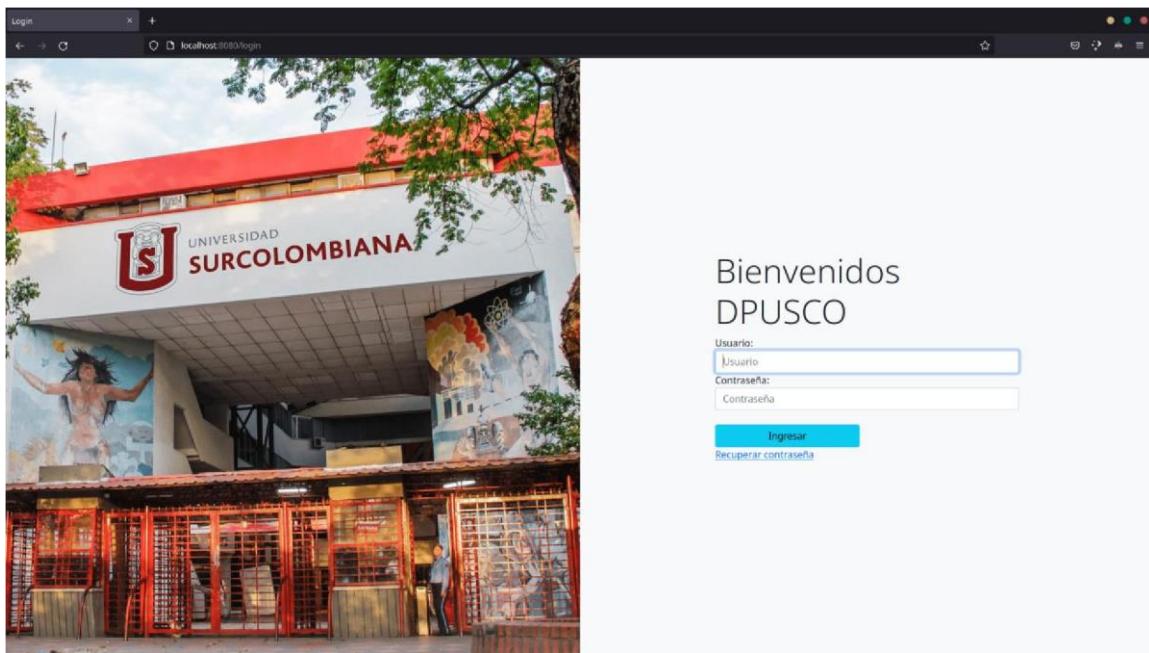


Ilustración 26 Página de ingreso al panel de administración del aplicativo

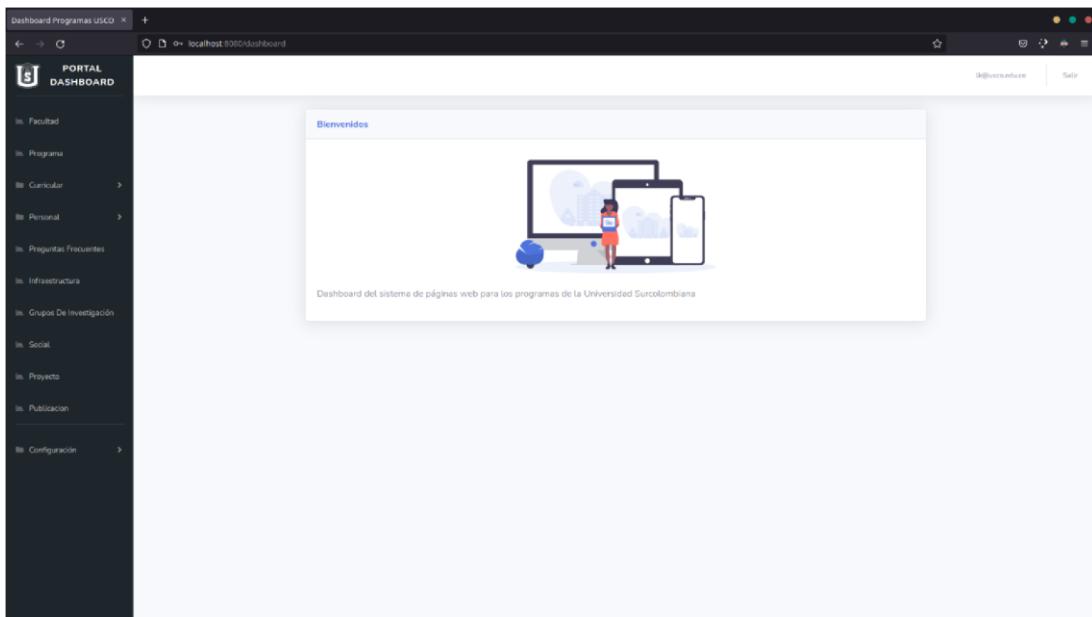


Ilustración 27 Página principal del portal de administración

9.7 Secciones del Dashboard

Tabla 9 Descripción de opciones en el Dashboard

Opción	Descripción
Facultad	CRUD

Programa	<ul style="list-style-type: none"> • Listado de programas por facultad • CRUD por programa
Curricular	Vínculo superior
• Agregar asignatura	<ul style="list-style-type: none"> • Creación de asignatura • CRUD de asignatura
• Lista curricular	<ul style="list-style-type: none"> • Lista de asignaturas por programa • Lista de programas por facultad
• Área de formación	<ul style="list-style-type: none"> • Área de formación de la asignatura • CRUD de Área
• Ntp	<ul style="list-style-type: none"> • Ntp de la asignatura • CRUD de Área
Personal	Vínculo superior
• Lista de personal	<ul style="list-style-type: none"> • Lista de personal por programa
• Educación personal	<ul style="list-style-type: none"> • Lista de títulos académicos de cada profesor • CRUD de personal
• Asignar jefe de programa	<ul style="list-style-type: none"> • CRUD de jefe de programa
Preguntas frecuentes	CRUD
Infraestructura	CRUD
Grupos de investigación	CRUD
Social	CRUD
Proyecto	CRUD
Publicación	CRUD
Configuración	CRUD
• Usuarios	Lista de usuarios del Dashboard

9.7.1 Mensajes de advertencia

La barra de navegación superior contiene un sistema de alerta que muestra mensajes relacionados con las acciones de la página actual. En la Ilustración 28 se observa una alerta por intento de borrado de una facultad, la cual tiene elementos asociados y una acción como esta obligaría a borrar todo en cascada (borrar todos los elementos asociados), o borrar el elemento único, lo que generaría inconsistencia en los datos.

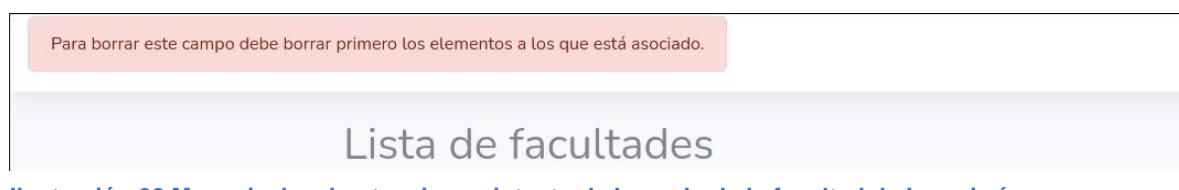


Ilustración 28 Mensaje de advertencia por intento de borrado de la facultad de Ingeniería

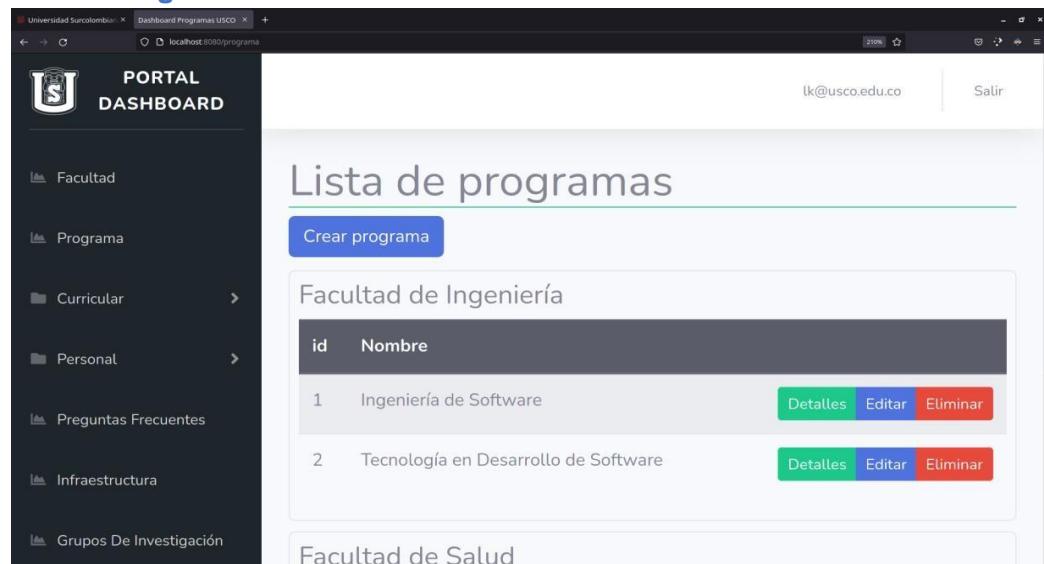
9.7.2 Facultad



The screenshot shows a web-based application interface for managing faculties. The left sidebar has a dark theme with white icons and text. It includes links for 'Facultad', 'Programa', 'Curricular', 'Personal', 'Preguntas Frecuentes', 'Infraestructura', and 'Grupos De Investigación'. The main content area has a light gray background. At the top, it says 'PORTAL DASHBOARD' with a logo. On the right, there are user details 'lk@usco.edu.co' and 'Salir'. The main title is 'Lista de facultades'. Below it is a blue button 'Crear facultad'. A search bar shows 'Mostrar 10 registros' and a 'Buscar:' input field. A table lists two faculties: 'Ingeniería' and 'Salud', each with 'Editar' and 'Eliminar' buttons. At the bottom, it says 'Mostrando registros del 1 al 2 de un total de 2 registros' and has navigation buttons for 'Anterior' (disabled), '1', and 'Siguiente'.

Ilustración 29 Menú CRUD de Facultades

9.7.3 Programa



The screenshot shows a web-based application interface for managing programs. The left sidebar has a dark theme with white icons and text. It includes links for 'Facultad', 'Programa', 'Curricular', 'Personal', 'Preguntas Frecuentes', 'Infraestructura', and 'Grupos De Investigación'. The main content area has a light gray background. At the top, it says 'PORTAL DASHBOARD' with a logo. On the right, there are user details 'lk@usco.edu.co' and 'Salir'. The main title is 'Lista de programas'. Below it is a blue button 'Crear programa'. A section titled 'Facultad de Ingeniería' contains a table with two program entries: 'Ingeniería de Software' (id 1) and 'Tecnología en Desarrollo de Software' (id 2). Each entry has 'Detalles', 'Editar', and 'Eliminar' buttons. Another section titled 'Facultad de Salud' is partially visible. At the bottom, it says 'Mostrando registros del 1 al 2 de un total de 2 registros' and has navigation buttons for 'Anterior' (disabled), '1', and 'Siguiente'.

