

Prototipo de entorno virtual para valorar el impacto de filósofos colombianos en diferentes contextos de la sociedad.

Daniel Alfonso Vega Camelo

Trabajo de Grado para Optar al título de Ingeniero de Sistemas

Directora

Sonia Cristina Gamboa Sarmiento

Ph. D en Educación, Filosofía y enseñanza de la Filosofía

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería de Sistemas e Informática

Bucaramanga

2019

A mi familia.

Daniel.

Agradecimientos

A mi hermana, mis padres y amigos que me han dado su apoyo incondicional durante este largo viaje y que han aportado a mi desarrollo como profesional y como persona.

A la profesora Sonia Cristina Gamboa Sarmiento, por confiar en mí para el desarrollo de este proyecto, por su apoyo, asesoría y guía que hicieron posible la culminación del mismo.

Muchas gracias.

Tabla de Contenido

Introducción	14
1. Planteamiento del problema	15
2. Justificación	17
3. Objetivos.....	19
3.1 Objetivo general.....	19
3.2 Objetivos específicos	19
4. Marco de antecedentes.....	20
5. Marco conceptual.....	26
5.1. Fundamentos teóricos	26
5.1.1. Ingeniería Ontológica.....	26
5.1.2. Entornos virtuales.	27
5.1.3. Framework web	29
5.1.4. MVVC.....	30
5.1.5. MVC	31
5.1.6. REST.....	33
5.1.7. SPA (Single Page Application).....	35
5.2. Tecnologías involucradas en el desarrollo del proyecto	36
5.2.1. HTML5	36
5.2.2. CSS3	36
5.2.3. Vue.js	37
5.2.4. Laravel	37
5.2.5. INSOMNIA.....	38
5.2.6. MySQL	39
6. Metodología.....	40
7. Resultados.....	44
7.1. Especificación de requisitos Software	44

7.1.1. Propósito.	45
7.1.2. Ámbito del sistema.	45
7.1.3. Perspectiva del producto.	45
7.1.4. Funciones del producto.	46
7.1.4.1. Gestión de los usuarios.	46
7.1.4.2. Gestión de los filósofos.....	46
7.1.4.3. Gestión de los aportes de los filósofos.....	46
7.1.4.4. Perfiles de filósofos y de aportes.	46
7.1.5. Características de los usuarios.	46
7.1.6. Restricciones.	46
7.1.7. Requerimientos funcionales.....	47
7.1.8. Requerimientos no funcionales.....	51
7.1.9. Actores del sistema.	52
7.1.10. Descripción de actores.	52
7.1.11. Diagrama de casos de uso.	54
7.1.12. Descripción de casos de uso.	56
7.1.12.1. Autenticar usuario.....	56
7.1.12.2. Enviar solicitud de registro.....	57
7.1.12.3. Recuperar contraseña.	58
7.1.12.4. Contactar administrador general.	59
7.1.12.5. Gestionar mensajes de contacto a administrador / curador / auxiliar.....	59
7.1.12.6. Contactar curador.....	60
7.1.12.7. Buscar información.....	61
7.1.12.8. Gestionar información de usuario.....	62
7.1.12.9. Gestionar filósofo.....	63
7.1.12.10. Gestionar periodo.....	64
7.1.12.11. Gestionar aporte.....	65
7.1.12.12. Gestionar evidencia.....	66
7.1.12.13. Gestionar acceso a auxiliares/curadores.....	67
7.1.12.14. Gestionar asignaciones/permisos de filósofos a auxiliares/curadores.....	68
7.1.13. Diagramas de flujo.....	68

7.2. Diseño de la base de datos	69
7.2.1. Diccionario de datos	70
7.3. Arquitectura Cliente-Servidor de tres capas	70
7.3.1. Frontend	73
7.3.2. Backend.....	77
7.4. Vistas de usuario	82
7.5. Pruebas.....	85
7.5.1. Prueba de contenido.....	86
7.5.2. Pruebas de compatibilidad.....	87
7.5.3. Prueba de navegación.....	87
7.5.4. Pruebas de función.....	89
7.5.5. Pruebas de seguridad.....	90
8. Conclusiones.....	93
9. Recomendaciones	94
Referencias bibliográficas	95
Apéndices	98

Lista de tablas

Tabla 1. Filósofos colombianos y sus correspondientes filósofos curadores.	45
Tabla 2. Requerimiento Funcional No. 01 – Registro de usuarios	47
Tabla 3. Requerimiento Funcional No. 02 – Autenticación de usuarios	47
Tabla 4. Requerimiento Funcional No. 03 – Gestión de permisos a usuarios	48
Tabla 5. Requerimiento Funcional No. 04 – Gestión de filósofos.....	48
Tabla 6. Requerimiento Funcional No. 05 – Perfiles de filósofos.....	49
Tabla 7. Requerimiento Funcional No. 06 – Búsqueda de filósofos y aportes.....	49
Tabla 8. Requerimiento Funcional No. 07 – Mensajes de contacto	50
Tabla 9. Requerimiento Funcional No. 08 – Editar información de usuario	50
Tabla 10. Requerimiento No Funcional No. 01 – Validación de datos	51
Tabla 11. Requerimiento No Funcional No. 02 – Interfaz del sistema.....	51
Tabla 12. Requerimiento No Funcional No. 03 – Accesibilidad del sistema.....	51
Tabla 13. Descripción de Actores – Visitante.	52
Tabla 14. Descripción de Actores – Administrador.....	52
Tabla 15. Descripción de Actores – Curador.....	53
Tabla 16. Descripción de Actores – Auxiliar.....	53
Tabla 17. Descripción de Casos de Uso – Autenticar usuario.....	56
Tabla 18. Descripción de Casos de Uso – Enviar solicitud de registro	57
Tabla 19. Descripción de Casos de Uso – Recuperar contraseña.	58
Tabla 20. Descripción de Casos de Uso – Contactar administrador general	59
Tabla 21. Descripción de Casos de Uso – Gestionar mensajes de contacto a administrador/curador/auxiliar.....	59
Tabla 22. Descripción de Casos de Uso – Contactar curador.....	60
Tabla 23. Descripción de Casos de Uso – Buscar información.....	61
Tabla 24. Descripción de Casos de Uso – Gestionar información de usuario.....	62
Tabla 25. Descripción de Casos de Uso – Gestionar filósofo.....	63
Tabla 26. Descripción de Casos de Uso – Gestionar periodo.....	64
Tabla 27. Descripción de Casos de Uso – Gestionar aporte	65
Tabla 28. Descripción de Casos de Uso – Gestionar evidencia.....	66
Tabla 29. Descripción de Casos de Uso – Gestionar acceso a auxiliares/curadores	67
Tabla 30. Descripción de Casos de Uso – Gestionar asignaciones de filósofos a auxiliares/curadores	68
Tabla 31. Descripción de los atributos de filósofo.	70
Tabla 32. Endpoints de la API del Backend	80
Tabla 33. Resultados de la prueba de contenido.....	86
Tabla 34. Resultados de la prueba de compatibilidad.....	87
Tabla 35. Resultados de la prueba de navegación	88
Tabla 36. Resultados prueba de seguridad.....	90

Lista de figuras

Figura 1. Sitio web dedicado a las obras de Estanislao Zuleta. (Facultad de humanidades Universidad del Valle, 2006).	21
Figura 2. Página principal del portal Dialnet. (Universidad de la Rioja, 2001)	22
Figura 3. Página principal de la biblioteca Scielo. (Fundacion para el apoyo a la investigacion del estado de Sao paulo [FAPESP], 1998).	22
Figura 4. Biblioteca virtual del banco de la república de Colombia. (Banco de la República de Colombia, 1997)	23
Figura 5. Página principal del Sistema de Información Científica Redalyc. (Universidad Autónoma del Estado de México, 2003).....	24
Figura 6. Sección de filósofos de la página Webdianoia. (Fouce, 2001).....	24
Figura 7. Patrón MVVM implementado en Vue.js. Adaptado de (You, 2014).....	30
Figura 8. Capas del patrón MVVM. Adaptado de (Microsoft Corporation, 2014)	31
Figura 9. Capas del patrón MVC.	32
Figura 10. Disciplinas y fases fundamentales en el ciclo de vida del desarrollo de software. Adaptado de (Tabares, Barrera, Arroyave, & Pineda, 2007).....	44
Figura 11. Actores del sistema.....	52
Figura 12. Diagrama general de casos de uso y actores del sistema.....	55
Figura 13. Detalle de los casos de uso Contactar administrador y Enviar solicitud de registro ...	56
Figura 14. Diagrama entidad-relación.	70
Figura 15. Capas de la arquitectura Cliente-Servidor. Adaptado de (Marini, 2012).....	71
Figura 16. Implementación de la arquitectura.	73
Figura 17. Estructura del código fuente del Frontend.....	74
Figura 18. Estructura del directorio /components.....	75
Figura 19. Componentes /AdminComponents.....	76
Figura 20. Componentes /GeneralComponents y /PerfilComponents.....	76
Figura 21. Componentes /HomeComponents.	76
Figura 22. Estructura básica de un componente.	77
Figura 23. Estructura del Backend de la plataforma.	78
Figura 24. Estructura del directorio /app.	78
Figura 25. Frontend y API mediante el patrón MVC.	80
Figura 26. Prototipo 1, sección de campos de acción.	83
Figura 27. Prototipo 1, sección de evidencias.....	83
Figura 28. Prototipo 2, sección línea de tiempo.....	84
Figura 29. Prototipo 3, sección línea de tiempo.....	84
Figura 30. Prototipo final, sección línea de tiempo.	85
Figura 31. Autenticación ssh con clave privada y pública.....	92

Lista de apéndices

Apéndices	98
Apéndice A. Diagramas de flujo.	98
Apéndice B. Diccionario de datos.	106
Apéndice C. Vistas de usuario.	112
Apéndice D. Casos de prueba.	127

RESUMEN

TÍTULO: PROTOTIPO DE ENTORNO VIRTUAL PARA VALORAR EL IMPACTO DE FILÓSOFOS COLOMBIANOS EN DIFERENTES CONTEXTOS DE LA SOCIEDAD*

AUTOR: VEGA CAMELO, Daniel Alfonso**

PALABRAS CLAVE: FILOSOFÍA COLOMBIANA, TIC, TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN, ENTORNO VIRTUAL, DESARROLLO WEB.

DESCRIPCIÓN:

La filosofía, entendida como una disciplina que usa un conjunto de razonamientos lógicos y metódicos, está presente en muchos contextos de la sociedad y la vida cotidiana, en los fundamentos éticos, filosóficos y racionales con los que las personas actúan y toman decisiones, sin embargo, no es vista necesariamente por la sociedad como algo relevante, se tiene la idea de que la filosofía es anticuada, no tiene usos prácticos y solo es ejercida por un conjunto reducido de personas. En Colombia, la filosofía goza de poco reconocimiento como disciplina básica necesaria en la formación intelectual y cultural de la ciudadanía, en sus aportes teóricos y prácticos a la resolución de problemas sociales, económicos y políticos. Contrario a lo que se cree, los hitos políticos, sociales, económicos, educativos más significativos han contado con la presencia y el soporte de filósofos, más reconocidos por sus publicaciones que por su participación anónima en tales eventos.

En este proyecto se plantea el desarrollo de un entorno virtual para visibilizar la obra no académica de los filósofos colombianos más representativos en este ámbito (no académico). Se propone como solución a la problemática planteada, mostrar de forma didáctica e interactiva la trayectoria de los filósofos colombianos y los aportes que han realizado a la sociedad en diferentes contextos sociales como la educación, la cultura y la política, entre otros, de manera que los visitantes del portal pierdan ciertos prejuicios que existen con relación a los filósofos y la filosofía. También servirá como una herramienta para evaluar el desempeño de los filósofos colombianos en cada uno de los contextos en los que participó y como un recurso informativo para ciudadanos interesados en esta disciplina en el ámbito colombiano.

* Proyecto de Grado

** Universidad Industrial de Santander, Facultad de Ingenierías Fisicomecánicas, Escuela de Ingeniería de Sistemas e Informática, Ingeniería de Sistemas. Director: Sonia Cristina Gamboa Sarmiento.

ABSTRACT

TITLE: PROTOTYPE OF VIRTUAL ENVIRONMENT TO VALUE THE IMPACT OF COLOMBIAN PHILOSOPHERS IN DIFFERENT CONTEXTS OF THE SOCIETY*

AUTHOR: VEGA CAMELO, Daniel Alfonso**

KEYWORDS: COLOMBIAN PHILOSOPHY, ICT, INFORMATION AND COMMUNICATIONS TECHNOLOGY, VIRTUAL ENVIRONMENT, WEB DEVELOPMENT.

DESCRIPTION:

Philosophy, understood as a discipline that uses a set of logical and methodical reasoning, is present in many contexts of society and everyday life, in the ethical, philosophical and rational foundations with which people act and make decisions, however, it is not necessarily seen by society as something relevant, one has the idea that philosophy is outdated, has no practical uses and is only exercised by a small group of people. In Colombia, philosophy enjoys little recognition as a necessary basic discipline in the intellectual and cultural formation of citizens, in its theoretical and practical contributions to the resolution of social, economic and political problems. Contrary to popular belief, the most significant political, social, economic, educational milestones have had the presence and support of philosophers, more recognized for their publications than for their anonymous participation in such events.

This project proposes the development of a virtual environment to make visible the non-academic work of the most representative Colombian philosophers in this field (non-academic). It is proposed as a solution to the problem raised, to show in a didactic and interactive way the trajectory of Colombian philosophers and the contributions they have made to society in different social contexts such as education, culture and politics, among others, so that Portal visitors lose certain prejudices that exist in relation to philosophers and philosophy. It will also serve as a tool to evaluate the performance of Colombian philosophers in each of the contexts in which they participated and as an information resource for citizens interested in this discipline in the Colombian sphere.

* Bachelor Thesis.

** Industrial University of Santander, Physical and Mechanical Engineering Faculty, Department of Systems Engineering and Informatics, Systems Engineering. Direct by Sonia Cristina Gamboa Sarmiento.

Introducción

La situación actual de posconflicto en Colombia –después de la firma de los acuerdos de paz con el grupo guerrillero de las FARC– requiere que los miembros de la sociedad se preparen en disciplinas relacionadas con la cultura humana, en especial sobre filosofía, para que puedan proveer un ambiente en el cual sea posible la reintegración de los desmovilizados y la reconciliación de los involucrados en la guerra.