Ilustración 30 Menú CRUD de Programas

9.7.4 Curricular

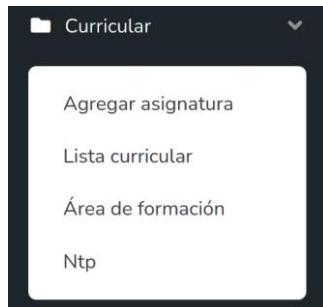


Ilustración 31 Menú superior desplegable para la sección Curricular

Este es un formulario para agregar una asignatura. Los campos incluyen: 'Asignatura' (campo de texto), 'Semestre' (campo de texto), 'Créditos' (campo de texto), 'Nivel Académico' (selección desplegable con opción 'Teórico'), 'Programa' (selección desplegable con opción 'Ingeniería de Software'), 'Áreas' (selección desplegable), 'Ntp' (campo de texto) y botones 'Cancelar' y 'Agregar asignatura'.

Ilustración 32 Formulario de creación de asignatura

Este es un submenú de la lista curricular. Muestra dos secciones: 'Facultad de Ingeniería' y 'Facultad de Salud'. Cada sección tiene una tabla con columnas 'id' y 'Nombre'. Los botones 'Curricular' están ubicados en la parte derecha de cada fila. La tabla para la Facultad de Ingeniería muestra dos filas: '1. Ingeniería de Software' y '2. Tecnología en Desarrollo de Software'. La tabla para la Facultad de Salud muestra una sola fila: '3. Medicina'.

Ilustración 33 Submenú de lista curricular por programa

Universidad Suroeste - PORTAL DASHBOARD

Lista curricular - Ingeniería de Software

Crear curricular

Semestre 1

Asignatura	Créditos	Carácter	Área	Ntp
Lectura de textos técnicos en inglés	3	Técnico		Leer Eliminar
Introducción a la ingeniería	3	Técnico		Leer Eliminar
Fundamentos de programación	3	Técnico - Práctico		Leer Eliminar
Fundamentos de Matemáticas	3	Técnico		Leer Eliminar
Comunicación Lingüística	2	Técnico		Leer Eliminar
Computación Física	3	Técnico - Práctico		Leer Eliminar
Algorítmica de programación	3	Técnico		Leer Eliminar

Semestre 2

Asignatura	Créditos	Carácter	Área	Ntp
Proyecto Integrador I	4	Técnico - Práctico		Leer Eliminar
Programación Orientada a Objetos	4	Técnico - Práctico		Leer Eliminar
Cálculo Diferencial	4	Técnico		Leer Eliminar
Arquitectura de Hardware	4	Técnico - Práctico		Leer Eliminar
Álgebra Lineal	4	Técnico		Leer Eliminar

Semestre 3

Asignatura	Créditos	Carácter	Área	Ntp
				Leer Eliminar

Ilustración 34 CRUD del currículo de Ingeniería de Software

Universidad Suroeste - PORTAL DASHBOARD

Lista de área de formación

Crear área de formación

No hay área de formación registradas

Mostrar 10 registros Buscar:

Nombre
Ningún dato disponible en esta tabla

Mostrando registros del 0 al 0 de un total de 0 registros Anterior Siguiente

Ilustración 35 CRUD para Área de formación por asignatura

Ilustración 36 CRUD para NTP por asignatura

9.7.5 Personal

Ilustración 37 Menú superior desplegable para la sección Personal

Ilustración 38 Submenú de lista de personal por programa

Imagen	Nombre	Correo	Celular	Cargo	Programa	Deshabilitar
	Esaú Silva Lara	gsv546s@usco.edu.co		Profesor de planta	Ingeniería de Software	Habilitado Editar Eliminar
	Eurípides Triana Tacuma	2sda@usco.edu.co		Profesor de catedra	Ingeniería de Software	Habilitado Editar Eliminar
	Ferley Medina Rojas	gfc4f@usco.edu.co		Profesor de catedra	Ingeniería de Software	Habilitado Editar Eliminar
	Fernando Rojas Rojas	asf3as@usco.edu.co	3004513252	Jefe de programa	Ingeniería de Software	Habilitado Editar Eliminar
	John Felipe Vargas Pardomo	ytrdgfh@usco.edu.co		Profesor de planta	Ingeniería de Software	Habilitado Editar Eliminar
	John Jairo Trujillo	av4hj@usco.edu.co		Profesor de	Ingeniería de	Habilitado Editar Eliminar

Ilustración 39 CRUD del personal de Ingeniería de Software

Facultad de Ingeniería		
id	Nombre	
1	Ingeniería de Software	Asignar jefe de programa
2	Tecnología en Desarrollo de Software	Asignar jefe de programa

Facultad de Salud		
id	Nombre	
3	Medicina	Asignar jefe de programa

Ilustración 40 Submenú de asignación de jefe por programa

Imagen	Nombre	
	Esaú Silva Lara	Asignar jefe de programa
	Eurípides Triana Tacuma	Asignar jefe de programa
	Ferley Medina Rojas	Asignar jefe de programa
	Fernando Rojas Rojas	Asignar jefe de programa

Ilustración 41 CRUD para jefe de programa

9.7.6 Preguntas frecuentes

Pregunta	Respuesta	Programa	Acciones
¿Cuándo puedo cancelar el semestre?	Desde el primero hasta el último día del período académico el estudiante puede cancelar el semestre llenando el formato MI-FOR-FO-28 e incluyendo la firma de cada profesor que orienta los cursos. También debe explicar el motivo de la cancelación.	Ingeniería de Software	Editar Eliminar
¿Cómo ingreso al programa de Ingeniería de Software?	La Universidad Surcolombiana realiza dos (2) convocatorias de admisión en cada año. El Consejo Académico aprueba mediante Acuerdo el Instructivo de Admisiones, documento que se publica a través de la página Institucional, www.usco.edu.co, el cual contiene la oferta académica, valor de la inscripción, las modalidades de inscripción, los cupos, fechas para tener en cuenta, documentos a presentar, ponderaciones, y en general las condiciones y requisitos para la inscripción, admisión e ingreso a la Universidad.	Ingeniería de Software	Editar Eliminar
¿Cómo valido un curso?	Valida el estudiante que tiene conocimientos, habilidades y experiencia en los contenidos de un curso. La validación requiere un examen desarrollado y supervisado por profesores del programa. Finalizando el semestre, el estudiante debe llenar el formato MI-FOR-FO-28 se entrega en el programa y el Consejo de Facultad define las fechas de los exámenes de validación. Luego las notas se actualizan en Registro y Control.	Ingeniería de Software	Editar Eliminar
¿Es posible rectificar una nota?	Las notas las puede rectificar el docente en la semana siguiente a la fecha de corte de notas o enviando la corrección en el formato MI-FOR-FO-28, firmada por el estudiante, el docente, el consejero y el jefe de Programa.	Ingeniería de Software	Editar Eliminar

Ilustración 42 CRUD para preguntas frecuentes

9.7.7 Infraestructuras

Foto	Nombre	Enfoque	Descripción	Programas	Acciones
	KAVAR	Data Science	esse id id ullamco sunt nostrud occaecat ut adipisciing molit molit cupidatat do occaecat anim ad fugiat ut pariatur duis adipisciing ex elit et ex labore culpa quis enim anim adipisciing fugiat dolor occaecat irure veniam deserunt consequat id ea molit tempor molit cupidatat ullamco excepteur adipisciing dolor sit anim laboris officia sint adipisciing dolor occaecat illum non quis illum Lorem quis aliqua nostrud incididunt esse id et proident commodo fugiat non nostrud incididunt minim duis minim cupidatat ex eu molit occaecat veniam proident consequat irure aute exercitation sunt adipisciing nisi labore labore non ea labora ut aliquip ipsum anim	• Ingeniería de Software • Tecnología en Desarrollo de Software	Borrar programa asociado Editar Eliminar
	KYOTO	Programación Web	tempor ipsum voluptate duis consectetur dolor eiusmod exercitation id cupidatat velit consectetur magna deserunt Lorem culpa eiusmod duis consequat irure non ut velit elit consequat et non voluptate cillum duis magna pariatur labore illum sint enim nisi labore labore non ea labore labore non ea labora ut aliquip ipsum anim	• Tecnología en Desarrollo de Software	Borrar programa asociado Editar Eliminar

Ilustración 43 CRUD para infraestructuras

9.7.8 Grupos de investigación

Foto	NombreDelGrupo	Categoría	Descripción	Programa
	GameSoft	A	<p><p>El semillero en desarrollo de video juegos pretende llevar a cabo proyectos de integración de orden interdisciplinario, teniendo en cuenta el enfoque de entretenimiento y científico.&nbsp; </p><p>Dentro de la línea de entretenimiento se profundiza en las diferentes categorías de jugabilidad teniendo en cuenta el fundamento de la aplicación y el nivel de interactividad, enfocándonos a diseño de guiones temáticos, con un factor de estética hiper realista dentro de los géneros Beat 'em up, Arcade, Plataformas, Shoot 'em up, Survival. De orden científicos tendremos la simulación de alta ingeniería.</p>	Ingeniería de Software
	La Colonia	A	<p><p>Tiene como objetivo fomentar la investigación para desarrollar procesos que permitan la construcción de soluciones eficientes en las áreas de la informática y difundir el uso de las Nuevas Tecnologías de la Información y Comunicación (NTIC). El grupo desarrolla las líneas de investigación del programa: Ingeniería de</p>	Ingeniería de Software

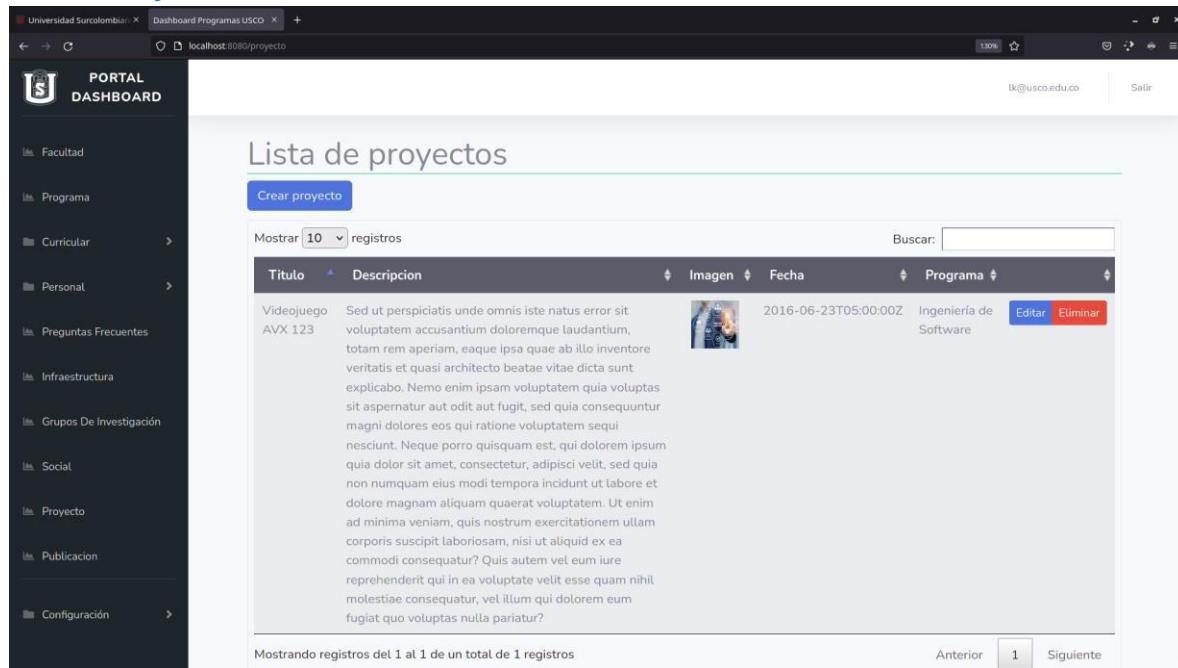
Ilustración 44 CRUD grupos de investigación