Se hace énfasis en la educación sobre filosofía ya que esta disciplina influye en la formulación, ejecución y seguimiento de políticas de estado en las que el sentido ético de lo público sea lo primordial, la integración del sector privado en proyectos humanistas, y el fomento de los principios democráticos en el ejercicio de la ciudadanía. Esta influencia no es visible debido a que las personas no tienen en cuenta los fundamentos filosóficos, éticos, racionales, morales y humanistas con los cuales actúan y toman decisiones, por lo tanto, varios sectores de la sociedad se han formulado la pregunta “¿Dónde están los filósofos?”.

Para resolver esta situación donde no se conoce la necesidad y la importancia de la filosofía, se diseñó un entorno virtual que muestra la trayectoria y el impacto que han tenido los filósofos en distintos contextos de la sociedad colombiana, mostrando evidencia de las investigaciones y de acciones concretas en diferentes contextos sociales en Colombia. El entorno virtual también sirve como una herramienta para evaluar el desempeño de los filósofos en cada uno de los contextos en los que participó y como un recurso informativo para los ciudadanos interesados en la disciplina de la filosofía.

El desarrollo del proyecto inicia con la definición de la problemática, la justificación del proyecto y los objetivos, los cuales se encuentran en los primeros tres apartados. A continuación,

se hace una revisión de antecedentes sobre herramientas tecnológicas y proyectos universitarios que han abordado el tema de la filosofía en Colombia y en Latinoamérica. Posteriormente, en los apartados 5 y 6, se definen los conceptos, metodología y tecnologías involucradas en el desarrollo del proyecto. En el siguiente capítulo se describirá el proceso de desarrollo del entorno virtual propuesto y finalmente se presentan las conclusiones y recomendaciones de este proyecto.

1. Planteamiento del problema

La filosofía realmente es de gran importancia en todos los contextos importantes de la sociedad como la educación, la cultura y la política, también está presente en el diario vivir de todas las personas en los fundamentos éticos, filosóficos, racionales, morales y humanistas con los que actúan y toman decisiones. Aun así, en la sociedad actual existen muchos prejuicios sobre esta área del conocimiento lo que ha resultado en críticas en medios de comunicación a las personas que deciden estudiar y ejercer la filosofía, un ejemplo es el artículo de la revista Arcadia titulado “¿Dónde están los filósofos?” (Restrepo, 2011) donde se dice que la filosofía es un área del conocimiento que solo es estudiada por un grupo muy reducido de la sociedad, los cuales según Restrepo “Parecen encontrarse en la torre de marfil de la academia, distanciados de una realidad compleja y fecunda para el pensamiento”, también afirma que los filósofos están fuera de la discusión pública de las problemáticas que enfrenta el país porque parece que les da miedo aparecer en los medios de comunicación, acto que los haría parecer livianos y superficiales en el filosofar de las cosas.

Otro prejuicio muy conocido es que la comunidad filosófica es machista y esto se ve reflejado en la participación de las mujeres en esta área del conocimiento, a lo largo de la historia se les ha omitido o no se les ha dado reconocimiento que se merecen, principalmente porque durante mucho tiempo se creía que las mujeres no estaban al mismo nivel intelectual de los hombres, afortunadamente durante los últimos siglos esto ha ido desapareciendo, y se cuenta con el reconocimiento de grandes filósofas como Simone de Beauvoir, Hannah Arendt, Judith Blutter, Martha Nussbaum, y, en el ámbito colombiano, con Beatriz Restrepo, fallecida en días pasados.

En la actualidad los pocos estudiantes que se atreven a estudiar filosofía descubren que el aprendizaje requiere de una gran cantidad de recursos bibliográficos que son de difícil acceso debido al poco número de ejemplares físicos en las bibliotecas e instituciones educativas y las versiones digitales muchas veces no se encuentran o tienen precios que los estudiantes no pueden costear, por lo cual generalmente se recurren a copias digitales que infringen los derechos de autor aun cuando en el Art. 2 de la ley 1030 del año 2006 dice que este tipo de infracción puede incurrir entre cuatro (4) y ocho (8) años de cárcel y una multa de entre aproximadamente veintiséis (26) y mil (1000) salarios mínimos legales mensuales vigentes (Ley N.º 1030, 2006). Esta ley es aplicada en muy pocos casos por lo cual el uso de las copias digitales no autorizadas sigue siendo el principal recurso para el estudio de la filosofía.

En el Grupo de investigación Filosofía y enseñanza de la filosofía, por su parte, se ha asumido como hipótesis de varias de sus investigaciones que así como el conocimiento y aprendizaje de habilidades como la lógica, la argumentación y el discernimiento moral se constituyen en elementos que habilitan a los sujetos para el ejercicio de la ciudadanía, en cuento los dota de herramientas para defender sus derechos, así el desconocimiento de tales habilidades (tanto del proceso como de su importancia en el ejercicio de la ciudadanía) lleva a los ciudadanos al

desconocimiento de los derechos de los otros, en cuanto se privilegia la voluntad propia sobre la necesidad de convivir por vía de la argumentación. Estas habilidades se constituyen, a su vez, en las estructuras cognitivas básicas del filosofar; no reconocer su relevancia, entre otros aspectos, implica no reconocer la relevancia de la filosofía, de sus contenidos y de su necesidad en la convivencia ciudadana, en el ejercicio de la filosofía. Ese parece ser el estado actual de una Colombia que se pregunta ¿Dónde están los filósofos?

Es por lo anterior que surge el interés de conocer qué tan importantes son los filósofos en la sociedad, qué rol han tenido en hechos importantes de la historia colombiana y cómo se pueden utilizar las tecnologías de la información para darle respuesta a estos interrogantes.

2. Justificación

La filosofía no es visible ante los ojos de la sociedad porque hay desconocimiento sobre cuál es el significado de la filosofía y la acción de filosofar, el campo de acción de los profesionales de esta disciplina y su importancia e impacto en contextos importantes, razones por la cual en este proyecto se decidió aprovechar las herramientas tecnológicas disponibles en la actualidad para desarrollar un entorno virtual de aprendizaje donde se muestran de manera didáctica todas las evidencias y aportes en diferentes contextos de la sociedad que demuestran el rol y la importancia de los filósofos en eventos importantes de la historia colombiana.

Por medio de la herramienta que aporta este proyecto se busca contribuir a eliminar muchos de los prejuicios relacionados a esta área del conocimiento, entre ellos, demostrar que la profesión de la filosofía no es solo es exclusiva para los hombres, sino que las mujeres también tienen grandes

representantes que han tenido gran influencia en la sociedad a lo largo de la historia. También se proporciona una herramienta de enseñanza y aprendizaje para los docentes, estudiantes y cualquier ciudadano que desee conocer más sobre esta área del conocimiento y de sus referentes a nivel nacional. Se espera que con esta herramienta los filósofos se animen a participar de manera más activa en los medios de comunicación, en las discusiones públicas de las problemáticas del país y que se familiaricen con más herramientas tecnológicas como la presentada en este proyecto para realizar sus investigaciones, proyectos y socialización con otros profesionales, de manera que su actividad e importancia para el país sea más visible ante los ojos de la sociedad y del gobierno, evitando así posibles recortes de presupuesto para su enseñanza y otras acciones gubernamentales que puedan afectar de manera negativa a la filosofía.

Las herramientas que se pueden encontrar en internet con las cuales se realizan las búsquedas de recursos bibliográficos están enfocadas en la producción escrita de los filósofos, lo que se propone en este proyecto y que lo diferencia de los demás es que está enfocada en los filósofos, destacando los aportes (con sus respectivas evidencias, las cuales pueden ser de diferentes tipos: libros, artículos, ponencias, conferencias, videos, entre otros) que hicieron en periodos relevantes a lo largo de su vida en dos líneas de tiempo relacionadas entre sí. De esa manera se consigue mostrar qué tan importante ha sido el papel de los filósofos en diferentes contextos sociales.

3. Objetivos

3.1 Objetivo general

Desarrollar un prototipo de entorno virtual que permita visualizar las evidencias de las acciones sociales concretas, investigaciones y contribuciones de filósofos en diferentes contextos sociales en Colombia.

3.2 Objetivos específicos

- Realizar la especificación de los requerimientos y el diseño de la arquitectura necesaria para el entorno virtual.
- Desarrollar un módulo que muestre la evidencia de los filósofos categorizada por el contexto social y entrelazada hipertextualmente entre sí con las evidencias de otros filósofos
- Desarrollar un módulo que permita gestionar los elementos que hacen parte de las evidencias de los filósofos
- Realizar un plan de pruebas para verificar el correcto funcionamiento del entorno virtual.
- Instalar en un servidor para su uso público con fines de evaluación tecnológica y pedagógica por parte del grupo de investigación FiloEn.

4. Marco de antecedentes

El estudio de la filosofía requiere de mucha lectura para poder conocer las posturas de los autores sobre distintos campos de la filosofía y para desarrollar habilidades como el uso de la lógica, la argumentación, etc. Para lo anterior se necesitan recursos como libros, vídeos y artículos de revistas, pero debido a que hay muy pocas existencias físicas en las bibliotecas disponibles para préstamo y el costo elevado de adquisición en librerías los estudiantes tienen que recurrir a fuentes más económicas como fotocopias o internet para encontrar dichos materiales.

Teniendo en cuenta que generalmente para los estudios que se realizan en filosofía es más importante la producción literaria de un personaje (libros, escritos, artículos, etc.) que su biografía, se realizó una búsqueda de sitios web, blogs y plataformas educativas en las cuales se pudiera encontrar la producción literaria más relevante de diversos filósofos, principalmente colombianos, ya que el contexto y el alcance de este proyecto están limitados al territorio nacional.

Se pueden encontrar fácilmente sitios web institucionales y blogs independientes dedicados a un filósofo en particular, en los cuales se encuentra la biografía y en algunos casos la recopilación de algunas obras relevantes, como es el caso del sitio web proporcionado por la Universidad del Valle dedicado al filósofo Estanislao Zuleta (Facultad de humanidades Universidad del Valle, 2006) en el cual se puede encontrar su biografía, ponencias y escritos sobre temas como la universidad pública, la guerra, la educación, entre otros.

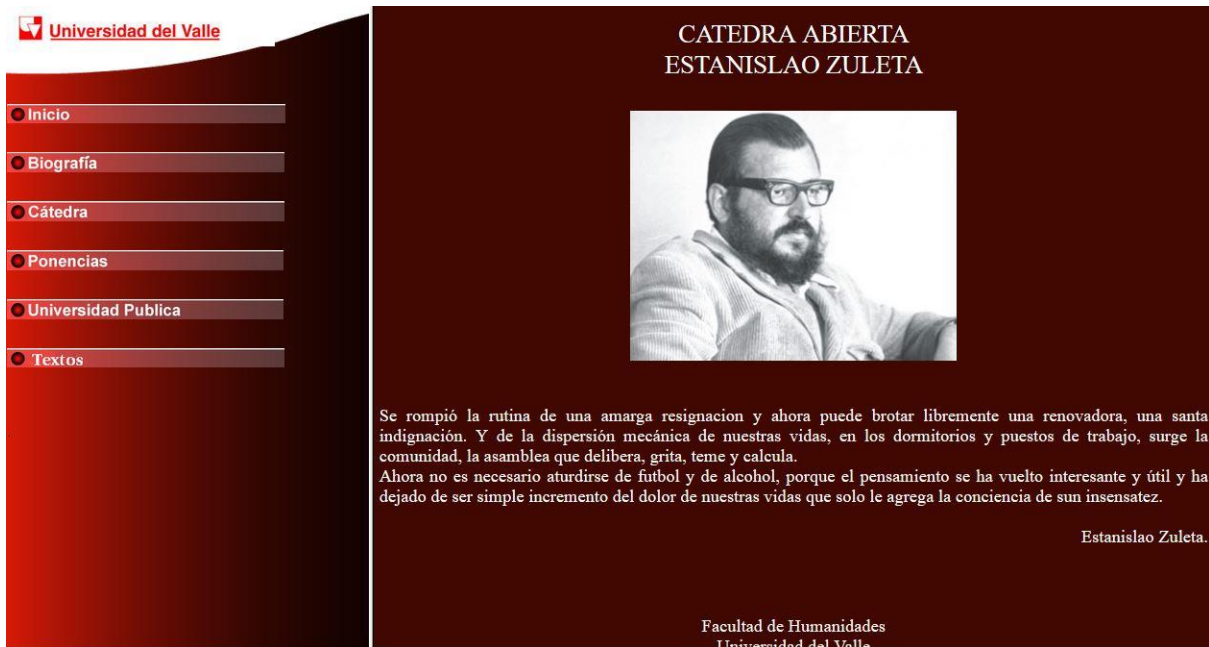


Figura 1. Sitio web dedicado a las obras de Estanislao Zuleta. (Facultad de humanidades Universidad del Valle, 2006).

Para encontrar recursos bibliográficos de otros filósofos colombianos reconocidos como Edgar Garavito, Daniel Herrera Restrepo, Danilo Cruz Vélez, Beatriz Restrepo Gallego, Carlos Enrique Restrepo, entre otros, existen plataformas que funcionan como repositorios de revistas, libros y otros recursos bibliográficos entre los cuales se pueden destacar los siguientes:

- Dialnet (Universidad de la Rioja, 2001), un portal bibliográfico de acceso libre proporcionado por la Universidad de la Rioja que tiene como principal objetivo darle mayor visibilidad a la literatura científica hispana, se centra fundamentalmente en las ciencias humanas, jurídicas y sociales.