9.7.9 Social

Usuario	Facebook	Instagram	Youtube	Whatsapp	Linkedin	Twitter
facebjuancastro						
Fernando Rojas	fbb/FRR					
Ingeniería de Software	https://www.facebook.com/people/IngenieriaSoftware-USCO/100057314783971/	https://www.instagram.com/fac_ingenieriausco/				

Ilustración 45 CRUD para redes sociales

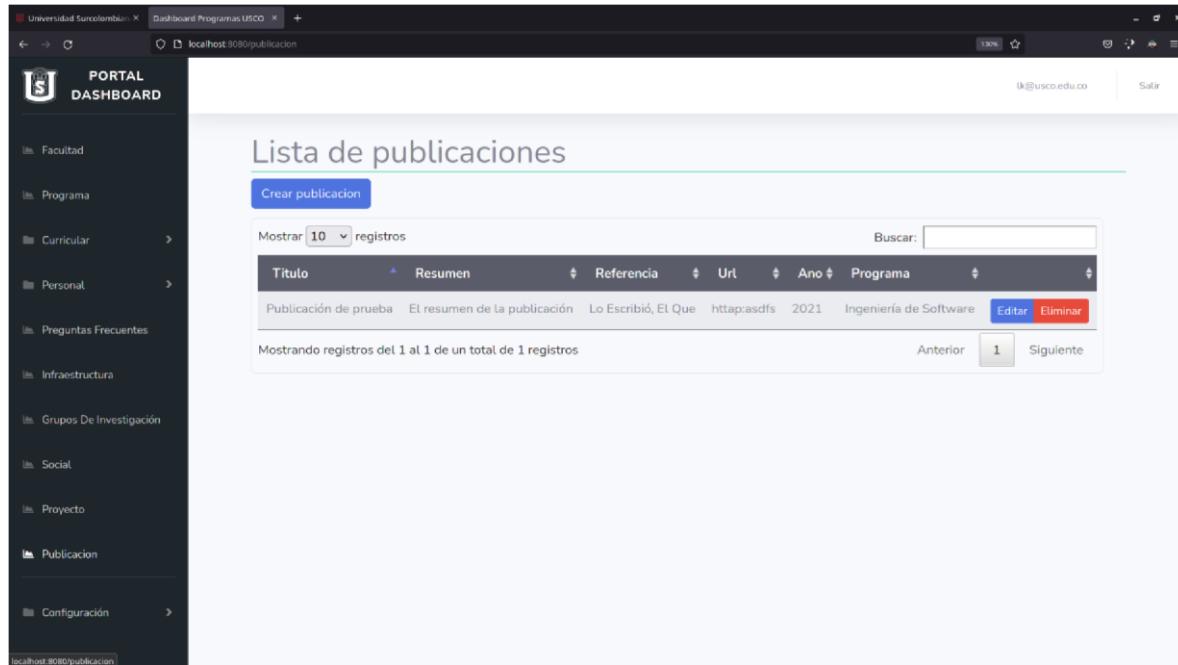
9.7.10 Proyectos



Titulo	Descripción	Imagen	Fecha	Programa
Videojuego AVX 123	Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?		2016-06-23T05:00:00Z	Ingeniería de Software

Ilustración 46 CRUD para proyectos

9.7.11 Publicaciones



Titulo	Resumen	Referencia	Url	Año	Programa
Publicación de prueba	El resumen de la publicación	Lo Escribió, El Que	http://asdfs	2021	Ingeniería de Software

Ilustración 47 CRUD para publicaciones

9.7.12 Usuarios

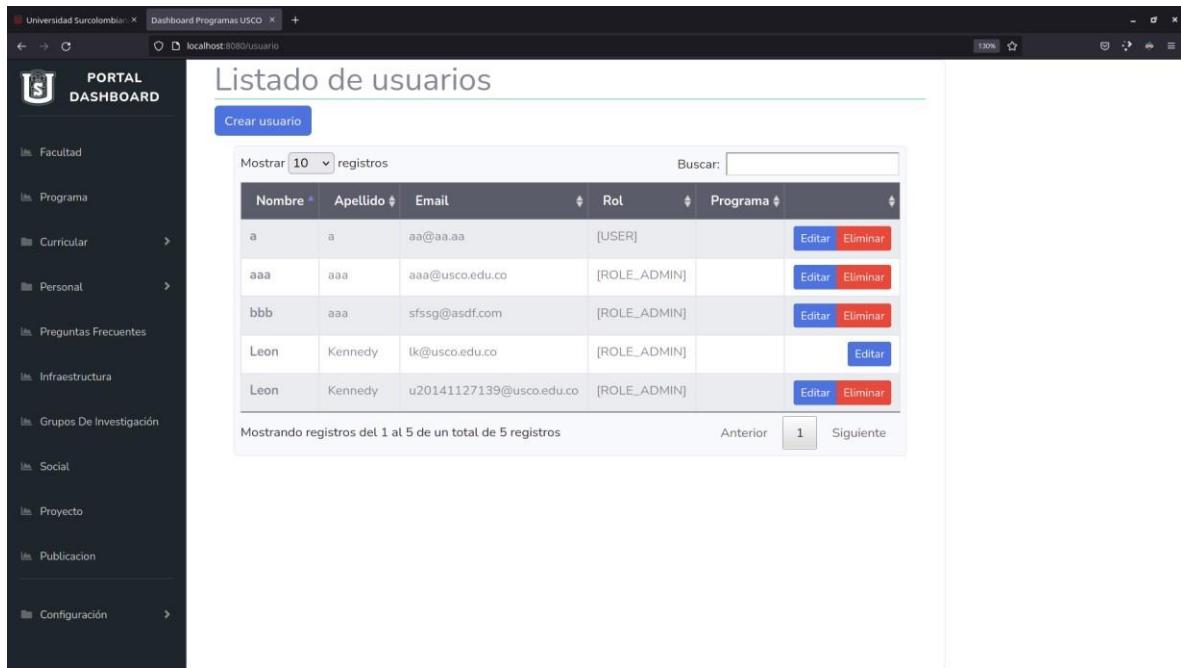
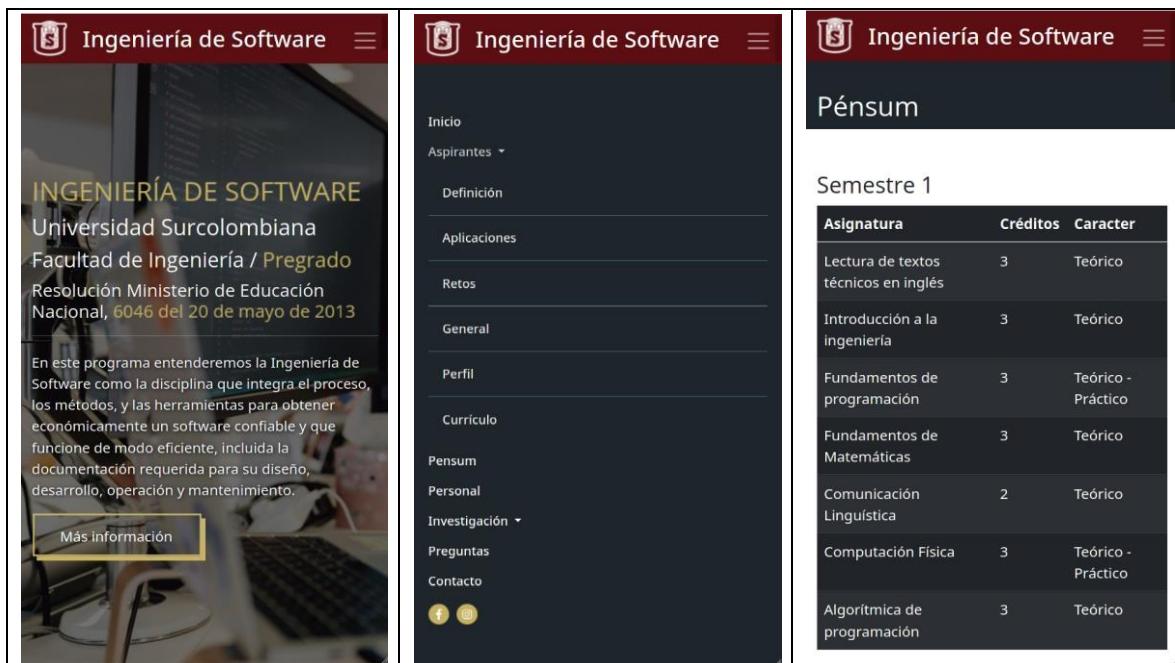


Ilustración 48 CRUD para usuarios

9.8 Diseño responsivo

La plantilla desarrollada cuenta con un sistema adaptable a dispositivos móviles, lo cual es un requerimiento actual muy importante para el consumo de información.

Tabla 10 Capturas de diseño responsivo en dispositivos móviles



S Ingeniería de Software ≡

Pénum

Semestre 1

Asignatura	Créditos	Carácter
Lectura de textos técnicos en inglés	3	Teórico
Introducción a la ingeniería	3	Teórico
Fundamentos de programación	3	Teórico - Práctico
Fundamentos de Matemáticas	3	Teórico
Comunicación Lingüística	2	Teórico
Computación Física	3	Teórico - Práctico
Algorítmica de programación	3	Teórico

S Ingeniería de Software ≡

Personal Académico

Jefe del programa



S Ingeniería de Software ≡

Infraestructura



KAVAR

Enfoque: Data Science
 esse id id ullamco sunt nostrud occaecat ut
 adipisicing mollit mollit cupidatat do occaecat
 anim ad fugiat ut pariatur quis enim anim
 adipisicing fugiat dolor occaecat irure veniam
 deserunt consequat id ea mollit tempor mollit
 cupidatat ullamco excepteur adipisicing dolor
 sit anim laboris officia sint adipisicing dolor

10 Implementación

10.1 Patrones de diseño

Los patrones de diseño son soluciones habituales a problemas que ocurren con frecuencia en el diseño de software. Son como planos prefabricados que se pueden personalizar para resolver un problema de diseño recurrente en el código. No se puede elegir un patrón y copiarlo en el programa como si se tratara de funciones o bibliotecas ya preparadas. El patrón no es una porción específica de código, sino un concepto general para resolver un problema particular (Refactoring.Guru, 2021).

10.1.1 MVC

MVC (Modelo-Vista-Controlador) es un patrón en el diseño de software comúnmente utilizado para implementar interfaces de usuario, datos y lógica de control. Enfatiza una separación entre la lógica de negocios y su visualización. Esta "separación de preocupaciones" proporciona una mejor división del trabajo y una mejora de mantenimiento (Mozilla, 2021).

El modelo está representado por las entidades individuales dentro del paquete de dominio.

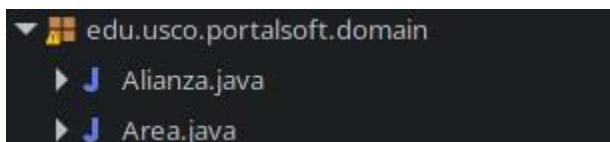


Ilustración 49 Ejemplo de clases de dominio

La vista está representada por los diferentes archivos de visualización, como por ejemplo html5, CSS, etc., que son usados para imprimir la información en pantalla.

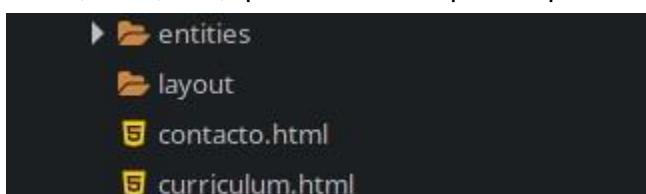


Ilustración 50 Ejemplo de archivos de visualización

El controlador en Spring es representado mediante la etiqueta `@Controller` y por convención general se nombra uniendo el nombre de la entidad y la palabra `Controller`, aunque no es obligatorio y en ningún caso su funcionamiento se ve afectado por el nombre, siempre y cuando respete las reglas de Java.



Ilustración 51 Ejemplo de archivos controladores

10.1.2 Fachada

Una fachada es una clase que proporciona una interfaz simple a un subsistema complejo que contiene muchas partes móviles. Una fachada puede proporcionar una funcionalidad limitada en comparación con trabajar directamente con el subsistema. Sin embargo, tan solo incluye las funciones realmente importantes para los clientes (Refactoring.Guru, 2021).

En lugar de hacer que el código trabaje con decenas de las clases del framework directamente, se crea una clase fachada que encapsula esa funcionalidad y la esconde del resto del código. Esta estructura también ayuda a minimizar el esfuerzo de actualizar a futuras versiones del framework o de sustituirlo por otro. Lo único que se tendría que cambiar en la aplicación es la implementación de los métodos de la fachada (Refactoring.Guru, 2021).

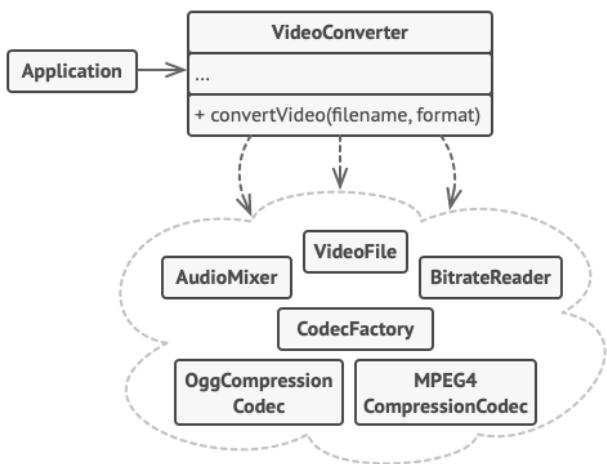


Ilustración 52 Ejemplo de aislamiento de múltiples dependencias dentro de una única clase fachada

Para implementar este patrón se creó tres paquetes con sus respectivas clases o interfaces:

- Repository (interfaces) o Se encarga de administrar las operaciones CRUD de las entidades mediante JpaRepository
- Service (interfaces) o Establece los métodos que podrán ser vistos libremente mediante la creación de un objeto tipo “NombreDeEntidadService”.
- Service.impl (clases) o Establece como serán implementadas las interfaces del paquete Service.

```

▽ edu.usco.portalsoft.repository
  > IAlianzaRepository.java
  > IAreaRepository.java
  > ICargoRepository.java

```

Ilustración 53 Ejemplo del paquete repository

```

▽ edu.usco.portalsoft.service
  > EmailService.java
  > IAlianzaService.java
  > IAreaService.java
  > ICargoService.java

```

Ilustración 54 Ejemplo del paquete service

```

▽ edu.usco.portalsoft.service.impl
  > AlianzaServiceImpl.java
  > AreaServiceImpl.java
  > CargoServiceImpl.java

```

Ilustración 55 Ejemplo del paquete service.impl

10.2 Paquetes

La siguiente es la estructura jerárquica de paquetes del programa.

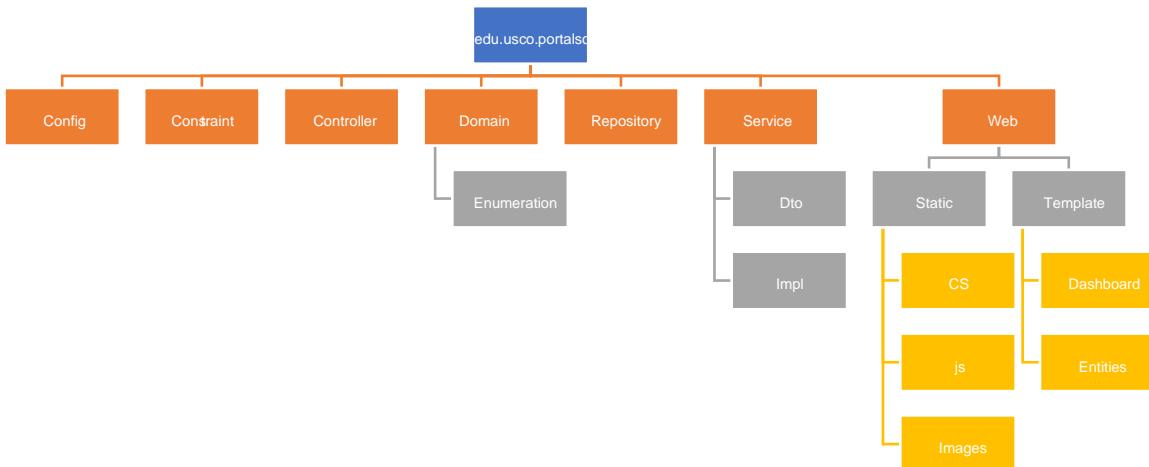


Ilustración 56 Estructura jerárquica de paquetes del programa

10.3 Controlador principal IndexController

La implementación del IndexController se lleva a cabo por medio de la anotación @Controller donde se encarga de redireccionar los diferentes métodos de petición HTML.

```
import org.springframework.beans.factory.annotation.Autowired;□
@Controller
public class IndexController {
```

Ilustración 57 Clase inicial asignada como controladora mediante la etiqueta @Controller

10.4 Validación de datos

Para la validación de los datos se usa las anotaciones de Java EE mediante la API Bean Validation del paquete “javax.validation”.

El modelo de validación de Bean está respaldado por restricciones en forma de anotaciones colocadas en un campo, método o clase de un componente JavaBeans, como un bean administrado. Las restricciones pueden ser incorporadas o definidas por el usuario. Las restricciones definidas por el usuario se denominan restricciones personalizadas. Varias restricciones integradas están disponibles en el paquete javax.validation.constraints (Oracle, 2014).

```
import javax.validation.constraints.Min;
import javax.validation.constraints.NotEmpty;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Pattern;
```

Ilustración 58 Ejemplo de librerías javax.validation importadas para ejecutar la validación de campos de entrada

```
@Entity
@Table(name = "programa")
@Cache(usage = CacheConcurrencyStrategy.READ_WRITE)
public class Programa implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @SequenceGenerator(name = "sequenceGenerator")
    private Long id;

    @NotNull
    @NotEmpty
    @Column(name = "nombre", nullable = false)
    private String nombre;
```

Ilustración 59 Anotaciones de JavaEE para la respectiva validación de datos de entrada

Cuando Thymeleaf realiza una consulta POST y captura los datos de un formulario dentro de un objeto declarado, estos valores son verificados a través de las anotaciones del validador. Realizar este proceso solo garantiza la detección de datos inválidos según su respectivo campo para preservar la estabilidad de la base de datos, pero esto no previene errores de consola y colapso de la aplicación, para ello es necesario implementar captura de errores mediante sentencias de captura de excepciones tales como “try/catch”.

```

@RequestMapping(value = "/eliminar/{id}")
public String eliminar(@PathVariable(value = "id") Long id, RedirectAttributes flash) {
    try {
        if (id > 0) {
            programaService.delete(programaService.findById(id));
        }
    } catch (NullPointerException e) {
    } catch (Exception e) {
        flash.addFlashAttribute("constraint",
                "Para borrar este campo debe borrar primero los elementos a los que está asociado.");
    }
    return "redirect:/programa";
}

```

Ilustración 60 Ejemplo de implementación Try/Catch para capturar errores de excepción

Un punto importante es que, aunque las anotaciones no previenen el colapso de la aplicación por errores de lógica, le permite a Thymeleaf establecer un elemento de validación dentro de los formularios al no cumplirse los parámetros establecidos por las anotaciones. Para implementar la validación en la vista, es necesario utilizar la anotación “@Valid” seguida del objeto a capturar en POST.

```

@RequestMapping(value = "/form", method = RequestMethod.POST)
public String guardar(@Valid Personal personal, BindingResult result, Model model,
SessionStatus status, {
    model.addAttribute("personales", personalService.findAll());
    ((Model) model).addAttribute("cargos", cargoService.findAll());
    ((Model) model).addAttribute("programas", programaService.findAll());
}

```

Ilustración 61 Implementación de la etiqueta @Valid en el método de consulta de tipo POST

En la Ilustración 62 se aprecia como al tratar de guardar un formulario vacío la aplicación HTML5 a cargo de Thymeleaf impide la operación que está anotada con “@Valid” (Ilustración 61) en el controlador y “@NotEmpty”, “@Pattern(regexp = “^[@\\s]+@[\\s]+\\.[^@\\s]+\$”) en la clase Personal.



Ilustración 62 Advertencias de validación

A través de Thymeleaf se añadieron mensajes de error en la ventana de navegación, en la Ilustración 62 se aprecia como aparece un mensaje de advertencia al intentar borrar la facultad de ingeniería ya que esta se encuentra unida a múltiples elementos, lo que causaría un error en cascada dentro de la base de datos.

Para borrar este campo debe borrar primero los elementos a los que está asociado.

Lista de facultades

Crear facultad

Ilustración 63 Mensaje de error ya que se impide el borrado en cascada

10.5 Repository

Para implementar el patrón repositorio se creó el paquete “Repository” el cual contiene las interfaces extendidas de la interfaz “JpaRespository” propio de Spring, la cual otorga un gran número de consultas CRUD prediseñadas y comunes. Esta solución ahorra mucho tiempo de desarrollo y establece un punto único de partida para crear nuevas consultas personalizadas que establecen el “que hacer” pero no el “cómo hacerlo”.

Se trató de respetar los siguientes tres ideales lo mejor posible para crear así repositorios entendibles y sostenibles en el tiempo:

- Nombre de repositorio claro
- No revelación de elementos de lógica del negocio en los argumentos del repositorio
- Solo un tipo de objeto por repositorio

10.6 Spring Security

Spring Security se encarga de autenticar y administrar las reglas de autenticación.