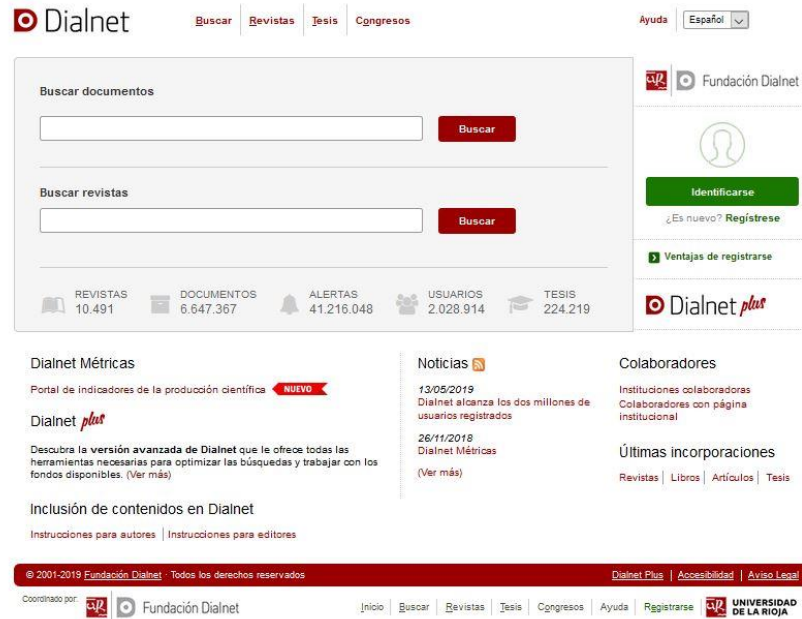


Figura 2. Página principal del portal Dialnet. (Universidad de la Rioja, 2001)

- SciELO (Fundacion para el apoyo a la investigacion del estado de Sao paulo [FAPESP], 1998) biblioteca científica electrónica en línea que facilita el acceso a colecciones de revistas y libros de todas las áreas del conocimiento procedentes de países latinoamericanos.

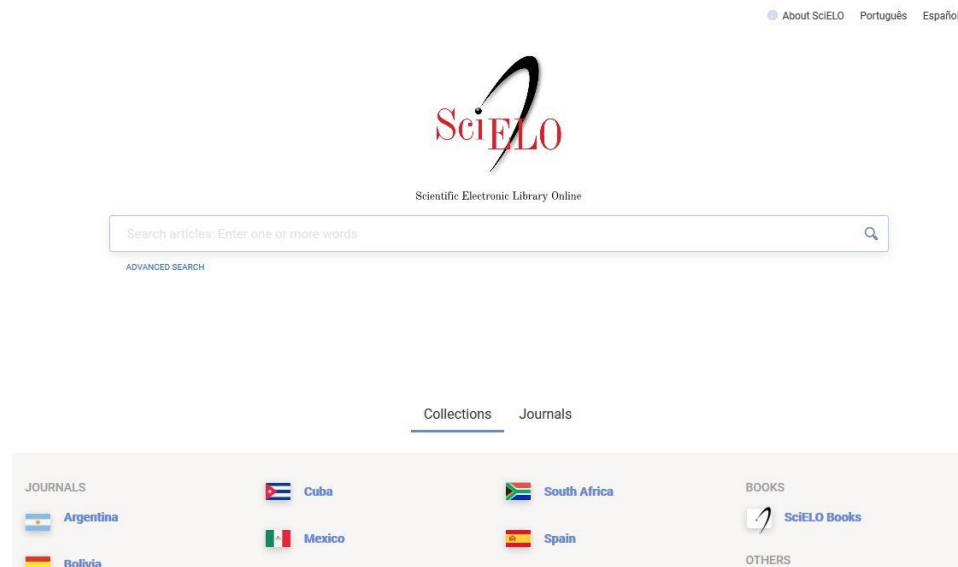


Figura 3. Página principal de la biblioteca Scielo. (Fundacion para el apoyo a la investigacion del estado de Sao paulo [FAPESP], 1998).

- Banrepcultural (Banco de la República de Colombia, 1997) la biblioteca virtual más antigua de América Latina en la cual se pone a disposición de los usuarios material bibliográfico principalmente referente a la cultura colombiana y de autores colombianos.

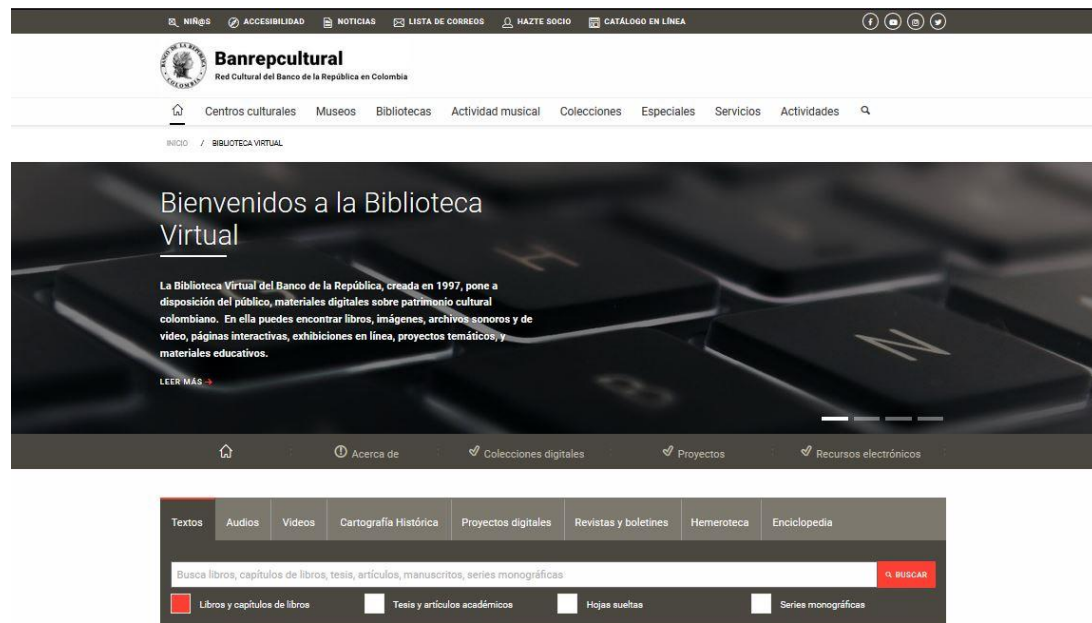


Figura 4. Biblioteca virtual del banco de la república de Colombia. (Banco de la República de Colombia, 1997)

- Redalyc (Universidad Autónoma del Estado de México, 2003) Red de Revistas Científicas de América Latina y el Caribe, España y Portugal es un proyecto para la difusión de revistas científicas realizado por la Universidad Autónoma del Estado de México junto a otras instituciones educativas, editoriales y centros de investigación.



Figura 5. Página principal del Sistema de Información Científica Redalyc. (Universidad Autónoma del Estado de México, 2003)

- Webdianoia (Fouce, 2001) Sitio web dirigido principalmente a estudiantes de secundaria en el cual se puede encontrar material de estudio sobre referentes mundiales de la filosofía a lo largo de toda la historia, como lo son Nietzsche, Marx, Descartes y muchos otros más. De cada autor se puede encontrar su biografía, obras, cronología, textos y ejercicios didácticos sobre los temas tratados por los filósofos.



Figura 6. Sección de filósofos de la página Webdianoia. (Fouce, 2001)

En la Universidad Industrial de Santander, Acosta García y Torres Gaona (2016) elaboraron como trabajo de grado una biblioteca digital enfocada a la recopilación de recursos bibliográficos orientados a temas filosóficos, en su proyecto que tiene como título “Herramienta web para la construcción colectiva de una biblioteca digital de recursos electrónicos sobre clásicos de la filosofía” identificaron el problema del alto costo y la falta de recursos bibliográficos para el estudio de filosofía para lo cual desarrollaron una herramienta web en la cual la comunidad de estudiantes y profesores pueden construir en forma comunitaria una biblioteca virtual con los recursos de interés subiendo los documentos o enlaces a otros sitios web, siempre respetando los derechos de autor.

Carrillo Garavito (2014) realizó como proyecto de grado un archivo digital para el filósofo colombiano Daniel Herrera Restrepo, en este se puede encontrar información relevante sobre su aporte a la filosofía, su biografía y una recopilación del material de su autoría dirigido hacia el público. Dicho proyecto, que tiene como título “Archivo digital para filósofo colombiano”, es un buen avance para facilitar el estudio de las obras y aportes a la comunidad filosófica y mejorar la presencia de los filósofos en medios de comunicación, como en este caso el internet.

Gomez Nossa y Hernández Serrano (2015) en su trabajo de grado titulado “Sistema de información para administrar objetos bibliográficos en archivos digitales” realizaron una labor de ingeniería inversa en el proyecto de Carrillo Garavito nombrado anteriormente, con el objetivo de implementar un sistema de información que administrara los objetos bibliográficos almacenados en archivos digitales de una manera más óptima realizando mejoras en los módulos e interfaces existentes.

Las herramientas y proyectos de grado anteriormente expuestos demuestran que durante los últimos años ya se han evidenciado problemáticas muy similares a la expuesta en este proyecto a

las cuales se les ha propuesto como solución una herramienta tecnológica. Cabe resaltar que los estudios enunciados previamente aportan significativamente al desarrollo de este proyecto puesto que fueron guía al momento de definir la solución propuesta y en el análisis de la problemática.

5. Marco conceptual

5.1. Fundamentos teóricos

5.1.1. Ingeniería Ontológica. La palabra Ontología proviene del griego, idioma en el cual significa “estudio del ser”. Es un término que tiene muchas interpretaciones a lo largo de la historia por muchos filósofos reconocidos como Tales de Mileto y Anaxímenes de Mileto quienes definieron conceptos claves con los cuales más adelante Pitágoras descubriría lo que ahora es la ciencia de la Ontología. En el diccionario “Webster’s Third New International Dictionary” podemos observar dos definiciones para el término de Ontología, la primera definición, dada por Fernández Breis (2013), tiene un significado de naturaleza filosófica y la segunda tiene un significado más terminológico.

“Ciencia o estudio del ser: específicamente, una rama de la metafísica relacionada con la naturaleza y las relaciones del ser; un sistema particular según el cual se investigan los problemas de la naturaleza del ser; esto es, filosofía fundamental.

Teoría relativa a los tipos de entidades y específicamente los tipos de entidades abstractas que se admiten en el lenguaje de un sistema” (p. 24).

Hace varias décadas el término de Ontología tomó especial interés en el campo de la Inteligencia Artificial donde es más común que se utilice la segunda definición mostrada anteriormente y con los años ha adquirido un nuevo impulso debido a que los científicos en esta

área se encontraron frente a la tarea de imitar el comportamiento y razonamiento humano en sistemas y se dieron cuenta de que la formalización del conocimiento humano es importante para cualquier sistema inteligente así como para los filósofos intentan revelar aspectos comunes de las cosas para entenderlas. En la actualidad este término también está siendo utilizado en el campo de la web semántica, la cual propone describir los recursos de la web con representaciones que puedan ser entendidas por programas y poder construir una web más cohesionada, donde localizar, compartir e integrar información y servicios sea más fácil (Castells, 2003).

Bravo Contreras, Pavón Gutiérrez, Reyes Ortiz, & Alcántara Ramírez (2018) definen la ingeniería ontológica como el proceso en el cual se realizan actividades concernientes a los métodos y metodologías para construir ontologías, sistemas de ontologías y las herramientas y lenguajes que las soportan. Le corresponde a este proyecto utilizar las ontologías obtenidas de la reconstrucción de la evidencia de los filósofos para desarrollar un entorno virtual en el cual las evidencias se relacionen entre sí y esté estructurada de manera que sea fácil medir el impacto de los autores.

5.1.2. Entornos virtuales. El auge de las tecnologías de información y comunicación en las últimas décadas ha permitido que la educación tenga un mayor alcance con herramientas como las plataformas de formación virtuales permitiendo que la educación no solo sea presencial, en las plataformas se encuentra una combinación de recursos, interactividad, apoyo y aprendizaje estructuradas de manera que los estudiantes puedan adquirir los conocimientos propuestos sin necesidad de estar presentes en un aula de clase.

Belloch (s.f, p.1) afirma que plataforma de formación virtual es el nombre genérico para el software utilizado por alguna institución para la educación virtual y toma distintos nombres según la finalidad y sus características, los más conocidos son los siguientes:

- Sistemas de gestión de contenidos (Content Management System, CMS).
- Sistemas de gestión del conocimiento (Learning Management System, LMS), que también son conocidos como Virtual Learning Environment (VLE) o Entornos Virtuales de aprendizaje (EVA).
- Sistemas de gestión de contenidos para el conocimiento o aprendizaje. (Learning Content Management System, LCMS).

De los nombrados anteriormente el más adecuado para la educación virtual es un entorno virtual de aprendizaje debido a que agrupan las características más importantes de los demás tipos de plataformas educativas por lo que tienen más herramientas y ventajas para la enseñanza a distancia, algunas de ellas son:

- Se facilita la distribución del material de estudio debido a que se encuentra de manera digital, por lo tanto, hay mayor soporte y recursos para las actividades académicas de los docentes y los estudiantes.
- Está ubicado en la red y se puede acceder a su contenido mediante cualquier dispositivo con conexión a internet.
- Existen muchas herramientas para realizar seguimiento, evaluación y control a los estudiantes, también se facilita la comunicación e interacción entre los docentes y estudiantes lo cual da como resultado una relación mucho más personal que facilita a los docentes ver de primera mano el proceso de aprendizaje de cada uno de sus estudiantes.
- Además de los recursos bibliográficos, el EVA proporciona a los estudiantes más herramientas didácticas que facilitan su aprendizaje.

- Hyraldo Trejo (2013) manifiesta otra ventaja de los entornos virtuales de aprendizaje, permite que los estudiantes desarrollen la capacidad de autoaprendizaje y de ser autodidactas, aptitudes que son muy importantes en la actualidad donde hay tanta información disponible.

El uso de los entornos virtuales para la educación virtual también se justifica por ciertos fenómenos socioculturales que están ocurriendo en lo que ahora se llama sociedad de la información, el cual es un entorno social que ha surgido en las últimas décadas debido a una gran evolución en las tecnologías, específicamente en las tecnologías para la información y comunicación, que ha provocado la digitalización de la información (Salinas, s.f.). En esta nueva sociedad la nueva generación de jóvenes, los estudiantes son “nativos digitales” por lo cual presentan una gran facilidad para relacionarse con la tecnología lo cual puede ser aprovechado por los entornos virtuales para la enseñanza ya que es un ambiente en el cual los estudiantes se sienten cómodos.

5.1.3. Framework web. En términos generales un framework define una estructura con la cual se va a construir una solución frente a un problema determinado. En el caso de un framework web el problema corresponde al desarrollo de una aplicación web (Sitios web, intranets, etc.), para facilitar el desarrollo de esta solución se establecen ciertas reglas para escribir y estructurar el código fuente, las librerías, archivos de configuración, etc. de manera que se pueda aplicar un patrón de diseño con mayor facilidad, también se agregan nuevas funcionalidades al lenguaje de programación y se automatizan procesos lo que resulta en un menor tiempo de desarrollo.