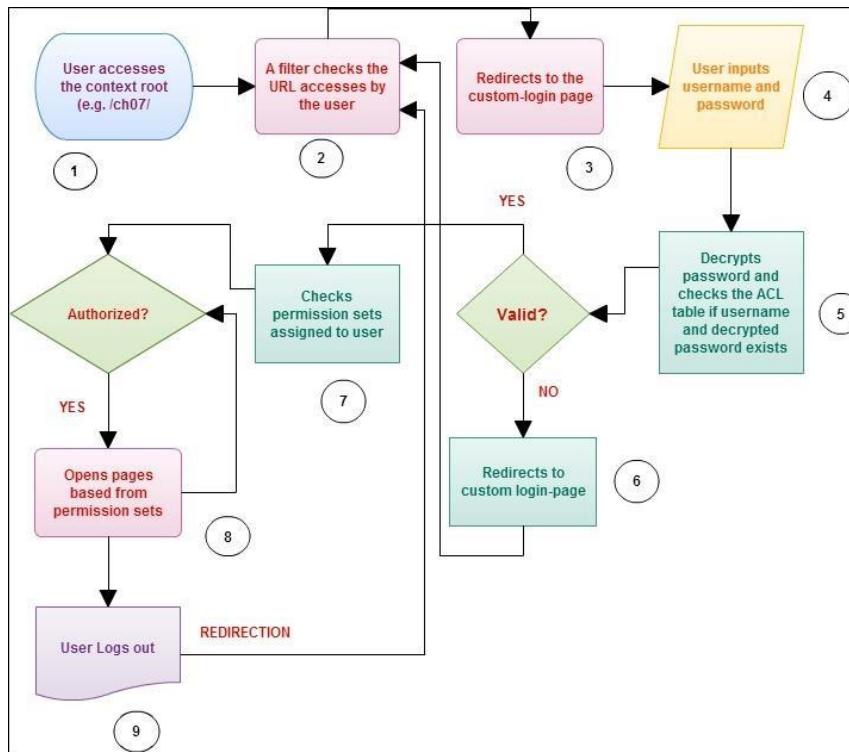


Ilustración 64 Diagrama de flujo de verificación de seguridad mediante Spring Security (Sherwin & Calleja , 2016)

10.6.1 Usuarios y roles

Para crear usuarios y roles es necesario crear dos clases que almacenen dichos objetos. El nombre de clase usuario o rol es indiferente, ya que la conexión con Spring Security se realizará en una interfaz de usuario que hereda de la interfaz `UserDetailsService`.

```
@Entity
@Table(name = "users", uniqueConstraints = @UniqueConstraint(columnNames = "email"))
public class User implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    @NotEmpty
    private String firstName;

    @NotEmpty
    private String lastName;

    @NotEmpty
    @Email
    private String email;

    @NotEmpty
    private String password;
}
```

Ilustración 65 Clase de usuarios para Spring Security

```
@Entity
public class Role {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    private String name;

    public Role() {
    }
}
```

Ilustración 66 Clase de roles para Spring Security

10.6.2 Repositorio de usuarios

Para obtener los usuarios, utilizaremos una etiqueta `@Service` que se conectará a un repositorio que implementa la interfaz JPA para acceder a nuestra base de datos. Para que el Service funcione, tiene que estar anotado como `@Service` y debe implementar la interfaz `UserDetailsService`. La interfaz `UserDetailsService` se utiliza para recuperar datos relacionados con el usuario. Tiene un método llamado `loadUserByUsername()` que se puede sobrescribir para personalizar el proceso de búsqueda del usuario.

```

@Repository
public interface UserRepository extends JpaRepository<User, Long> {
    User findByEmail(String mail);

    public User findByResetPasswordToken(String token);

    User findByToken(String token);

```

Ilustración 67 Repositorio de Usuario con herencia de JpaRepository

10.6.3 Interfaz de UserService

Se crea una interfaz que hereda de la interfaz UserDetailsService, propia de Spring, para localizar al usuario.

```

import org.springframework.security.core.userdetails.UserDetailsService;

public interface UserService extends UserDetailsService {

    User findByEmail(String email);

    User save(UserRegistrationDto registration);

    public void updateResetPasswordToken(String token, String email);

    public User getByResetPasswordToken(String token);

    public void updatePassword(User user, String newPassword);

```

Ilustración 68 Interfaz UserService que hereda de UserDetailsService

10.6.4 Clase de servicio UserServiceImpl

Esta clase implementa los métodos en UserService y será la encargada de llevar cabo las operaciones de manipulación de los usuarios registrados en el sistema.

```

@Service
public class UserServiceImpl implements UserService {

    @Autowired
    private UserRepository userRepository;

    @Autowired
    private BCryptPasswordEncoder passwordEncoder;

    private static final long EXPIRE_TOKEN_AFTER_MINUTES = 30;

    @Override
    public UserDetails loadUserByUsername(String email) throws UsernameNotFoundException {
        User user = userRepository.findByEmail(email);
        if (user == null){
            throw new UsernameNotFoundException("Invalid username or password.");
        }
        return new org.springframework.security.core.userdetails.User(user.getEmail(),
                user.getPassword(),
                mapRolesToAuthorities(user.getRoles()));
    }

    public User findByEmail(String email){
        return userRepository.findByEmail(email);
    }

```

Ilustración 69 Clase de tipo servicio para la implementación de UserService

10.6.5 Habilitar la configuración de seguridad

Luego de añadir el repositorio de seguridad, es necesario crear el archivo de configuración de seguridad de Spring Security. El nombre del método configureGlobal no es importante. Sin embargo, es importante configurar AuthenticationManagerBuilder solo en una clase anotada con @EnableWebSecurity, @EnableGlobalMethodSecurity o @EnableGlobalAuthentication. Hacer lo contrario tiene resultados impredecibles (Granja, 2021).

```
@Configuration
@EnableWebSecurity
public class SecurityConfiguration extends WebSecurityConfigurerAdapter {

    @Autowired
    private UserService userService;

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.csrf().disable();
        http
    }
}
```

Ilustración 70 Ejemplo de archivo de configuración de seguridad en Spring Security

10.6.6 Dialecto de seguridad

El dialecto de seguridad de Spring nos permite mostrar contenido o enlaces de forma condicional en función de los roles del usuario. La etiqueta principal disponible con Spring Security Thymeleaf es la etiqueta <sec: authorize *>. La etiqueta <sec: authorize *> puede tomar expresión y mostrará el contenido de la página de forma condicional (Sharma, 2020).

10.6.7 Codificación de contraseña

Spring Security nos permite asignar un codificador de contraseña seguro a nuestro objeto UserDetail. De forma predeterminada, utiliza BCrypt para cifrar las contraseñas, que se considera un algoritmo completo para codificar contraseñas. También es posible establecer el número de rondas hash (o la fuerza como sugiere el nombre del parámetro) y la implementación segura del algoritmo aleatorio que se utilizará en el proceso (Erinç, 2021).

```
@Bean
public BCryptPasswordEncoder passwordEncoder(){
    return new BCryptPasswordEncoder();
}

@Bean
public DaoAuthenticationProvider authenticationProvider(){
    DaoAuthenticationProvider auth = new DaoAuthenticationProvider();
    auth.setUserDetailsService(userService);
    auth.setPasswordEncoder(passwordEncoder());
    return auth;
}

@Override
protected void configure(AuthenticationManagerBuilder auth) throws Exception {
    auth.authenticationProvider(authenticationProvider());
}
```

Ilustración 71 Método de encriptado de usuarios mediante BCrypt

10.6.8 Reglas de acceso URL

La biblioteca de etiquetas de seguridad de Spring también brinda la capacidad de controlar el contenido según las reglas de autorización de URL existentes definidas en nuestro archivo de configuración de seguridad. Podemos hacer esto con el uso de los métodos authorizeRequests () y antMatchers () (Sharma, 2020).

```
.antMatchers("/programa/**").hasAuthority("ROLE_ADMIN")
.antMatchers("/facultad/**").hasAuthority("ROLE_ADMIN")
.antMatchers("/programa/**").hasAuthority("ROLE_ADMIN")
.antMatchers("/curricular/**").hasAuthority("ROLE_ADMIN")
.antMatchers("/area/**").hasAuthority("ROLE_ADMIN")
.antMatchers("/ntp/**").hasAuthority("ROLE_ADMIN")
```

Ilustración 72 Ejemplo de bloqueo de URL mediante la regla de autorización "hasAuthority"

10.6.9 Dialecto de Spring Security

En entornos Spring MVC, el módulo de integración Spring Security funciona como un reemplazo del taglib de seguridad Spring. Usamos este dialecto en el ejemplo para imprimir las credenciales del usuario registrado y mostrar contenido diferente a roles diferentes. El atributo sec: authorize muestra su contenido cuando la expresión del atributo se evalúa como verdadera (Thymeleaf.org, 2021):

```
<div sec:authorize="isAuthenticated()>
    This content is only shown to authenticated users.
</div>
<div sec:authorize="hasRole('ROLE_ADMIN')>
    This content is only shown to administrators.
</div>
<div sec:authorize="hasRole('ROLE_USER')>
    This content is only shown to users.
</div>
```

Ilustración 73 Etiquetas de autorización en la vista según el rol de usuario

10.7 Thymeleaf

El motor de plantilla permite procesar documentos html y datos de modelo para generar un documento de salida, esta habilidad es la que permite generar un programa o sitio web dinámico.

La estructura de un sistema de plantillas se divide en tres partes principales:

- Modelo de datos o Contiene la lógica del negocio y es la que genera la nueva información que pueda solicitar el usuario.
- Plantilla o Es un esquema basado en múltiples tecnologías html5 que mediante etiquetas específicas de Thymeleaf, comprende como procesar y generar un archivo nuevo para la vista.
- Documento de salida o Es el elemento resultante con la nueva información, aquel que ve el usuario en el navegador.

10.7.1 Dependencia Thymeleaf

Para usar el dialecto Thymeleaf con Spring Security en la aplicación web, se necesita incluir el módulo Spring Security con Thymeleaf en la aplicación dentro del archivo “pom.xml”.

```
<dependency>
    <groupId>org.thymeleaf.extras</groupId>
    <artifactId>thymeleaf-extras-springsecurity5</artifactId>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-mail</artifactId>
</dependency>
```

Ilustración 74 Dependencia de Spring Security para Maven

10.7.2 Variables y mensajes

En los controladores se crean variables mediante la asociación con modelos. El modelo es básicamente una clase POJO (Plain Old Java Object) creada para vincular campos de formulario con propiedades del objeto. Este objeto se colocará en el modelo (objeto modelo).

```
@RequestMapping(value = "")
public String ver(Model model) {
```

Ilustración 75 Argumento modelo utilizado para entregar valores

En el controlador se crean variables que podrá leer Thymeleaf mediante el método “model.addAttribute” y dentro de este se establecen dos argumentos, el primero, el nombre de la variable entre comillas dobles; el segundo, el valor de la variable.

```
@RequestMapping(value = "")
public String ver(Model model) {

    model.addAttribute("titulo", "Lista de programas");
    model.addAttribute("update", "crear programa");
    model.addAttribute("empty", "No hay programas registradas");

    model.addAttribute("programas", programaService.findAll());
    model.addAttribute("facultades", facultadService.findAll());
```

Ilustración 76 Variables establecidas en el método model.addAttribute

Dentro de la plantilla se usan etiquetas de tipo Thymeleaf, estas comienzan siempre con “th:”. En la Ilustración 76 se aprecia como se está haciendo un recorrido de tipo bucle “para”, representado por “th:each”, donde se recorren los valores que puedan existir en el arreglo “programas” y cada uno de ellos es almacenado en la variable “programa” para

luego ser utilizado en las siguientes etiquetas HTML y así poder extraer la información asociada al objeto “programa”.

```
<tr th:each="programa: ${programas}"  
     th:if="${programa?.facultad?.id == facultad?.id}">  
    <td th:text="${programa.id}"></td>  
    <td th:text="${programa.nombre}"></td>  
    <td th:if="${ocultar != true}" class="text-right">  
        <div class="btn-group">
```

Ilustración 77 Ejemplo de uso de variable en Thymeleaf para hacer el recorrido de un arreglo

10.7.3 Fragmentos

Thymeleaf permite crear una plantilla base la cual provee un diseño común en las diferentes páginas; para ello se utiliza la etiqueta “th:fragment=“nombreDelFragmento” para declarar el fragmento global, y con la etiqueta “th:replace=“nombreDelArchivoOriginal :: nombreDelFragmento” se establece el uso de dicho fragmento.

```
<!-- Footer -->  
<footer th:fragment="footer">  
    <section class="footer text-light">  
        <div class="container">
```

Ilustración 78 Declaración del fragmento Footer

```
<!-- Footer -->  
<footer th:replace="index :: footer"></footer>
```

Ilustración 79 Implementación del fragmento Footer

10.8 Estilo CSS y JS

Como se estableció anteriormente, la plantilla está prediseñada y contiene múltiples tipos de archivos CSS y JS, esto aceleró el proceso de creación de la solución de software, pero introdujo nuevos problemas, como son el alto acoplamiento al diseño, lógica y componentes, por este motivo fue necesario invertir tiempo modificando algunos elementos clave en estos archivos para alcanzar el estilo estético aceptado por la Universidad Surcolombiana. Aún con todos los pequeños cambios que fue necesario realizar, fue mucho mejor partir con una plantilla altamente funcional y sin errores de compilación.

11 Despliegue

11.1 Heroku

Se parte por instalar el CLI de Heroku, una vez instalado, se usa el comando *heroku login* en la terminal para iniciar sesión con la dirección de correo electrónico y la contraseña que usó al crear la cuenta de Heroku.

```
$ heroku login
heroku: Press any key to open up the browser to login or q to exit
  >  Warning: If browser does not open, visit
  >  https://cli-auth.heroku.com/auth/browser/***
heroku: Waiting for login...
Logging in... done
Logged in as me@example.com
```

Ilustración 80 Ejemplo de ingreso por terminal a través de Heroku CLI

Antes de poder implementar la aplicación en Heroku, se debe crear un repositorio Git para la aplicación y agregarle todo el código ejecutando estos comandos:

```
$ git init
$ git add .
$ git commit -m "first commit"
```

Ilustración 81 Comandos de inicio de repositorio git

Seguido se ejecuta la orden “push” hacia el servidor de Heroku.

```
$ git push heroku master
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Java app detected
remote: -----> Installing JDK 1.8... done
remote: -----> Executing Maven
...
remote: [INFO] BUILD SUCCESS
remote: -----
remote: [INFO] Total time: 15.129 s
remote: [INFO] Finished at: 2020-05-20T09:17:47Z
remote: [INFO] -----
```

Ilustración 82 Ejemplo de proceso push hacia el master en Heroku

Una vez terminado el proceso ya está lista la aplicación para ser accedida a través de un navegador.

11.2 PostgreSQL

Una de las ventajas que ofrece Heroku en su modalidad gratuita es la posibilidad de utilizar una base de datos PostgreSQL sin necesidad de realizar ningún gasto adicional. Heroku Postgres ofrece la base de datos de código abierto más avanzada del mundo como un servicio confiable, seguro y escalable que está optimizado para desarrolladores (Heroku, 2021).

Para configurar el acceso solo es necesario añadir la dependencia PostgreSQL en el archivo “pom.xml” y configurar el archivo “application.properties” con los datos de acceso.

```
<dependency>
  <groupId>org.postgresql</groupId>
  <artifactId>postgresql</artifactId>
</dependency>
```

Ilustración 83 Declaración de dependencia PostgreSQL en Maven

Si se usa Gradle la configuración es similar, del mismo modo si se usa otra base de datos SQL. Lo importante es tratar de crear consultas JPA o SQL nativas genéricas para permitirle a Spring poder ser compatible con cualquier tecnología de bases de datos SQL.

11.3 Despliegue alternativo

11.3.1 Despliegue a contenedores

Si está ejecutando la aplicación desde un contenedor, puede usar un jar ejecutable, pero a menudo también es una ventaja descomprimirlo y ejecutarlo de una manera diferente. Algunas implementaciones de PaaS (Platform-as-a-Service) también pueden optar por descomprimir archivos antes de que se ejecuten (Spring, 2020).

11.3.2 Kubernetes

Spring Boot detecta automáticamente los entornos de implementación de Kubernetes comprobando el entorno “*_SERVICE_HOST” y las “*_SERVICE_PORT” variables. Puede anular esta detección con la `spring.main.cloud-platform` propiedad de configuración (Spring, 2020).

11.3.3 Instalación de aplicaciones Spring Boot

Además de ejecutar aplicaciones Spring Boot mediante el uso `java -jar`, también es posible crear aplicaciones totalmente ejecutables para sistemas Unix. Un jar completamente ejecutable se puede ejecutar como cualquier otro binario ejecutable o se puede registrar con `init.d` o `systemd`. Esto ayuda al instalar y administrar aplicaciones Spring Boot en entornos de producción comunes (Spring, 2020).

11.3.4 Servicios de Microsoft Windows

Una aplicación Spring Boot se puede iniciar como un servicio de Windows usando `winsw` (Spring, 2020).

12 Implementación por PHP

Debido a restricciones económicas impuestas por la Universidad Surcolombiana no se permitió el despliegue de la aplicación en un servidor basado en Java, por lo que fue necesario convertir la vista del formato Thymeleaf a un formato HTML que fuese compatible con servicios de hospedaje compartido, para esta tarea se eligió utilizar PHP el cual es muy común en servicios de hospedaje de bajo costo.

12.2 Laravel

Para el diseño de la aplicación se siguió el patrón MVC (Modelo Vista Controlador) mediante el framework Laravel. Los modelos en Laravel se generaron utilizando la función “make model”, esta creó archivos de modelo para PHP en los cuales se insertaron los modelos hechos en Java que se modificaron ligeramente para encajar con el lenguaje PHP, luego mediante la función “migration” de Laravel fue posible generar las tablas necesarias en el motor de bases de datos MySQL.

Los controladores fueron creados mediante la función “make:controller”, de esta forma fue posible implementar rápidamente la lógica entre la base de datos y la vista.

Para las vistas se utilizó la función “make:view” la cual generar la vista en formato “nombre_modelo.blade.php” a partir de los modelos previamente establecidos.

12.3 Despliegue

El despliegue de la aplicación fue realizado mediante la generación de un proyecto Laravel a través de las opciones de creación en el software CPanel, acto seguido se actualizó el contenido del proyecto y se asignó la ruta de redirecciónamiento en el servidor.

Para la base de datos se utilizó MySQL y la estructura se conservó igual que en el proyecto basado en Spring Boot, a excepción de tablas nuevas requeridas por PHP para habilitar el sistema de ingreso de usuarios al dashboard.

12.4 Dashboard

Para el dashboard se convirtió el proyecto Spring Web en aplicación de escritorio y se redirigió la conexión de la base de datos hacia el almacenamiento compartido, lamentablemente la empresa prestadora de almacenamiento es muy restrictiva por lo cual no fue posible establecer un canal abierto para conectar la base de datos hacia aplicaciones externas; por este motivo fue necesario generar un nuevo dashboard para servicios CRUD basado en Laravel.

El nuevo dashboard conserva las mismas opciones CRUD que su contraparte en Java, con la excepción que las opciones para crear múltiples facultades fue eliminado ya que días antes de entregar el proyecto terminado se comunicó la separación de los programas de Ingeniería de Software y Tecnología en Desarrollo de Software en programas independientes, por lo que cada uno es responsable de su propia página web.

15.1 Portada principal

Para garantizar que no existiera problemas de tipo legal con respecto a los derechos de autor se creó una imagen de bienvenida utilizando el diseño del escudo de la Universidad Surcolombiana junto a una combinación de colores institucionales.

Aunque el logotipo oficial no tiene versión en dorado para su escudo, se eligió esta paleta de colores para hacer juego con los detalles que fueron resaltados con el color dorado número c7b363, el cual fue utilizado ampliamente en el diseño final de la plantilla del portal de Ingeniería de Software.