Generalmente en este ámbito se utilizan frameworks para aplicar patrones de diseño que separan la presentación (la que interactúa con el usuario) y la lógica (la que corresponde a la gestión de los datos), estos también son llamados Frontend y Backend respectivamente. Para el desarrollo de este

proyecto se utilizó el framework Vue.js para el Frontend y Laravel para el Backend, se explicará el funcionamiento de cada uno de estos frameworks más adelante. (Martínez Villalobos, Camacho Sánchez, & Biancha Gutiérrez, 2010).

5.1.4. MVVC. EL patrón de arquitectura MVVC, por sus siglas en inglés Model-View-ViewModel o en español Modelo-Vista-Modelo de Vista, es una variación del patrón de diseño MVC (Modelo-Vista-Controlador) generalmente usado para el desarrollo de las interfaces software modernas, debido a que simplifica la tarea de desarrollo y facilita el mantenimiento del código fuente.

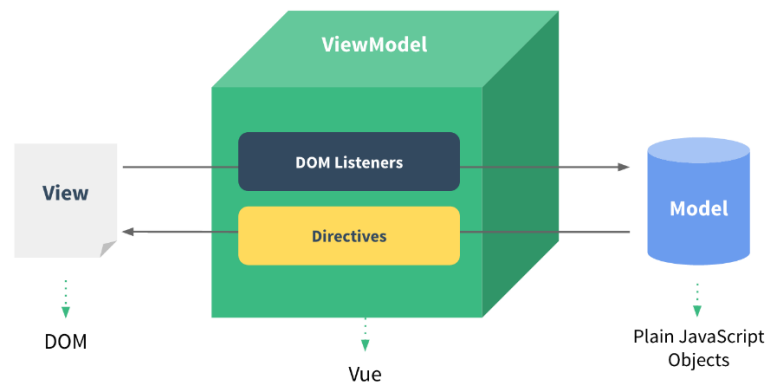


Figura 7. Patrón MVVM implementado en Vue.js. Adaptado de (You, 2014)

El principal objetivo de este patrón de diseño es separar la capa del modelo y la capa de la vista introduciendo una capa intermedia llamada modelo de vista que cumple la función de comunicar las otras dos capas, en la Figura 7 podemos ver un ejemplo de aplicación del patrón con el framework Vue.js.

- **Modelo:** Es la capa en la que se representan como objetos los elementos relevantes para la lógica del negocio. En dichos objetos no hay necesidad de implementar funciones con las cuales se puedan administrar los atributos ya que estos están vinculados a la capa de

vista y cualquier cambio se verá reflejado automáticamente en la vista, esto es lo que caracteriza a los frameworks reactivos.

- **Vista:** Es la capa encargada de definir la estructura, la apariencia y el comportamiento de los datos que genera la aplicación, y a diferencia de otros patrones, también cumple un papel activo interactuando con el usuario por medio de eventos. Está relacionada directamente con la capa de modelo de vista por medio de “data bindings” o enlaces de datos.
- **Modelo de vista:** La capa de modelo de vista es la encargada de coordinar la interacción entre la capa de vista y la capa de modelo donde se encuentra los objetos con la información relevante para la aplicación. Entre sus responsabilidades principales se encuentran: controlar las interacciones del usuario con la vista, trasladar los datos ingresados por parte del usuario al modelo implementando en algunos casos una capa de validación de datos y notificar a la vista de los cambios realizados en el modelo.

En la Figura 8 se pueden ver las tres capas nombradas anteriormente con sus respectivas interacciones.

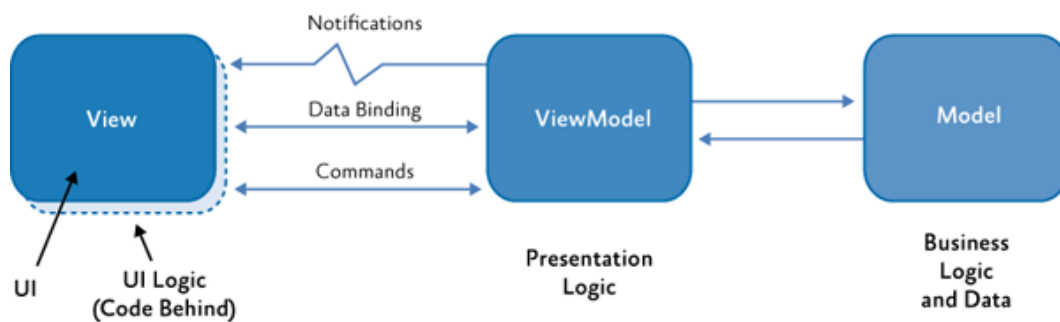


Figura 8. Capas del patrón MVVM. Adaptado de (Microsoft Corporation, 2014)

5.1.5. MVC. El patrón MVC (Modelo-Vista-Controlador) es un patrón de arquitectura utilizado para el desarrollo de software que propone la separación de las distintas funcionalidades de un software en tres capas, la capa del modelo, la de vista y la del controlador. Esta separación

le proporciona al software ventajas como mayor escalabilidad de las aplicaciones, facilita la funcionalidad y mantenimiento y evita lo que se denomina código espagueti, que ocurre cuando el código fuente de una aplicación carece de una estructura y modularidad comprensible.

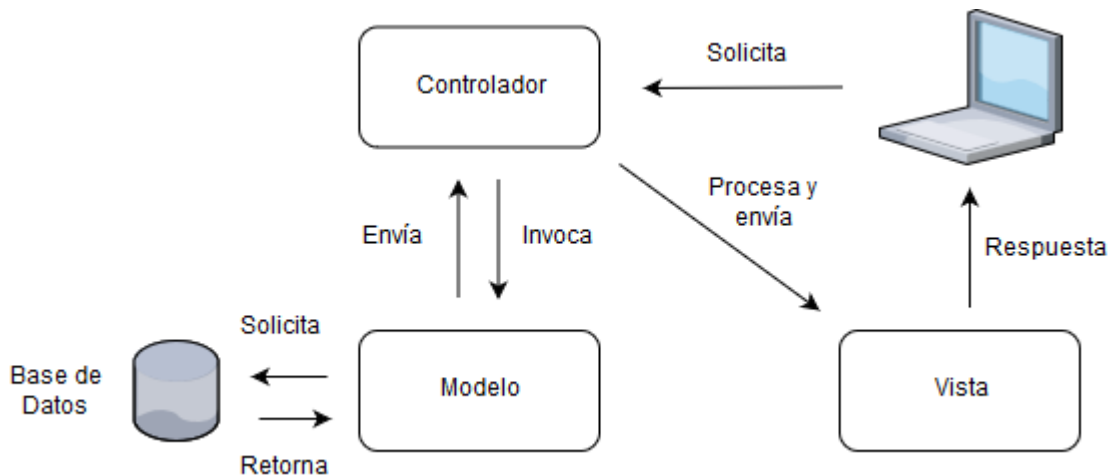


Figura 9. Capas del patrón MVC.

En la figura 9 se pueden ver los tres niveles de abstracción propuestos por el patrón MVC:

- **Modelo:** Es la capa donde se gestiona la lógica del negocio correspondiente a la aplicación, en esta se proporcionan mecanismos para acceder y solicitar la información almacenada en la base de datos, dicha información está representada internamente como objetos con sus respectivos atributos y métodos.
- **Vista:** Es la encargada de mostrar al usuario la información solicitada correspondiente a la lógica del negocio generalmente por medio de interfaces gráficas en lenguaje HTML o XML.
- **Controlador:** Es la capa que se encarga de capturar los eventos generados por el usuario para luego realizar las respectivas solicitudes de datos al modelo y entregar los resultados a la vista.

Según Bahit (2011) el funcionamiento básico del patrón MVC puede resumirse en los siguientes pasos:

- El usuario realiza una interacción con un elemento mostrado en la vista.
- El controlador registra el evento.
- Realiza una llamada a los modelos correspondientes solicitando los datos necesarios.
- El modelo se encarga de interactuar con la base de datos y devuelve la información necesaria al controlador.
- El controlador recibe la información por parte del modelo y la modifica en caso de que sea necesario, luego procede a entregar los datos a la vista.
- La vista se encarga de mostrar al usuario la información por medio de los elementos de la interfaz gráfica.

5.1.6. REST. REST por sus siglas en inglés Representational State Transfer (Transferencia de estado representacional) no es una tecnología, ni una arquitectura, es un conjunto de principios arquitectónicos con los cuales se desarrollan servicios web haciendo énfasis en los recursos del sistema, los cuales representan un concepto de negocio que puede ser accedido públicamente como, por ejemplo, un usuario con identificación 123. Se ha vuelto muy popular en los últimos años y ha desplazado protocolos como SOAP (Simple Object Access Protocol) e interfaces como WSDL (Web Services Description Language).

Según Zulian (2010), cualquier arquitectura de servicios web que cumpla con los siguientes cuatro principios puede considerarse una arquitectura REST:

- No mantiene estado: El cliente es el encargado de mantener el estado del recurso y no el servidor, en un servicio REST el servidor provee las operaciones necesarias para gestionar dichos recursos. Lo anterior resulta en una mejora del rendimiento y

simplificación en el diseño e implementación de los componentes del servidor ya que no es necesario sincronizar los datos con una aplicación externa.

- Utiliza los métodos HTTP de manera explícita: Para que la gestión de los recursos sea consistente se definen cuatro operaciones básicas: crear, actualizar, eliminar y editar, más comúnmente conocidas por el acrónimo CRUD (Create, Read, Update, Delete). A cada una de estas operaciones le corresponde un método HTTP de la siguiente manera:
 - POST: Se usa para agregar recursos y ocasionalmente para definir el estado inicial de dichos recursos.
 - GET: Se usa para obtener una representación del estado de un recurso almacenado en el servidor.
 - UPATE: Se usa para actualizar el estado de los recursos en el servidor
 - DELETE: Se usa para eliminar recursos por medio sus identificadores únicos
- Expone URL's como forma de directorios: Los clientes acceden a los recursos por medio del URI, por sus siglas en inglés Uniform Resource Identifier (Identificador de recursos uniforme), los cuales son representados como directorios con una estructura similar a un árbol en el cual existe una raíz que se va abriendo en distintas ramas a través de diferentes subrutinas hasta encontrar un recurso en específico. La estructura propuesta tiene la siguiente forma: `http://paginaweb/raiz/rama/{identificador}`. Estas direcciones también son llamadas endpoints o puntos finales.
- Transfiere información en formatos XML, JSON (JavaScript Object Notation) o ambos: Se establece un formato de datos para el intercambio de información entre el servidor y el cliente, este formato de datos debe ser capaz de representar el estado de un recurso.

Eso permite que el servicio pueda ser usado en varios clientes escritos en diferentes lenguajes de programación, los formatos más usados son XML y JSON.

5.1.7. SPA (Single Page Application). Una Single Page Application o en español Aplicación de una sola página, es un sistema web que inicialmente descarga una versión básica de la aplicación que contiene recursos básicos para su funcionamiento como imágenes, información estática y la estructura básica de la aplicación en forma de componentes, de manera que solo se vuelven a hacer solicitudes al servidor para traer información como nuevo contenido o nuevos componentes dinámicamente, sin recargar la página, lo que resulta en una mejor experiencia de usuario.

En la siguiente definición, realizada por Jadhav, M. A, Sawant, B. R, & Deshmukh, A. (2015) se explican las cuatro principales características de una SPA.

Una Single Page Application (SPA) está compuesta de componentes individuales que pueden ser reemplazados/actualizados independientemente, sin recargar/refrescar completamente la página de manera que la página no necesita ser recargada con cada acción del usuario.

1. Componentes individuales: La página entera está dividida en otros componentes más pequeños que interactúan entre ellos.
2. Reemplazar/Actualizar: Un componente/región/parte de una página se actualiza/recarga con cada cambio requerido por el usuario.
3. Refrescar/Recargar: La página entera nunca se actualiza/recarga, sino que se carga nuevo contenido en alguna parte o sección según los nuevos datos.
4. Acciones de usuario: una SPA es responsable por manejar rápidamente todas las interacciones de usuario con la interfaz tales como clic en botones, entrada de nuevos datos desde el teclado, etc. de manera que la interfaz de usuario sea muy fluida. (Jadhav, Sawant, & Deshmukh, 2015, págs. 2876-2879).

5.2. Tecnologías involucradas en el desarrollo del proyecto

5.2.1. HTML5. Hypertext Markup Language o lenguaje de marcado de hipertexto, es un lenguaje creado por Tim Berners-Lee en el año 1989 para definir la estructura de los documentos alojados en internet y enlazarlos con otros documentos. La estructura de los documentos está diferenciada y definida por etiquetas, una de apertura y otra de cierre. Cada etiqueta está compuesta por una palabra clave rodeada de los símbolos mayor y menor, y puede contener atributos, los cuales están definidos dentro de la etiqueta de apertura. Ejemplo: `<html lang="es"></html>`, en este caso la palabra clave es 'html' y tiene un atributo llamado 'lang' con valor 'es', nótese que en la etiqueta de cierre siempre se escribe una barra diagonal después del símbolo menor que.

Con el tiempo se lanzaron nuevas versiones del lenguaje agregando nuevas etiquetas y mejorando la compatibilidad con los diferentes navegadores hasta llegar a la versión número 5, donde se definieron los estándares con los cuales se estructuran las páginas web y las funciones que realizan cada una de las tecnologías involucradas en su desarrollo, siendo estas tecnologías HTML, CSS y JavaScript, (Gauchat, 2012). Actualmente este estándar está definido por la W3C (World Wide Web Consortium).

5.2.2. CSS3. Cascading Style Sheet o hojas de estilo en cascada, es un lenguaje utilizado para darle estilos visuales a los elementos de un documento definidos por html y separar en distintos archivos la definición de la estructura y la declaración de los estilos. (Herrera Ríos, 2011). Su historia se remonta a los inicios del internet cuando el organismo W3C, propuso crear un lenguaje de hojas de estilo para el lenguaje HTML con el fin de darle estilos visuales a los elementos definidos en la estructura del documento, de las propuestas que se hicieron solo se tomaron en cuenta dos: CHSS (Cascading HTML Style Sheets) y SSP (Stream-based Style Sheet Proposal), propuestos por Håkon Wium Lie y Bert Bos respectivamente, personajes que decidieron unir

esfuerzos y definieron un lenguaje que tuviera lo mejor de las dos propuestas, hasta el momento se han presentado tres versiones del lenguaje, siendo CSS3 la más actual. (Uniwebsidad, s.f.).

5.2.3. Vue.js. Vue.js es un framework de JavaScript de código abierto utilizado para el desarrollo de interfaces de usuario de aplicaciones web. Su principal característica es que está diseñado para ser usado de manera progresiva, permitiendo la escalabilidad de los proyectos. Su librería principal provee funcionalidades básicas para la realización de un proyecto de baja complejidad como lo puede ser un Landing page y en caso de ser necesario se pueden agregar otros módulos o librerías para resolver necesidades de mayor complejidad que requieran de funcionalidades extra. También se puede destacar que Vue.js es un framework reactivo y basado en componentes que permiten modularizar la aplicación en varios componentes reutilizables.

Su funcionamiento interno de componentes reactivos está basado en el patrón de diseño MVVM (Model -View-ViewModel), específicamente en la capa de ViewModel, de manera que conecta en las dos direcciones la capa de View y la de Model. La manipulación del DOM (Document Object Model) es realizada mediante directivas, las cuales son definidas en los atributos de las etiquetas HTML, y mediante filtros

5.2.4. Laravel. Laravel es un framework de código abierto utilizado para desarrollar aplicaciones y servicios web en el lenguaje PHP. Fue creado por Taylor Otwell en el año 2011 y tiene muchas influencias de otros frameworks conocidos como Symfony y CodeIgniter, de los cuales intenta aprovechar todas las ventajas y características que los hacen destacar. Es un framework modular y extensible que provee gran variedad de paquetes que agregan nuevas funcionalidades y agilizan el desarrollo de aplicaciones web, de las cuales se pueden destacar las siguientes:

- **Blade:** Es un sistema de plantillas que facilita la creación de las interfaces de la aplicación. Permite a los desarrolladores insertar instrucciones en PHP dentro del código que define la estructura de una vista, el cual está escrito en HTML. Se puede integrar fácilmente con otros frameworks enfocados en el desarrollo de interfaces como el utilizado en este proyecto, Vue.js.
- **Eloquent:** Eloquent es un ORM (Object-Relational-Map) que simplifica el manejo de las bases de datos, se puede integrar fácilmente con sistemas gestores de bases de datos como MySQL y PostgreSQL.
- **Routing:** Es un sistema de enrutamiento que permite relacionar las vistas y controladores de nuestra aplicación con las rutas que el usuario introduce en el navegador.
- **Middlewares:** Son controladores que se encargan de analizar las peticiones HTTP realizadas al servidor con el fin implementar aumentar la seguridad y evitar problemas como por ejemplo XSS (Cross-Site-Scripting).

Laravel implementa el patrón de arquitectura MVC (Modelo-Vista-Controlador), que es uno de los patrones más usados por los desarrolladores, esto hace que el mantenimiento del proyecto sea fácil (Saunier, 2014). Además, tiene una comunidad muy grande de profesionales que le da soporte al framework, realiza documentación y efectúa mejoras constantemente.

5.2.5. INSOMNIA. Insomnia es un cliente REST que permite a sus usuarios realizar peticiones o solicitudes HTTP a una API REST mediante formatos de texto utilizados para el intercambio de datos como lo son JSON y XML. Provee una interfaz gráfica que facilita crear los formularios, modificar el tipo de solicitud HTTP y editar los endpoints o URL'S. También provee funcionalidades como personalización de la interfaz, atajos de teclado, configuración proxy,

variables generales, ayudas para la autenticación e importar datos de otros clientes REST como POSTMAN, HAR, etc.

5.2.6. MySQL. MySQL es un sistema de gestión de bases de datos relacionales de código abierto basado en SQL (Lenguaje de consulta estructurado) generalmente utilizado en el desarrollo de aplicaciones web por muchos desarrolladores en todo el mundo. Cualquier persona puede utilizar MySQL bajo la licencia de uso GPL (General Public License), pero empresas y corporaciones deben obtener una licencia comercial de Oracle, empresa que actualmente lo desarrolla.

Es uno de los más usados debido a que está disponible para los sistemas operativos más utilizados incluyendo Mac, Windows, Linux y Unix. Su característica más interesante es su versatilidad, ya que se pueden acceder a las bases de datos en un servidor multiusuario con diferentes lenguajes de programación. MySQL es un componente importante para varios de los sistemas de infraestructura de internet más utilizados para el desarrollo de aplicaciones dentro de los cuales se puede destacar LAMP llamado así por las tecnologías involucradas (Linux, Apache, MySQL, PHP), en este se usa Linux como sistema operativo, Apache como servidor web, MySQL como el gestor de bases de datos y PHP como el lenguaje de programación orientado a objetos. (Rouse, 2018).

Otras características de MySQL que se pueden destacar son:

- Es uno de los gestores de bases de datos con mejor rendimiento y mayor velocidad al realizar las operaciones.
- No es necesario aprender nuevos comandos para interactuar con MySQL ya que se pueden usar los comandos SQL estándar.

- Provee de gran durabilidad y rendimiento gracias a que usa particiones de tablas y replicación de datos.
- Los usuarios pueden comunicarse con el servidor de MySQL por medio de diferentes protocolos de comunicación incluyendo sockets TCP/IP en cualquier plataforma.

6. Metodología

Una metodología de desarrollo software es un conjunto de actividades, métodos y modelos para el desarrollo de un producto, la metodología orientada a objetos que será la utilizada en este proyecto, ha evolucionado para ayudar a los desarrolladores a explotar el poder de los lenguajes de programación basados y orientados en objetos usando las clases y objetos como elementos básicos para la construcción del software. Weitzenfeld (2005, págs. 25-27), afirma que el uso de esta metodología presenta ciertas ventajas que mejoran la calidad del software, principalmente son la abstracción, modularidad, extensibilidad y la reutilización.

- La abstracción es el mecanismo con el cual se realiza una representación de los elementos más importantes de la realidad para poder utilizar sus funcionalidades y características por medio de un software y poder solucionar una determinada problemática. A medida que la representación de la realidad sea de mayor nivel, menor será el número de elementos necesarios para representar un sistema y más fácil será el manejo de la complejidad. La abstracción de los problemas se hace mediante objetos que poseen atributos, funciones y se relacionan con otros objetos, son la estructura

básica del sistema y permiten al programador organizar el software de manera que se vea simplificada la complejidad del problema.

- La modularidad permite a un sistema repartir las funcionalidades a diferentes componentes facilitando las operaciones y mantenimiento
- La extensibilidad corresponde a la facilidad para modificar el software durante el transcurso de su vida ya sea para agregar o modificar funcionalidades debido a cambios en los requerimientos. A medida que el proceso de abstracción se haya hecho a un alto nivel y el grado de modularidad sea mayor, la extensibilidad del sistema será más alta y se logrará un sistema que sea más fácil de modificar.
- La reutilización se basa en el aprovechamiento de definiciones similares de objetos dentro de la misma aplicación o incluso dentro de otras aplicaciones para simplificar funcionalidades, código y reducir tiempo y esfuerzos en el proceso.

Se han propuesto un gran número de metodologías para el análisis y diseño de software orientado a objetos de los cuales se pueden destacar la metodología de Booch (Booch, 1994), el proceso de desarrollo de software unificado de Jacobson (Rumbaugh, Jacobson, & Booch, 2000) y el OMT (Técnica de modelado de objetos) de Rumbaugh (Rumbaugh, Blaha, Premerlani, Eddy, & Lorensen, 1991), los autores de estos métodos unificaron la notación de sus respectivos métodos dando lugar al lenguaje de modelado conocido como UML (Rumbaugh et al, 2000). Tiempo después el Object Management Group (OMG) estandarizó este proceso en el año 1997 y Jacobson definió el Proceso unificado de desarrollo software (Rumbaugh et al, 2000) generalmente referenciado como Proceso unificado (UP) basándose en lo propuesto por Booch y Rumbaugh, metodología que servirá de guía para el desarrollo de este proyecto. El UP se define como un proceso dirigido por casos de uso, iterativo e incremental. Está basado en componentes y usa el

lenguaje UML para la representación de cada uno de los diagramas generados durante el desarrollo. El UP presenta cinco disciplinas de trabajo fundamentales:

- **Obtención de requerimientos:** Durante esta etapa el cliente y los desarrolladores se reúnen para establecer cuáles son las características del sistema en forma de requerimientos funcionales y no funcionales. A partir de los requerimientos se definen los actores del sistema y se realizan diagramas de casos de uso para describir todas las posibles interacciones entre los actores y el sistema para cada una de las funcionalidades establecidas.
- **Análisis:** En esta etapa los desarrolladores se basan en los resultados de la etapa anterior para producir un modelo de objetos que represente la solución al problema planteado. Este modelo de objetos consiste en objetos con atributos y relaciones con otros objetos y sirve para detectar posibles inconsistencias en la etapa anterior, las cuales deben ser resueltas junto con el cliente y si es necesario rediseñar los diagramas realizados.
- **Diseño:** Se divide en dos subetapas el diseño del sistema y el diseño de objetos. En el diseño del sistema se eligen las estrategias de implementación del sistema dividiéndolo en subsistemas para una mejor organización de las responsabilidades, también se seleccionan las tecnologías hardware y software en las cuales funcionará el sistema, la estrategia de almacenamiento de los datos persistentes y la política de control de acceso y el manejo de las condiciones de frontera. Durante el diseño de objetos se define de manera más específica cada una de las interfaces y componentes del sistema representados en objetos. Como resultado de esta etapa se obtiene un modelo de objetos mucho más detallado, comentado con restricciones y descripciones precisas para cada elemento (Bruegge & H. Dutoit, 2002).

- Implementación: En esta etapa se traduce el modelo de objetos obtenido en la etapa anterior a código fuente utilizando el lenguaje de programación orientado a objetos elegido. Lo anterior no debería suponer mucha dificultad ya que la codificación se realiza en base a los modelos planteados en las etapas anteriores los cuales también siguen el paradigma orientado a objetos. Para que un programa sea considerado a objetos debe cumplir con las siguientes características planteadas por Booch (Booch, 1994):
 - Los bloques lógicos en la estructura fundamental del programa deben ser objetos, no algoritmos.
 - Cada objeto es una instancia de alguna clase.
 - Las clases están relacionadas con otras clases por medio de relaciones de herencia.
- Pruebas: La etapa de implementación no está exenta de errores por lo cual es necesario realizar un plan de pruebas para verificar el funcionamiento del sistema y corregir los errores encontrados.

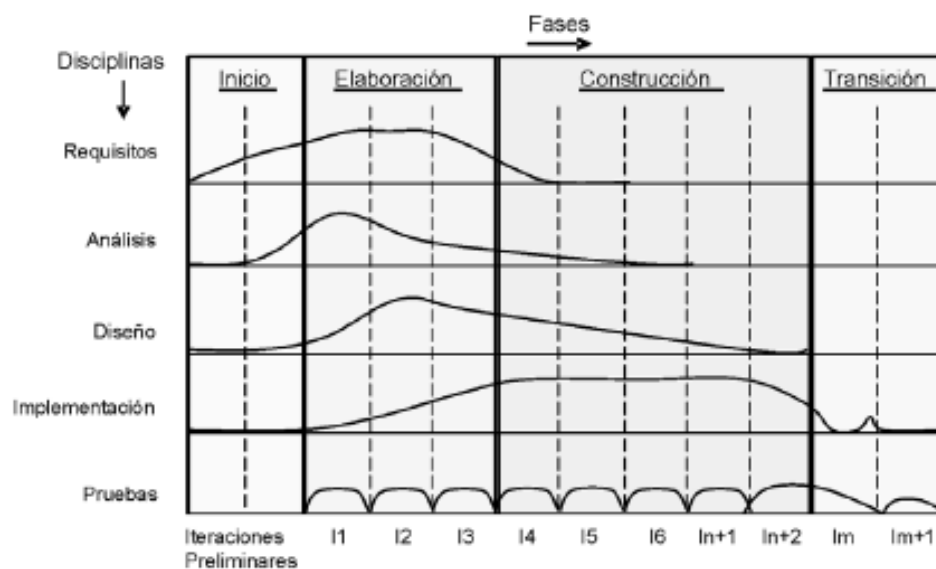


Figura 10. Disciplinas y fases fundamentales en el ciclo de vida del desarrollo de software. Adaptado de (Tabares, Barrera, Arroyave, & Pineda, 2007)

El ciclo de vida del desarrollo de software mostrado en la figura 10 también muestra cuatro fases principales en el proceso, cada una de las cuales también se divide en cierta cantidad de iteraciones que representan una realización funcional de un conjunto de casos de uso. Se dice que el proceso unificado es incremental debido a que en cada iteración se toma un nuevo conjunto de casos de uso para desarrollar y se añaden los de la iteración anterior.

En la fase de inicio se establece una visión del proyecto y su alcance, también se estima la viabilidad del proyecto. La fase de elaboración es donde se establece con claridad cuáles son los requerimientos de le proyecto y se presentan avances en los casos de uso y arquitectura del sistema. En la fase de construcción se desarrolla un producto que esté preparado para el uso por parte de los usuarios y finalmente en la etapa de transición el software se despliega en la comunidad de usuarios y se recibe una retroalimentación para realizar los ajustes necesarios, corregir errores no detectados anteriormente y finalizar algunas funcionalidades.

7. Resultados

7.1. Especificación de requisitos Software

La especificación de requisitos (ERS) se realizó con base en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830, 1998, y con el acompañamiento de la Doctora Sonia Cristina Gamboa Sarmiento PhD en Educación y los filósofos expertos, los cuales están encargados de realizar un proceso de curaduría sobre la

trayectoria intelectual de la muestra inicial de filósofos colombianos con los cuales se elaborará el entorno virtual de aprendizaje. En la siguiente tabla se pueden observar los filósofos colombianos y sus respectivos expertos.

Tabla 1. *Filósofos colombianos y sus correspondientes filósofos curadores.*

Filósofos colombianos	Expertos
Daniel Herrera Restrepo	Germán Vargas Guillén
Carlos Enrique Restrepo	Luz Gloria Cárdenas
Beatriz Restrepo Gallego	Judith Nieto

7.1.1. Propósito. El propósito de esta sección es definir los requerimientos funcionales y no funcionales necesarios para el desarrollo del entorno virtual de aprendizaje donde se podrán encontrar los aportes relevantes de los filósofos colombianos en la historia colombiana y de esa manera evidenciar su importancia en la sociedad.