Ilustración 84 Imagen de bienvenida del portal de Ingeniería de Software

13 EDT

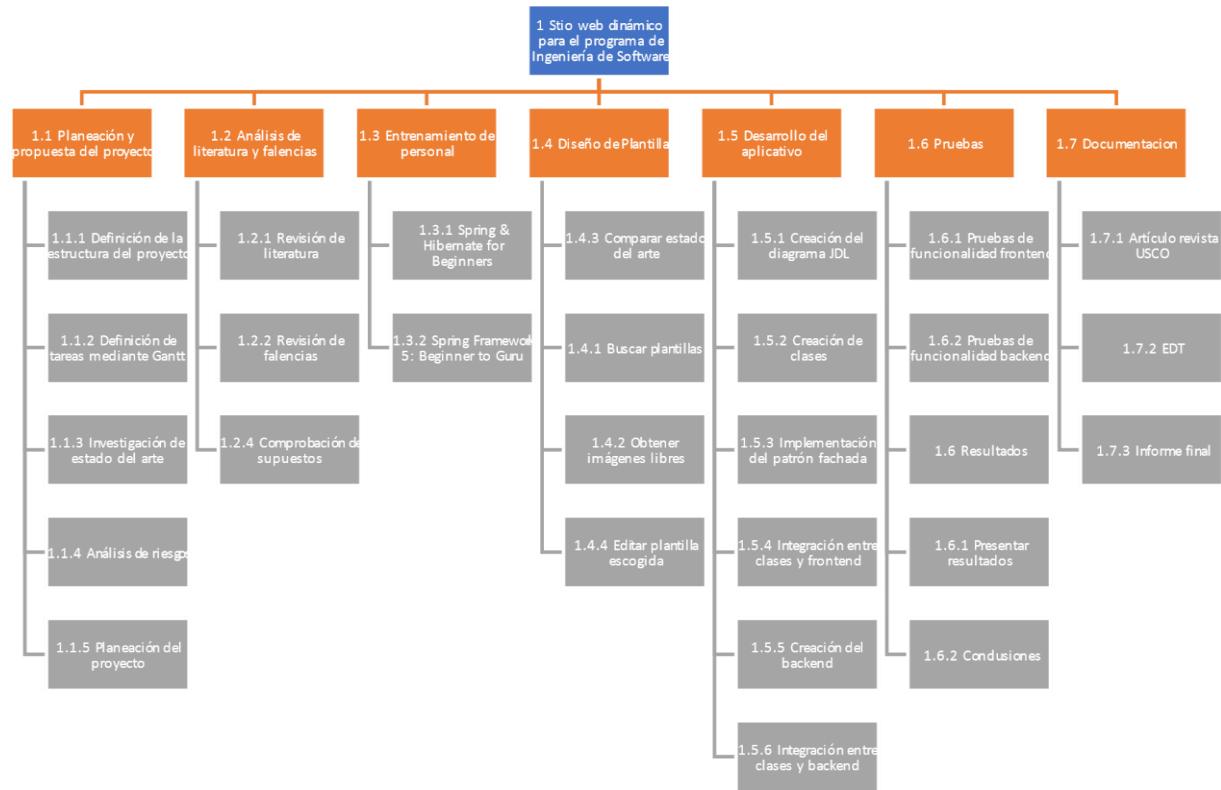


Ilustración 85 Estructura de desglose de trabajo

14 Cronograma

Tabla 11 Cronograma de actividades del proyecto

EDT	Nombre de tarea	Duración	Inicio	Final
1	Web dinámica para el programa de Ingeniería de Software	37 días	Mon 2/08/21	Tue 21/09/21
1.1	Planeación y propuesta del proyecto			
1.1.1	Definición de la estructura del proyecto	1 día	Mon 2/08/21	Mon 2/08/21
1.1.2	Definición de tareas mediante Gantt	1 día	Tue 3/08/21	Tue 3/08/21
1.1.3	Investigación de estado del arte	1 día	Wed 4/08/21	Wed 4/08/21
1.1.4	Ánalisis de riesgos	1 día	Thu 5/08/21	Thu 5/08/21
1.1.5	Planeación del proyecto	1 día	Fri 6/08/21	Fri 6/08/21
1.2	Análisis de literatura y falencias			
1.2.1	Revisión de literatura	1 día	Mon 9/08/21	Mon 9/08/21
1.2.2	Revisión de falencias	1 día	Tue 10/08/21	Tue 10/08/21
1.2.4	Comprobación de supuestos	1 día	Wed 11/08/21	Wed 11/08/21
1.3	Entrenamiento de personal			
1.3.1	Spring & Hibernate for Beginners	5 días	Mon 9/08/21	Fri 13/08/21
1.3.2	Spring Framework 5: Beginner to Guru	5 días	Mon 9/08/21	Fri 13/08/21
1.4	Diseño de Plantilla			
1.4.3	Comparar estado del arte	1 día	Mon 16/08/21	Mon 16/08/21
1.4.1	Buscar plantillas	1 día	Mon 16/08/21	Mon 16/08/21
1.4.2	Obtener imágenes libres	1 día	Mon 16/08/21	Mon 16/08/21
1.4.4	Editar plantilla escogida	3 días	Tue 17/08/21	Thu 19/08/21
1.5	Desarrollo del aplicativo			
1.5.1	Creación del diagrama JDL	2 días	Mon 23/08/21	Tue 24/08/21
1.5.2	Creación de clases	1 día	Tue 24/08/21	Tue 24/08/21
1.5.3	Implementación del patrón fachada	2 días	Tue 24/08/21	Wed 25/08/21
1.5.4	Integración entre clases y frontend	2 días	Wed 25/08/21	Thu 26/08/21
1.5.5	Creación del backend	2 días	Thu 26/08/21	Fri 27/08/21
1.5.6	Integración entre clases y backend	1 día	Fri 27/08/21	Fri 27/08/21
1.6	Pruebas			
1.6.1	Pruebas de funcionalidad frontend	3 días	Mon 30/08/21	Wed 1/09/21
1.6.2	Pruebas de funcionalidad backend	2 días	Thu 2/09/21	Fri 3/09/21
1.6	Resultados			
1.6.1	Presentar resultados	3 días	Sat 4/09/21	Tue 7/09/21
1.6.2	Conclusiones	2 días	Tue 7/09/21	Wed 8/09/21
1.7	Documentación			

1.7.1	Artículo revista USCO	1 día	Tue 21/09/21	Tue 21/09/21
1.7.2	EDT	1 día	Tue 21/09/21	Tue 21/09/21
1.7.3	Informe final	6 días	Tue 14/09/21	Tue 21/09/21

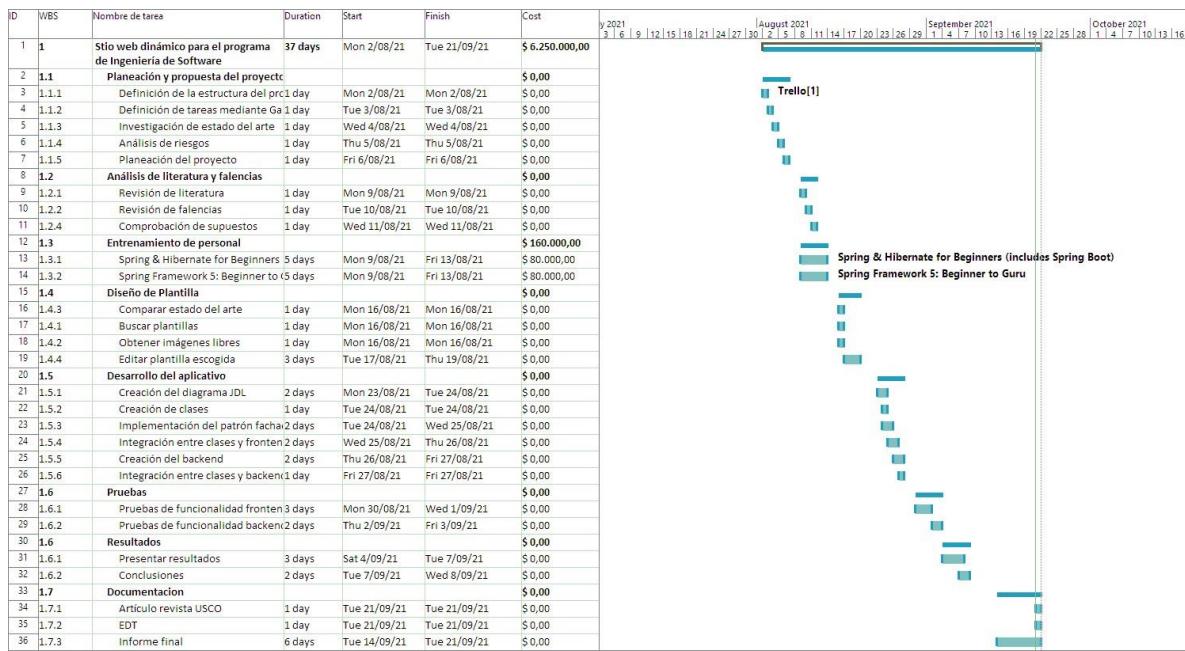


Ilustración 86 Diagrama de Gantt del proyecto

15 Presupuesto

El valor del sueldo por hora se calcula a partir de la información suministrada por el Gobierno Nacional para un empleador (MinTrabajo, 2021). A partir de esa información se dividió el total del sueldo entre 30 días para encontrar el valor por día.

Tabla 12 Cálculo del costo total de un empleado en Colombia, con un salario de 3M de pesos en 2021

SALARIO	3,000,000
Transporte	0
PRESTACIONES SOCIALES	
Cesantías	250000
Intereses sobre cesantías	30000
Primas	250000
Vacaciones	125000
APORTES A LA SEGURIDAD SOCIAL	
Pensiones (AFP)	360000
Salud (EPS)	255000
Riesgos Laborales (ARL)	15660
PARAFISCALES	
Caja de compensación familiar	120000
ICBF	90000
SENA	60000
TOTAL	4,150,660

Tabla 13 Listado de recursos utilizados y sus costos por periodo

Recurso	Tipo	Grupo	Tasa estándar
Ingeniero 1	Trabajo	Personal	\$ 17.294/hora
Spring & Hibernate for	Trabajo	Capacitación	\$ 0/hora
Spring FrameTrabajo 5: Beginner to Guru	Trabajo	Capacitación	\$ 0/hora
Energía	Trabajo	Servicios	\$ 130.000/mes
Agua	Trabajo	Servicios	\$ 40.000/mes
Gas	Trabajo	Servicios	\$ 30.000/mes
Internet	Trabajo	Servicios	\$ 100.000/mes
Arriendo	Trabajo	Planta física	\$ 800.000/mes
Mesa	Material	Planta física	\$ 150.000
Silla de malla	Material	Planta física	\$ 500.000
Ventilador	Material	Planta física	\$ 130.000
Elementos de limpieza	Material	Planta física	\$ 70.000
Ryzen 7 PC	Material	Hardware	\$ 3.000.000
Honor 5x	Material	Hardware	\$ 350.000
UPS 600WA	Material	Hardware	\$ 110.000
AWS ml.m4.4xlarge	Trabajo	Servicio Cloud	\$ 3.673/hora
Krita	Material	Software	\$ 0
Trello	Material	Software	\$ 0

Tabla 14 Distribución de costos por tarea

1	Sitio web dinámico para el programa de Ingeniería de Software	37 días	\$ 11.774.903
1.1	Planeación y propuesta del proyecto		\$ 138.352
1.1.1	Definición de la estructura del proyecto	1 día	\$ 0
1.1.2	Definición de tareas mediante Gantt	1 día	\$ 0
1.1.3	Investigación de estado del arte	1 día	\$ 0
1.1.4	Ánalisis de riesgos	1 día	\$ 0
1.1.5	Planeación del proyecto	1 día	\$ 0
1.2	Análisis de literatura y falencias		\$ 415.056
1.2.1	Revisión de literatura	1 día	\$ 0
1.2.2	Revisión de falencias	1 día	\$ 0
1.2.4	Comprobación de supuestos	1 día	\$ 0
1.3	Entrenamiento de personal		\$ 1.820.224
1.3.1	Spring & Hibernate for Beginners	5 días	\$ 80.000
1.3.2	Spring Framework 5: Beginner to Guru	5 días	\$ 80.000
1.4	Diseño de Plantilla		\$ 1.106.816
1.4.3	Comparar estado del arte	1 día	\$ 0
1.4.1	Buscar plantillas	1 día	\$ 0
1.4.2	Obtener imágenes libres	1 día	\$ 0
1.4.4	Editar plantilla escogida	3 días	\$ 0
1.5	Desarrollo del aplicativo		\$ 1.085.991
1.5.1	Creación del diagrama JDL	2 días	\$ 0
1.5.2	Creación de clases	1 día	\$ 0
1.5.3	Implementación del patrón fachada	2 días	\$ 0
1.5.4	Integración entre clases y frontend	2 días	\$ 117.527
1.5.5	Creación del backend	2 días	\$ 0
1.5.6	Integración entre clases y backend	1 día	\$ 0
1.6	Pruebas		\$ 0
1.6.1	Pruebas de funcionalidad frontend	3 días	\$ 0
1.6.2	Pruebas de funcionalidad backend	2 días	\$ 0
1.6	Resultados		\$ 138.352
1.6.1	Presentar resultados	3 días	\$ 0
1.6.2	Conclusiones	2 días	\$ 0
1.7	Documentación		\$ 830.112
1.7.1	Artículo para la revista USCO	1 día	\$ 0
1.7.2	EDT	1 día	\$ 0
1.7.3	Informe final	6 días	\$ 0

El costo final del proyecto en su etapa de desarrollo para Spring fue de 11.774.903 pesos. Obviamente este es un precio exagerado y casi irreal si se considera la relativa simplicidad de los requisitos, pero es importante recordar que la lista de elementos que eleva tanto la cuenta se expone para dar a conocer todo lo que se necesita en el caso de iniciar un proyecto desde cero.

15.1 Presupuesto para conversión hacia PHP

Para la modificación del proyecto al lenguaje PHP se calcula por días de trabajo a 30.000 pesos hora.