7.1.2. Ámbito del sistema. El nombre del entorno virtual propuesto será DELFIC, que corresponde a una abreviación de la pregunta ‘¿Dónde están los filósofos en Colombia?’ que se hace en un artículo de la revista Arcadía, el cual fue una de las motivaciones para la realización de este proyecto.

Con este proyecto se espera darle mayor visibilidad a la importancia de los filósofos en la historia y en la sociedad colombiana mostrando los aportes realizados en distintos contextos sociales de manera didáctica en un entorno virtual de aprendizaje.

7.1.3. Perspectiva del producto. El entorno virtual está diseñado para trabajar en entornos web, lo que le permitirá ser accedido por medio de cualquier dispositivo con un navegador web. Su desarrollo se puede descomponer en dos partes, el Frontend, que constituye la parte del sistema con la que interactúa el usuario y será desarrollada con el framework Vue.js, y el Backend, que se

encarga de interactuar con la base de datos, se desarrollará con el framework Laravel y el Sistema de gestión de base de datos MySQL.

7.1.4. Funciones del producto. Las principales funcionalidades del entorno virtual de aprendizaje son:

7.1.4.1. Gestión de los usuarios. Por medio de un panel de administración, los usuarios que tengan cierto tipo de usuario podrán gestionar permisos y el acceso de otros usuarios administradores.

7.1.4.2. Gestión de los filósofos. El sistema proporciona un panel de administración en el cual los usuarios administradores podrán gestionar toda la información de los filósofos colombianos.

7.1.4.3. Gestión de los aportes de los filósofos. El sistema proporciona un panel de administración en el cual los usuarios administradores podrán gestionar toda la información correspondiente a los aportes de los filósofos y sus respectivas evidencias.

7.1.4.4. Perfiles de filósofos y de aportes. Cada filósofo colombiano tendrá su propio perfil dentro del entorno virtual de aprendizaje donde se mostrará de manera didáctica en diferentes líneas de tiempo sus periodos de vida más importantes y los aportes más relevantes en diferentes contextos sociales. También existirá un perfil para cada uno de los aportes, en el cual se mostrarán sus evidencias y otros filósofos y aportes relacionados.

7.1.5. Características de los usuarios. Existirán cuatro tipos de usuarios: visitante, administrador, curador y auxiliar, estos tienen diferentes niveles de experticia en el manejo de herramientas tecnológicas, por lo cual, el entorno virtual debe tener una interfaz intuitiva, fácil de comprender y sencilla de manejar.

7.1.6. Restricciones.

- Para el correcto funcionamiento del entorno virtual es necesario que el dispositivo con el cual el usuario accede al sistema tenga conexión a internet.
- La interfaz proporcionada debe funcionar para dispositivos móviles y de escritorio.
- Los servidores deben estar disponibles para atender consultas las 24 horas del día.
- Los lenguajes de programación utilizados en el desarrollo del entorno virtual son, HTML, CSS, JavaScript, PHP y MySQL. Con los frameworks Vue.js y Laravel.
- El entorno se diseñará siguiendo un patrón de arquitectura cliente-servidor.

7.1.7. Requerimientos funcionales

Tabla 2. *Requerimiento Funcional No. 01 – Registro de usuarios*

Requerimiento funcional No. 01	
Código	RF-01
Nombre del requerimiento	Registro a usuarios
Descripción	El sistema permitirá el registro a tres diferentes tipos de usuarios que podrán administrar el contenido del entorno virtual.
Restricciones	<ul style="list-style-type: none"> - Roles de usuarios: Administradores, curadores, auxiliares - Para que un usuario con el rol de administrador pueda registrarse es necesario que un usuario administrador que ya se haya registrado previamente acepte la solicitud de registro. - Para que un usuario con el rol de curador pueda registrarse es necesario que un usuario administrador que ya se haya registrado previamente acepte la solicitud de registro. - Para que un usuario con el rol de auxiliar pueda registrarse debe realizar una solicitud de registro a un curador en específico, el cual resolverá dicha solicitud de registro. - Un usuario no puede tener más de un rol - Solamente los usuarios con el rol de Curador tendrán auxiliares.
Prioridad	Alta

Tabla 3. *Requerimiento Funcional No. 02 – Autenticación de usuarios*

Requerimiento funcional No. 02

Código	RF-02
Nombre del requerimiento	Autenticación de usuarios
Descripción	Todos los usuarios tendrán que realizar una autenticación de usuario por medio de un correo electrónico y una contraseña para acceder al panel de administración.
Restricciones	<ul style="list-style-type: none"> - Los usuarios deben haber realizado el registro - Para el caso de un usuario auxiliar, un usuario administrador o curador debe haber aceptado la solicitud de registro previamente. - Para el caso de un usuario curador, un usuario administrador debe haber aceptado la solicitud de registro previamente.
Prioridad	Alta

Tabla 4. *Requerimiento Funcional No. 03 – Gestión de permisos a usuarios*

Requerimiento funcional No. 03	
Código	RF-03
Nombre del requerimiento	Gestión de permisos a usuarios
Descripción	Los usuarios con el rol de curadores y administradores podrán los diferentes permisos de otros usuarios en el entorno virtual por medio de un panel de administración.
Restricciones	<ul style="list-style-type: none"> - Los usuarios con el rol de administradores podrán revocar o permitir el acceso al entorno virtual de cualquier usuario que posea el rol de curador o auxiliar. - Los usuarios con el rol de administrador podrán gestionar los permisos de gestión de la información de un filósofo de cualquier usuario que posea el rol de curador o auxiliar. - Los usuarios con el rol de curadores podrán revocar o permitir el acceso al entorno virtual de cualquier usuario auxiliar que tenga. - Los usuarios con el rol de curadores podrán gestionar los permisos de gestión de la información de un filósofo de cualquier usuario auxiliar que tenga. - Los usuarios con el rol de auxiliar no tendrán permisos sobre otros usuarios.
Prioridad	Alta

Tabla 5. *Requerimiento Funcional No. 04 – Gestión de filósofos*

Requerimiento funcional No. 04	
Código	RF-04

Nombre del requerimiento	Gestión de filósofos
Descripción	Los usuarios podrán gestionar los filósofos que les correspondan dependiendo del rol que posean por medio del panel de administración
Restricciones	<ul style="list-style-type: none"> - Los usuarios con el rol de administrador podrán crear, editar, modificar y eliminar la información de cualquier filósofo en el entorno virtual. La información corresponde a su información básica, periodos de tiempo, fotografías, aportes, periodos y evidencias. - Los usuarios con los roles de curador y auxiliar podrán crear, editar, modificar y eliminar la información de cualquier filósofo sobre la cual posean permisos de gestión.
Prioridad	Alta

Tabla 6. *Requerimiento Funcional No. 05 – Perfiles de filósofos*

Requerimiento funcional No. 05	
Código	RF-05
Nombre del requerimiento	Perfiles de filósofos
Descripción	En el entorno virtual se encontrará un perfil para cada uno de los filósofos donde estará la información básica (biografía, contacto del curador, etc.). Los aportes y evidencias en los diferentes periodos relevantes de su vida se mostrarán en dos líneas de tiempo relacionadas entre sí.
Restricciones	<ul style="list-style-type: none"> - La línea de tiempo debe tener distintas opciones para filtrar la información mostrada. - El sistema debe permitir almacenar diferentes tipos de evidencia para los aportes (fotografías, documentos y enlaces a videos de YouTube). - El perfil debe tener una sección en la cual un usuario pueda realizar una búsqueda de las evidencias de los aportes del filósofo en cuestión. - Debe tener una sección en la cual se mostrarán sugerencias de aportes u otros filósofos dependiendo del aporte que se esté visualizando en la línea de tiempo.
Prioridad	Alta

Tabla 7. *Requerimiento Funcional No. 06 – Búsqueda de filósofos y aportes*

Requerimiento funcional No. 06	
Código	RF-06

Nombre del requerimiento	Búsqueda de filósofos y aportes
Descripción	El sistema tendrá un módulo en el cual se podrán realizar búsquedas de filósofos y aportes
Restricciones	- Debe tener distintas opciones para filtrar los resultados de las búsquedas
Prioridad	Alta

Tabla 8. *Requerimiento Funcional No. 07 – Mensajes de contacto*

Requerimiento funcional No. 07	
Código	RF-07
Nombre del requerimiento	Mensajes de contacto
Descripción	El sistema dispondrá de dos secciones, una para contactar a los administradores del sistema y otra para contactar a los curadores de cada uno de los perfiles de los filósofos
Restricciones	- El usuario tendrá que haber iniciado sesión
Prioridad	Alta

Tabla 9. *Requerimiento Funcional No. 08 – Editar información de usuario*

Requerimiento funcional No. 08	
Código	RF-08
Nombre del requerimiento	Editar información de usuario
Descripción	El sistema dispondrá de una sección para que los usuarios puedan editar su información básica como, nombre, contraseña, correo electrónico, etc.
Restricciones	- El usuario tendrá que haber iniciado sesión
Prioridad	Alta

7.1.8. Requerimientos no funcionales

Tabla 10. *Requerimiento No Funcional No. 01 – Validación de datos*

Requerimiento no funcional No. 01	
Código	RNF-01
Nombre del requerimiento	Validación de datos
Descripción	El sistema debe realizar una validación de datos para que no se introduzcan valores incorrectos en los campos al momento de crear o actualizar datos en los formularios del panel de administración y en el panel de registro.
Restricciones	<ul style="list-style-type: none"> - El sistema debe mostrar los errores de validación en los campos que contengan información errónea. - El sistema debe mostrar los campos obligatorios en cada uno de los formularios.
Prioridad	Alta

Tabla 11. *Requerimiento No Funcional No. 02 – Interfaz del sistema*

Requerimiento no funcional No. 02	
Código	RNF-02
Nombre del requerimiento	Interfaz del sistema
Descripción	El entorno debe tener una interfaz intuitiva estándar, que se adapte a las diferentes resoluciones de pantalla de diferentes dispositivos (móviles y de escritorio)
Restricciones	<ul style="list-style-type: none"> - La interfaz del sistema será desarrollada para la web.
Prioridad	Alta

Tabla 12. *Requerimiento No Funcional No. 03 – Accesibilidad del sistema*

Requerimiento no funcional No. 03	
Código	RNF-03
Nombre del requerimiento	Accesibilidad del sistema
Descripción	Los servidores deben en continuo funcionamiento para atender consultas debido a que el sistema está diseñado para la búsqueda de datos y el uso constante de los usuarios a cualquier hora del día.

Restricciones

Prioridad Alta

7.1.9. Actores del sistema. Como se puede observar en la figura 11, los actores del sistema son cuatro: visitante, administrador, curador y auxiliar.

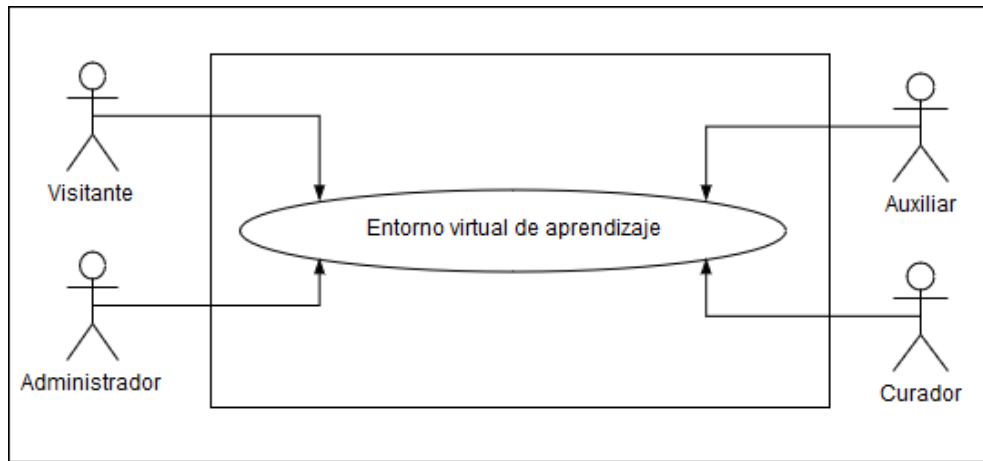


Figura 11. Actores del sistema.

7.1.10. Descripción de actores. En las siguientes tablas se observa una descripción más detallada de cada uno de los actores.

Tabla 13. Descripción de Actores – Visitante.

Actor	Visitante
Casos de uso	Enviar solicitud de registro, Contactar administrador, Iniciar sesión, Recuperar contraseña, Buscar información
Descripción	Cualquier usuario que navegue y consulte información en el entorno virtual sin iniciar sesión.
Tipo	Primario

Tabla 14. Descripción de Actores – Administrador.

Actor	Administrador
Casos de uso	Buscar información, Gestionar evidencia, Gestionar información de usuario, Gestionar periodo, Gestionar aporte, Gestionar filósofo, Gestionar acceso a auxiliares y curadores, Gestionar asignaciones de filósofos a auxiliares y curadores, Autenticar usuario.
Descripción	Usuario que tiene la capacidad de gestionar toda la información dentro del sistema, tiene todos los permisos de acceso y configuración del entorno virtual.
Tipo	Primario

Tabla 15. Descripción de Actores – Curador.

Actor	Curador
Casos de uso	Buscar información, Gestionar evidencia, Gestionar información de usuario, Gestionar periodo, Gestionar aporte, Gestionar filósofo, Gestionar acceso a auxiliares, Gestionar asignaciones de filósofos a auxiliares, Autenticar usuario.
Descripción	Usuario que está a cargo de gestionar toda la información correspondiente de un filósofo que haya creado o que le haya sido asignado por un administrador, también podrá gestionar permisos de acceso y de configuración a los auxiliares que tenga a cargo.
Tipo	Primario

Tabla 16. Descripción de Actores – Auxiliar.

Actor	Auxiliar
Casos de uso	Buscar información, Gestionar evidencia, Gestionar aporte, Gestionar información de usuario, Gestionar periodo, Gestionar filósofo, Autenticar usuario.
Descripción	Usuario que está a cargo de gestionar la información correspondiente de un filósofo que haya creado o que se le haya sido asignado por parte de un administrador o un curador.
Tipo	Primario

7.1.11. Diagrama de casos de uso. Los principales casos de uso identificados del sistema son los siguientes:

- Enviar solicitud de registro
- Iniciar sesión
- Contactar administrador
- Recuperar contraseña
- Buscar información
- Gestionar evidencia
- Gestionar aporte
- Gestionar periodo
- Gestionar filósofo
- Gestionar información de usuario
- Gestionar acceso a usuarios
- Gestionar asignaciones a usuarios
- Gestionar mensajes de contacto
- Autenticar usuario

En la figura 12 se pueden observar los actores del sistema y sus correspondientes casos de uso, en la figura 13 se muestran dos casos de uso con más detalle.

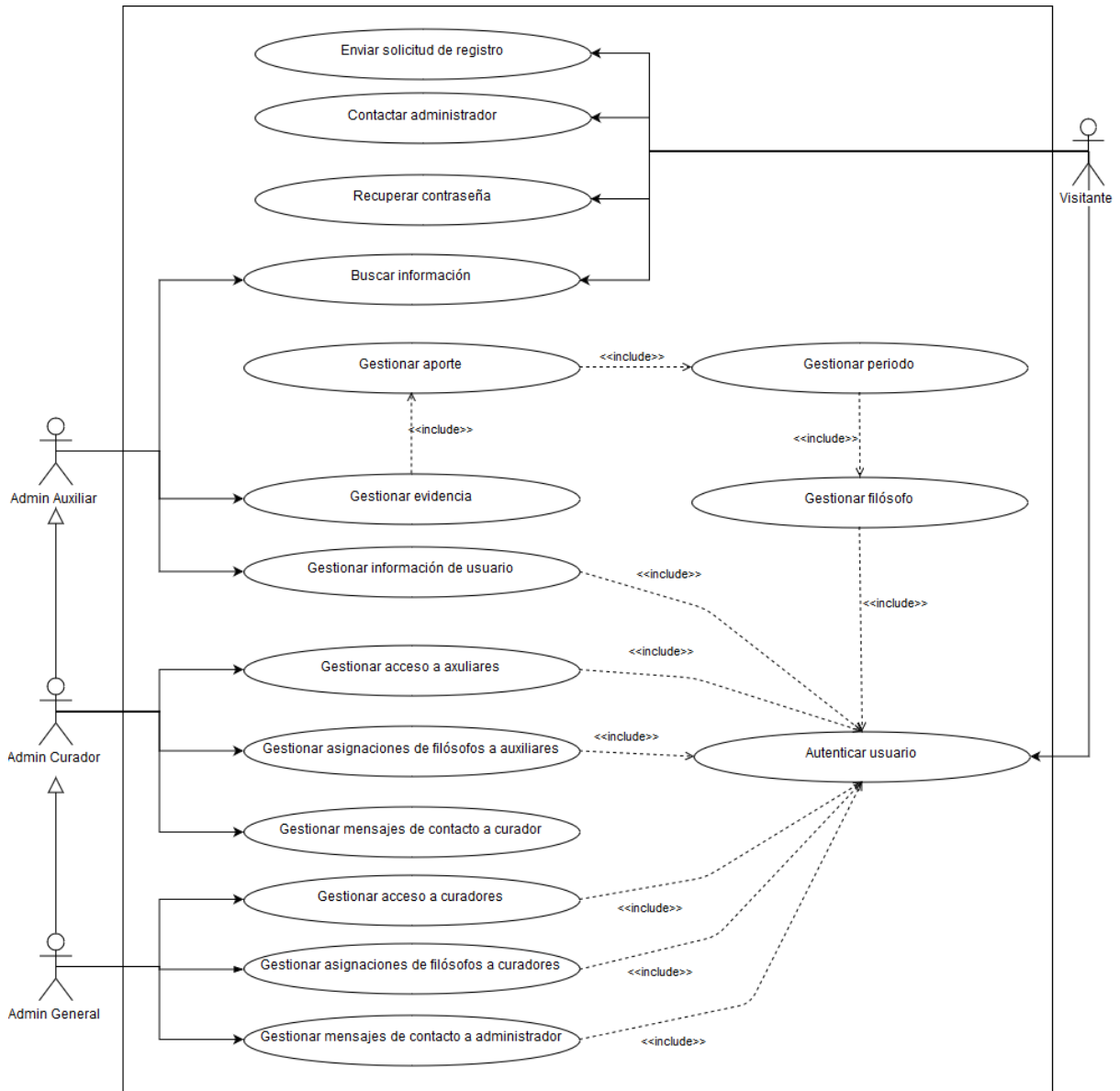


Figura 12. Diagrama general de casos de uso y actores del sistema.

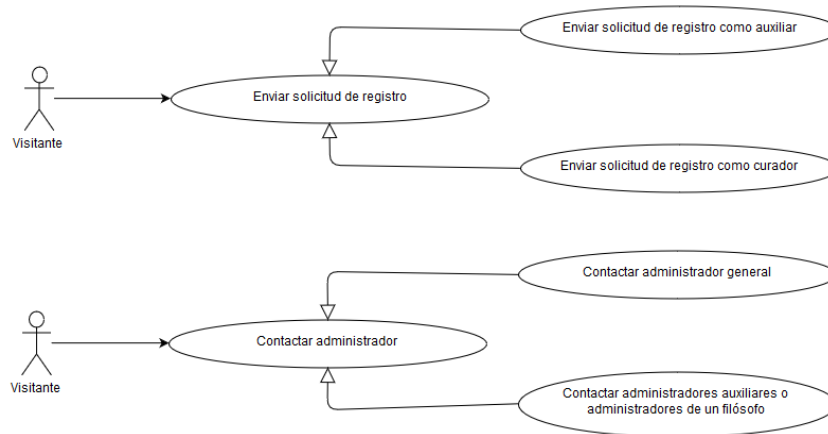


Figura 13. Detalle de los casos de uso Contactar administrador y Enviar solicitud de registro

7.1.12. Descripción de casos de uso.

7.1.12.1. Autenticar usuario

Tabla 17. Descripción de Casos de Uso – Autenticar usuario

Título del caso de uso	Autenticar usuario	Id del caso de uso	CU1
Descripción general del caso de uso o requerimiento			
Este caso de uso es iniciado cuando un usuario desea acceder al panel de administración proporcionado por el sistema.			
Entidades/Actores involucrados			
Administrador, Curador y Auxiliar			
Condiciones de entrada			
El usuario tiene que estar registrado anteriormente, para lo cual, debe haber realizado una solicitud de registro y que un curador o administrador (dependiendo del tipo de solicitud) haya aceptado.			
Flujo normal de Eventos			
	Administrador, Curador o Auxiliar		Sistema
1.	Selecciona la opción 'Iniciar sesión'		Se muestra un formulario con los campos necesarios para autenticar el usuario: correo electrónico y contraseña. Se mostrará un botón para confirmar el inicio de sesión, un enlace para recuperar la contraseña y un botón para iniciar el proceso de registro en caso de que no esté registrado.
Condiciones de salida			
El sistema redirige al usuario al panel de administración			

Caminos de Excepción /Flujos alternativos

- En caso de que los datos ingresados no sean correctos el sistema mostrará un mensaje de error.
- En caso de que el usuario seleccione el botón de registro el sistema redireccionará al usuario a una página para realizar el proceso de registro.

7.1.12.2. Enviar solicitud de registro

Tabla 18. Descripción de Casos de Uso – Enviar solicitud de registro

Título del caso de uso	Enviar solicitud de registro	Id del caso de uso	CU2
Descripción general del caso de uso o requerimiento			
Es iniciado cuando un usuario visitante requiere un rol de usuario dentro del sistema para administrar la información dentro del entorno virtual.			
Entidades/Actores involucrados			
Visitante, Curador, Administrador			
Condiciones de entrada			
El usuario tiene que haber seleccionado la opción de registro en la página principal del entorno virtual.			
Flujo normal de Eventos			
	Sistema	Visitante	Administrador o Curador
1.	Se muestra un formulario con los campos necesarios para el registro de un usuario: correo electrónico, nombres, apellidos, contraseña, confirmar contraseña y el rol que desea. En caso de que seleccione el rol de Auxiliar, el sistema mostrará una lista de Curadores a los cuales enviar la solicitud de registro	Completa los campos mostrados en el formulario y envía la solicitud de registro.	
2.	En caso de que sea una solicitud de registro para curador, se envía una solicitud a todos los administradores y en caso de que sea una solicitud de registro para auxiliar, se envía una solicitud al curador correspondiente y a todos los administradores.		El Administrador o Curador correspondiente resuelve la solicitud de registro (acepta o rechaza la solicitud)
3.	Se notificará por correo electrónico si la solicitud fue rechazada o		

aceptada. En caso de que sea aceptada se le garantizarán permisos de acceso al entorno virtual.

Condiciones de salida

El sistema redirige al usuario al panel de administración

Caminos de Excepción /Flujos alternativos

- En caso de que los datos ingresados no sean correctos el sistema mostrará un mensaje de error.
- En caso de que el usuario seleccione el botón de registro el sistema redireccionará al usuario a una página para realizar el proceso de registro.
- En caso de que el usuario seleccione el botón de recuperar contraseña el sistema redireccionará el usuario a una página que recupere la contraseña introduciendo el correo electrónico.

7.1.12.3. Recuperar contraseña.

Tabla 19. Descripción de Casos de Uso – Recuperar contraseña.

Título del caso de uso	Recuperar contraseña	Id del caso de uso	CU3
Descripción general del caso de uso o requerimiento			
Este caso de uso es iniciado cuando un usuario que ya se ha registrado previamente desea recuperar la contraseña de acceso por motivo de olvido.			
Entidades/Actores involucrados			
Administrador, Curador y Auxiliar			
Condiciones de entrada			
El usuario tiene que estar registrado anteriormente, para lo cual, debe haber realizado una solicitud de registro y que un curador o administrador (dependiendo del tipo de solicitud) haya aceptado.			
Flujo normal de Eventos			
	Administrador, Curador o Auxiliar		Sistema
1.	Selecciona la opción de recuperar contraseña después de haber seleccionado la opción de ‘Iniciar sesión’ en la página principal del entorno virtual.		Se muestra un formulario con el campo de correo electrónico.
2.	Completa los campos necesarios del formulario mostrado y envía los datos al seleccionar el botón ‘recuperar contraseña’		El sistema envía un correo electrónico al email proporcionando las instrucciones que el usuario debe seguir
Condiciones de salida			

Después de enviar el formulario se le notificará al usuario que ingrese al correo para que pueda ver las indicaciones para recuperar la contraseña.

Caminos de Excepción /Flujos alternativos

- En caso de que los datos ingresados no sean correctos o el correo electrónico no exista el sistema mostrará un mensaje de error.

7.1.12.4. Contactar administrador general.

Tabla 20. Descripción de Casos de Uso – Contactar administrador general

Título del caso de uso	Contactar administrador general	Id del caso de uso	CU4
Descripción general del caso de uso o requerimiento			
Es iniciado cuando un usuario visitante desea contactar a un administrador del entorno virtual para comunicar algún inconveniente o sugerencia.			
Entidades/Actores involucrados			
Visitante			
Condiciones de entrada			
Ninguna, cualquier usuario que entre a la página principal puede comunicarse con los administradores del entorno virtual.			
Flujo normal de Eventos			
	Visitante	Sistema	
1.	Selecciona la opción de contactar administrador en la parte inferior de la página principal	Se muestra un formulario con los campos necesarios para enviar el formulario de contacto.	
2.	El usuario completa los campos mostrados y envía el formulario.	El sistema envía un correo electrónico a los administradores del entorno virtual y guarda el mensaje en el sistema.	
Condiciones de salida			
No debe haber errores en los campos del formulario para poder enviar el mensaje			
Caminos de Excepción /Flujos alternativos			
<ul style="list-style-type: none"> - Los administradores podrán gestionar los mensajes recibidos en el panel de administración. - En caso de que los datos ingresados no sean correctos el sistema mostrará un mensaje de error. 			

7.1.12.5. Gestionar mensajes de contacto a administrador / curador / auxiliar

Tabla 21. Descripción de Casos de Uso – Gestionar mensajes de contacto a administrador/curador/auxiliar

Título del caso de uso	Gestionar mensajes de contacto a administrador/curador/auxiliar	Id del caso de uso	CU5
------------------------	---	--------------------	-----

Descripción general del caso de uso o requerimiento		
Un administrador/curador/curador puede cambiar un estado de un mensaje enviado por parte de un usuario visitante y notificar a los demás administradores/auxiliares si ya se leyó y se resolvió el posible inconveniente o sugerencia.		
Entidades/Actores involucrados		
Administrador, Curador y Auxiliar		
Condiciones de entrada		
El usuario debe haberse autenticado como administrador/curador/auxiliar y debe haber mensajes enviados.		
Flujo normal de Eventos		
	Visitante	Sistema
1.	Selecciona la opción de mensajes en el panel de administración	Se muestra una lista con los mensajes recibidos Se muestra el mensaje completo, sus atributos y las opciones de: Ver mensaje y enviar mensaje. También cambia el estado del mensaje a “Visto” por el administrador correspondiente.
2.	Selecciona uno de los mensajes	El sistema muestra un formulario para enviar un correo electrónico
3.	Selecciona Enviar mensaje	El sistema envía un correo electrónico al email proporcionado y cambia el estado del mensaje a “Respondido” por el administrador correspondiente
4.	Redacta un correo electrónico y confirma su envío.	
Condiciones de salida		
Ninguna		
Camino de Excepción /Flujos alternativos		
<ul style="list-style-type: none"> - El estado del mensaje cambia ‘No leído’ a ‘Leído’ cuando el usuario entra a ver el mensaje - El estado del mensaje solo puede cambiar de ‘Sin resolver’ a ‘Resuelto’ - Quedará registrado quién y cuándo se cambió el estado del mensaje. 		

7.1.12.6. Contactar curador

Tabla 22. Descripción de Casos de Uso – Contactar curador.

Título del caso de uso	Contactar curador	Id del caso de uso	CU6
Descripción general del caso de uso o requerimiento			
Es iniciado cuando un usuario visitante desea contactar a un curador de un filósofo para comunicar algún inconveniente o sugerencia.			

Entidades/Actores involucrados

Visitante

Condiciones de entrada

- El usuario debe ingresar al perfil del filósofo en cuestión.
- Cualquier usuario que entre a un perfil de un filósofo puede comunicarse con el administrador curador y sus auxiliares.

Flujo normal de Eventos

	Visitante	Sistema
1.	Selecciona la opción de contactar curador ubicado en la parte superior del perfil del filósofo.	Se muestra un formulario con los campos necesarios para enviar el formulario de contacto.
2.	El usuario completa los campos mostrados y envía el formulario.	El sistema envía un correo electrónico a los curadores y auxiliares encargados de gestionar la información del filósofo y guarda el mensaje en el sistema.