Tabla 15 Costo de trabajo por días para conversión a PHP

DATOS LIQUIDACIÓN	
Periodo (DD-MM-AAAA)	01/02/2022 al 15/02/2022
Días Laborados (mensualizado)	21,65
Salario Diario	30.000
Salario Mensualizado	649.500
Transporte	0
Prima primer semestre*	27.063
Prima segundo semestre	0
Cesantías	27.063
Intereses sobre cesantías	135
Vacaciones	13.531
TOTAL	67.792
APORTES A LA SEGURIDAD SOCIAL (Valor mensual)	
Pensiones (AFP)	0

Tabla 16 Prestaciones sociales por parte del empleador

PRESTACIONES SOCIALES	
Cesantías	54.125
Intereses sobre cesantías	6.495
Primas	54.125
Vacaciones	27.063
APORTES A LA SEGURIDAD SOCIAL EMPLEADOR	
Pensiones (AFP)	0
Riesgos Laborales (ARL)	0
PARAFISCALES	
Caja de compensación	0
TOTAL	94.854

15.2 Presupuesto de hosting

Comparativas de servicios de hosting compatibles con Spring y PHP. Algunos precios son relativamente bajos a la hora de elegir un servidor con el soporte necesario, el problema radica en que para obtener el mejor descuento es necesario adquirir planes de permanencia dos a tres años. Los precios de oferta mensual luego son incrementados al final del contrato.

Es una práctica común que se incluya el nombre de dominio en el primer año, pero hay que tener en cuenta este valor en años posteriores. Los precios de dominios varían mucho dependiendo del tema, los ".com" se consiguen entre 7 a 15 dólares americanos.

15.2.1 Spring

Tabla 17 Comparativa de precios para hosting con soporte para Tomcat

Nombre	Dailyrazor	a2hosting	Hostinger
Dominios	1	1	1
Espacio (gb)	2	50	20
Transferencia (gb)	50	2000	1000
Bases de datos	5	No informa	No informa
Mes (usd)	3.55	6.59	4.95
Anual (usd)	42.60	79.08	59.4
Mes al siguiente año	8.95	8.99	9.05

15.2.2 PHP

Tabla 18 Comparativa de precios para hosting con soporte para Tomcat

Nombre	Hostinger	namecheap	bluehost
Dominios	1	3	1
Espacio (gb)	30	20	50
Transferencia (gb)	100	Ilimitado	Ilimitado
Bases de datos	2	50	20
Mes (usd)	3.49	1.88	2.95
Anual (usd)	41.88	31.04	47.28
Mes al siguiente año	9.99	3,88	9.99

16 Conclusiones

Se puede afirmar que el objetivo principal se ha conseguido con éxito ya que la aplicación se ha desplegado de manera satisfactoria y cumple con los requerimientos establecidos para ser utilizada de forma dinámica por uno o múltiples programas académicos de la Universidad Surcolombiana.

Construir una solución personalizada no es una tarea sencilla ya que requiere conocer las variables que maneja cada programa que use dicho diseño, por lo que tratar de satisfacer a todos los interesados puede no ser una tarea fácil y en el caso de querer expandir el uso del aplicativo a otros programas se requeriría la estandarización de los contenidos o la adición de nuevas características.

Afortunadamente la arquitectura en la que fue desarrollada la aplicación permite añadir rápidamente nuevas funcionalidades mediante el patrón fachada, y la creación de nuevas vistas no es una tarea complicada ya que la interacción es directa entre vista y servidor de aplicaciones, por lo que el nivel de lógica del lado de la vista es casi inexistente y solo se limita a representar la información entregada por los controladores.

El cambio en los requisitos de un proyecto puede obligar a los desarrolladores a realizar modificaciones demasiado grandes que obligan a reconsiderar la viabilidad del proyecto, en este caso la transformación del software desde el lenguaje Java hacia PHP se realizó relativamente sin problemas debido a que desde un principio se realizó el software siguiendo normas de ingeniería entre las cuales se encuentran patrones de diseño los cuales permitieron transformar la arquitectura Java a su equivalente en PHP.

17 Bibliografía

"maleadora", N. (2019). *Qué es Frontend y Backend: diferencias y características - Platzi*. Obtenido de Plazi: <https://platzi.com/blog/que-es-frontend-y-backend/>

Amazon. (19 de 9 de 2021). *What is PostgreSQL?* Obtenido de Amazon: <https://aws.amazon.com/rds/postgresql/what-is-postgresql/>

Build With. (16 de 9 de 2021). *Detailed Technology Profile*. Obtenido de BuildWith.com: <https://builtwith.com/detailed/usco.edu.co>

Built With. (15 de 9 de 2021). *usco.edu.co Detailed Technology Profile*. Obtenido de Built With: <https://builtwith.com/detailed/usco.edu.co>

Commentator, E. (3 de 1 de 2020). *What is the importance of web design for your audience?* Obtenido de Smart Insights: <https://www.smartsights.com/user-experience/websitedesign/what-is-the-importance-of-web-design-for-your-audience/>

Content, R. R. (5 de 6 de 2019). *¿Qué es Java? Conoce las particularidades de este lenguaje de programación.* Obtenido de RockContent: <https://rockcontent.com/es/blog/que-es-java/>

Erinç, Y. (15 de 4 de 2021). *Spring Security Overview*. Obtenido de Auth0.com: <https://auth0.com/blog/spring-security-overview/>

EZ Marketing. (21 de 8 de 2018). *Why good web design is important, and why you need it.* Obtenido de EZ Marketing: <https://blog.ezmarketing.com/why-good-web-design-isimportant>

Google. (2012). *The role of visual complexity and prototypicality regarding first impression of websites: Working towards understanding aesthetic judgments.* Obtenido de Google Research: <https://research.google/pubs/pub38315/>

Granja, J. (20 de 9 de 2021). *Hello Spring Security with Boot.* Obtenido de Spring.io: <https://docs.spring.io/spring-security/site/docs/current/guides/helloworld-boot.html>

Heroku. (16 de 9 de 2021). *Managed PostgreSQL from Heroku.* Obtenido de Heroku: <https://www.heroku.com/postgres>

IBM. (20 de 1 de 2021). *JPA (Java Persistence API).* Obtenido de IBM.com: <https://www.ibm.com/docs/es/was-liberty/nd?topic=overview-java-persistence-api-jpa>

Inovatik. (16 de 9 de 2021). *License Details.* Obtenido de Inovatik: <https://inovatik.com/license.html>

javaTpoint. (15 de 9 de 2021). *Spring Boot Thymeleaf.* Obtenido de javaTpoint: <https://www.javatpoint.com/spring-boot-thymeleaf-view>

javaTpoint. (15 de 9 de 2021). *Spring Boot Tutorial.* Obtenido de javaTpoint: <https://www.javatpoint.com/spring-boot-tutorial>

JHipster. (16 de 9 de 2021). *JHipster Domain Language (JDL)*. Obtenido de Jhipster.tech: <https://www.jhipster.tech/jdl/intro>

Megler, V. (2019). En *Managing Machine Learning*.

Microsoft. (2020). *Access SQL*. Obtenido de Microsoft.com: <https://support.microsoft.com/eses/office/access-sql-conceptos-b%C3%A1sicos-vocabulario-y-sintaxis-444d0303-cde1424e-9a74-e8dc3e460671>

MinTrabajo. (21 de 9 de 2021). *Mi Calculadora*. Obtenido de MinTrabajo.gov.co: Cálculo del costo total de un empleado en Colombia, con un salario de 3M de pesos en 2021

Mozilla. (18 de 9 de 2021). *CRUD*. Obtenido de MDN Web Docs Mozilla: <https://developer.mozilla.org/es/docs/Glossary/CRUD>

Mozilla. (20 de 9 de 2021). *MVC*. Obtenido de MDN Web Docs Mozilla: <https://developer.mozilla.org/es/docs/Glossary/MVC>

Nikolaieva, A. (20 de 9 de 2021). *8 Best Software Development Methodologies* . Obtenido de uptech.team: <https://www.uptech.team/blog/software-development-methodologies>

Oracle. (1 de 1 de 2014). *Using Bean Validation Constraints*. Obtenido de Oracle: <https://docs.oracle.com/javaee/7/tutorial/bean-validation001.htm>

Rae.es. (20 de 9 de 2021). *Pésum*. Obtenido de RAE: <https://dle.rae.es/p%C3%A9sum>

Refactoring.Guru. (29 de 9 de 2021). *¿Qué es un patrón de diseño?* Obtenido de Refactoring.Guru:

<https://refactoring.guru/es/design-patterns/what-is-pattern>

Refactoring.Guru. (20 de 9 de 2021). *Facade*. Obtenido de refactoring.guru: <https://refactoring.guru/es/design-patterns/facade>

Sharma, M. (15 de 12 de 2020). *Spring Security with Thymeleaf*. Obtenido de JavaDevJournal: <https://www.javadevjournal.com/spring-security/spring-security-with-thymeleaf/>

Sherwin, J., & Calleja , T. (1 de 1 de 2016). *Spring MVC Blueprints*. Obtenido de O'Reilly: <https://www.oreilly.com/library/view/spring-mvcblueprints/9781785888274/ch07s05.html>

Skelia. (23 de 4 de 2017). *Scrum vs Kanban: The Good, The Bad and The Ugly*. Obtenido de Skelia:

<https://skelia.com/articles/scrum-vs-kanban-the-good-the-bad-and-the-ugly/>

Spring. (22 de 9 de 2020). *Deploying Spring Boot Applications*. Obtenido de Spring.io: <https://docs.spring.io/spring-boot/docs/current/reference/html/deployment.html>

Spring.io. (20 de 9 de 2021). *Spring Security*. Obtenido de Spring.io:
<https://spring.io/projects/spring-security>

Techopedia. (18 de 7 de 2011). *PostgreSQL*. Obtenido de Techopedia:
<https://www.techopedia.com/definition/3499/postgresql>

Themewagon. (16 de 9 de 2021). *Mirko - Free Bootstrap 5 HTML5 Business Website Template*. Obtenido de Themewagon: <https://themewagon.com/themes/free-bootstrap-5-html-5business-website-template-mirko/>

Thymeleaf. (21 de 12 de 2020). *Thymeleaf*. Obtenido de Thymeleaf.org:
<https://www.thymeleaf.org/>

Thymeleaf.org. (20 de 9 de 2021). *Thymeleaf + Spring Security integration basics*. Obtenido de Thymeleaf.org: <https://www.thymeleaf.org/doc/articles/springsecurity.html>

UC Merced. (22 de 9 de 2021). *What Is a Data Dictionary?* Obtenido de UC Merced Library: <http://library.ucmerced.edu/data-dictionaries>

Universidad de Murcia. (15 de 9 de 2021). *Páginas web dinámicas*. Obtenido de Universidad de Murcia: <https://www.um.es/docencia/barzana/DAWEB/2017-18/daweb-tema-13-paginasweb-dinamicas.html>

Universidad Surcolombiana. (20 de 9 de 2021). *Misión y Visión*. Obtenido de Universidad Surcolombiana: <https://www.usco.edu.co/movil/es/la-universidad/mision-y-vision/>

USCO. (16 de 9 de 2021). *Imagen Institucional*. Obtenido de Universidad Surcolombiana: <https://www.usco.edu.co/imagen-institucional/>

WebFX. (17 de 9 de 2021). *Why Is Web Design Important?* Obtenido de webfx.com: <https://www.webfx.com/web-design/why-is-web-design-important.html>

Yadanar, T. (19 de 7 de 2020). *What is JPA?* Obtenido de Dev.to: <https://dev.to/reytechlesson/what-is-jpa-589f>