Condiciones de salida

No debe haber errores en los campos del formulario para poder enviar el mensaje

Caminos de Excepción /Flujos alternativos

- Los curadores y auxiliares podrán gestionar los mensajes recibidos en el panel de administración.
- En caso de que los datos ingresados no sean correctos el sistema mostrará un mensaje de error.

7.1.12.7. Buscar información

Tabla 23. Descripción de Casos de Uso – Buscar información

Título del caso de uso	Buscar información	Id del caso de uso	CU7
Descripción general del caso de uso o requerimiento			
Es iniciad cuando un usuario selecciona el botón de búsqueda para realizar una consulta sobre filósofos, aportes o evidencias existentes en la plataforma.			
Entidades/Actores involucrados			
Visitante, Auxiliar, Curador y Administrador			
Condiciones de entrada			
<ul style="list-style-type: none"> - El usuario debe ingresar a la página principal del entorno virtual. - El usuario debe ingresar al perfil del filósofo donde se encontrará el botón para desplegar el módulo de búsqueda. 			
Flujo normal de Eventos			
	Visitante		Sistema

1.	Selecciona la opción de realizar búsqueda en la parte superior de la página en la que se encuentre.	Se muestra un formulario con los campos y filtros necesarios para enviar el formulario de contacto.
2.	El usuario completa los campos mostrados y realiza la consulta	El sistema muestra una lista con los resultados
Condiciones de salida		
No debe haber errores en los campos del formulario para poder realizar la consulta		
Caminos de Excepción /Flujos alternativos		
<ul style="list-style-type: none"> - Solo se permiten caracteres alfanuméricos. - El usuario podrá ver la evidencia o ingresar al perfil de un filósofo o de un aporte mostrado en los resultados. 		

7.1.12.8. Gestionar información de usuario

Tabla 24. Descripción de Casos de Uso – Gestionar información de usuario

Título del caso de uso	Gestionar información de usuario	Id del caso de uso	CU8
Descripción general del caso de uso o requerimiento			
Cada usuario registrado puede entrar al panel de administración y gestionar su información de usuario.			
Entidades/Actores involucrados			
Auxiliar, Curador y Administrador			
Condiciones de entrada			
<ul style="list-style-type: none"> - El usuario debe haberse autenticado como administrador/curador/auxiliar e ingresar al panel de administración 			
Flujo normal de Eventos			
	Administrador, Curador o Auxiliar	Sistema	
1.	Selecciona la opción de editar información de usuario mostrada en el menú de opciones.	Se muestra un formulario con los campos y filtros necesarios para gestionar la información de usuario.	
2.	El usuario realiza los cambios necesarios en los campos del formulario y selecciona la opción de guardar.	El sistema actualiza la información en la base de datos y notifica al usuario.	
Condiciones de salida			
No debe haber errores en los campos del formulario para poder realizar modificaciones			
Caminos de Excepción /Flujos alternativos			
<ul style="list-style-type: none"> - En caso de que los datos ingresados no sean correctos el sistema mostrará un mensaje de error. 			

7.1.12.9. Gestionar filósofo

Tabla 25. Descripción de Casos de Uso – Gestionar filósofo

Título del caso de uso	Gestionar filósofo	Id del caso de uso	CU9
Descripción general del caso de uso o requerimiento			
Cada usuario registrado puede entrar al panel de administración y gestionar la información relacionada a un filósofo.			
Entidades/Actores involucrados			
Auxiliar, Curador y Administrador			
Condiciones de entrada			
<ul style="list-style-type: none"> - El usuario debe haberse autenticado como administrador/curador/auxiliar e ingresar al panel de administración 			
Flujo normal de Eventos			
	Administrador, Curador o Auxiliar	Sistema	
1.	Selecciona la opción de perfiles administrados mostrada en el menú de opciones.	Se muestra una lista de los perfiles de filósofo administrados y diferentes opciones para gestionar la información correspondiente.	
2.	<ul style="list-style-type: none"> - Selecciona eliminar filósofo - Selecciona ver filósofo - Selecciona editar filósofo - Selecciona crear filósofo 	<ul style="list-style-type: none"> - El sistema pide confirmación en caso de que se haya seleccionado eliminar filósofo - Redirige al usuario al perfil del filósofo - Muestra un formulario para editar la información de un filósofo - Muestra un formulario para crear un filósofo 	
3.	<ul style="list-style-type: none"> - Confirma la eliminación del filósofo - Actualiza los campos del formulario para editar filósofo y confirma la acción - Llena los campos del formulario para crear un filósofo 	<ul style="list-style-type: none"> - El sistema elimina toda la información correspondiente al filósofo - El sistema actualiza los datos correspondientes - El sistema crea el nuevo perfil de filósofo 	
Condiciones de salida			
Ninguna			
Caminos de Excepción /Flujos alternativos			
<ul style="list-style-type: none"> - En caso de que los datos ingresados en los formularios correspondientes no sean correctos el sistema mostrará un mensaje de error. 			

7.1.12.10. Gestionar periodo

Tabla 26. Descripción de Casos de Uso – Gestionar periodo

Título del caso de uso	Gestionar periodo	Id del caso de uso	CU10
Descripción general del caso de uso o requerimiento			
Cada usuario registrado puede entrar al panel de administración y gestionar los periodos relevantes en la vida de los filósofos sobre los cuales tengan permiso de gestión.			
Entidades/Actores involucrados			
Auxiliar, Curador y Administrador			
Condiciones de entrada			
<ul style="list-style-type: none"> - El usuario debe haberse autenticado como administrador/curador/auxiliar e ingresar al panel de administración - Se debe haber creado previamente al menos un perfil de filósofo 			
Flujo normal de Eventos			
	Administrador, Curador o Auxiliar	Sistema	
1.	Selecciona la opción de perfiles administrados mostrada en el menú de opciones.	Se muestra una lista de los perfiles de filósofo administrados y diferentes opciones para gestionar la información correspondiente.	
2.	Selecciona uno de los perfiles mostrados	El sistema muestra una lista de periodos <ul style="list-style-type: none"> - El sistema pide confirmación en caso de que se haya seleccionado eliminar periodo 	
3.	<ul style="list-style-type: none"> - Selecciona eliminar periodo - Selecciona editar periodo - Selecciona crear periodo 	<ul style="list-style-type: none"> - Muestra un formulario para editar la información de un periodo - Muestra un formulario para crear un periodo 	
4.	<ul style="list-style-type: none"> - Confirma la eliminación del periodo - Actualiza los campos del formulario para editar periodo y confirma la acción - Llena los campos del formulario para crear un periodo 	<ul style="list-style-type: none"> - El sistema elimina toda la información correspondiente al periodo - El sistema actualiza los datos correspondientes - El sistema crea el nuevo periodo 	
Condiciones de salida			
Ninguna			
Caminos de Excepción /Flujos alternativos			
<ul style="list-style-type: none"> - En caso de que los datos ingresados en los formularios correspondientes no sean correctos el sistema mostrará un mensaje de error. 			

7.1.12.11. Gestionar aporte

Tabla 27. Descripción de Casos de Uso – Gestionar aporte

Título del caso de uso	Gestionar aporte	Id del caso de uso	CU11
Descripción general del caso de uso o requerimiento			
Cada usuario registrado puede entrar al panel de administración y gestionar los aportes en los periodos relevantes de la vida de los filósofos sobre los cuales tengan permiso de gestión.			
Entidades/Actores involucrados			
Auxiliar, Curador y Administrador			
Condiciones de entrada			
<ul style="list-style-type: none"> - El usuario debe haberse autenticado como administrador/curador/auxiliar e ingresar al panel de administración - Se deben haber creado previamente uno o varios perfiles de filósofos - Se deben haber creado previamente uno o varios periodos de tiempo 			
Flujo normal de Eventos			
	Administrador, Curador o Auxiliar	Sistema	
1.	Selecciona la opción de perfiles administrados mostrada en el menú de opciones.	Se muestra una lista de los perfiles de filósofo administrados y diferentes opciones para gestionar la información correspondiente.	
2.	Selecciona uno de los perfiles mostrados	El sistema muestra una lista de periodos	
3.	Selecciona uno de los periodos mostrados	El sistema muestra una lista de aportes <ul style="list-style-type: none"> - El sistema pide confirmación en caso de que se haya seleccionado eliminar aporte 	
4.	<ul style="list-style-type: none"> - Selecciona eliminar aporte - Selecciona editar aporte - Selecciona crear aporte 	<ul style="list-style-type: none"> - Muestra un formulario para editar la información del aporte - Muestra un formulario para crear un aporte 	
5.	<ul style="list-style-type: none"> - Confirma la eliminación del aporte - Actualiza los campos del formulario para editar periodo y confirma la acción - Llena los campos del formulario para crear un aporte 	<ul style="list-style-type: none"> - El sistema elimina toda la información correspondiente a un aporte - El sistema actualiza los datos correspondientes - El sistema crea el nuevo aporte 	
Condiciones de salida			
Ninguna			
Caminos de Excepción /Flujos alternativos			
<ul style="list-style-type: none"> - En caso de que los datos ingresados en los formularios correspondientes no sean correctos el sistema mostrará un mensaje de error. 			

7.1.12.12. Gestionar evidencia

Tabla 28. Descripción de Casos de Uso – Gestionar evidencia

Título del caso de uso	Gestionar evidencia	Id del caso de uso	CU12
Descripción general del caso de uso o requerimiento			
Cada usuario registrado puede entrar al panel de administración y gestionar las evidencias de los aportes de un filósofo sobre el cual tenga permiso de gestión.			
Entidades/Actores involucrados			
Auxiliar, Curador y Administrador			
Condiciones de entrada			
<ul style="list-style-type: none"> - El usuario debe haberse autenticado como administrador/curador/auxiliar e ingresar al panel de administración - Se deben haber creado previamente uno o varios perfiles de filósofos - Se deben haber creado previamente uno o varios periodos de tiempo - Se deben haber creado previamente uno o varios aportes 			
Flujo normal de Eventos			
	Administrador, Curador o Auxiliar		Sistema
1.	Selecciona la opción de perfiles administrados mostrada en el menú de opciones.		Se muestra una lista de los perfiles de filósofo administrados y diferentes opciones para gestionar la información correspondiente.
2.	Selecciona uno de los perfiles mostrados		El sistema muestra una lista de periodos
3.	Selecciona uno de los periodos mostrados		El sistema muestra una lista de aportes
4.	Selecciona uno de los aportes mostrados		El sistema muestra una lista de las evidencias <ul style="list-style-type: none"> - El sistema pide confirmación en caso de que se haya seleccionado eliminar aporte
5.	<ul style="list-style-type: none"> - Selecciona eliminar evidencia - Selecciona editar evidencia - Selecciona crear evidencia 		<ul style="list-style-type: none"> - Muestra un formulario para editar la información de la evidencia - Muestra un formulario para crear una evidencia
6.	<ul style="list-style-type: none"> - Confirma la eliminación de la evidencia - Actualiza los campos del formulario para editar evidencia y confirma la acción - Llena los campos del formulario para crear una evidencia 		<ul style="list-style-type: none"> - El sistema elimina toda la información correspondiente a una evidencia - El sistema actualiza los datos correspondientes - El sistema crea la nueva evidencia
Condiciones de salida			

Ninguna

Caminos de Excepción /Flujos alternativos

- En caso de que los datos ingresados en los formularios correspondientes no sean correctos el sistema mostrará un mensaje de error.

7.1.12.13. Gestionar acceso a auxiliares/curadores

Tabla 29. Descripción de Casos de Uso – Gestionar acceso a auxiliares/curadores

Título del caso de uso	Gestionar acceso a auxiliares/curadores	Id del caso de uso	CU13
Descripción general del caso de uso o requerimiento			
Un usuario administrador o curador puede gestionar el acceso al entorno virtual de un usuario curador o auxiliar			
Entidades/Actores involucrados			
Curador y Administrador			
Condiciones de entrada			
<ul style="list-style-type: none"> - El usuario debe haberse autenticado como administrador/curador/auxiliar e ingresar al panel de administración 			
Flujo normal de Eventos			
	Administrador y Curador	Sistema	
1.	Selecciona la opción de gestionar acceso mostrada en el menú de opciones.	Se muestra una lista de las solicitudes de registro para el rol de curadores y auxiliares, con las opciones de rechazar y aceptar solicitud. También una lista de los curadores y auxiliares con permiso de acceso garantizado, con la opción para revocar o garantizar el acceso.	
2.	<ul style="list-style-type: none"> - Selecciona una de las opciones para resolver la solicitud de acceso y confirma la acción - Selecciona la opción para revocar o garantizar el acceso a un usuario 	<ul style="list-style-type: none"> - El sistema acepta o deniega el acceso al usuario solicitante realizando los cambios necesarios en la base de datos. - El sistema revoca o garantiza el acceso al usuario correspondiente realizando los cambios necesarios en la base de datos 	
Condiciones de salida			
Ninguna			
Caminos de Excepción /Flujos alternativos			
<ul style="list-style-type: none"> - Un curador solo puede resolver solicitudes de registro para el rol de auxiliar - Un curador solo puede revocar o garantizar el acceso a usuarios auxiliares 			

7.1.12.14. Gestionar asignaciones/permisos de filósofos a auxiliares/curadores

Tabla 30. Descripción de Casos de Uso – Gestionar asignaciones de filósofos a auxiliares/curadores

Título del caso de uso	Gestionar asignaciones de filósofos a auxiliares/curadores	Id del caso de uso	CU14
Descripción general del caso de uso o requerimiento			
Un usuario administrador o curador puede gestionar el permiso de gestión a un usuario auxiliar o curador dependiendo del rol que posea			
Entidades/Actores involucrados			
Curador y Administrador			
Condiciones de entrada			
<ul style="list-style-type: none"> - El usuario debe haberse autenticado como administrador/curador/auxiliar e ingresar al panel de administración 			
Flujo normal de Eventos			
	Administrador y Curador	Sistema	
1.	Selecciona la opción de gestionar acceso mostrada en el menú de opciones.	Se muestra una lista de los curadores y auxiliares con los filósofos sobre los cuales tienen permiso de gestión y la opción de agregar un filósofo más o revocar el permiso de gestión sobre un filósofo.	
2.	Selecciona una de las opciones para agregar o revocar permisos de gestión sobre un filósofo de un curador o auxiliar	El sistema agrega o revoca el permiso de gestión sobre un filósofo del usuario seleccionado realizando los cambios necesarios en la base de datos	
Condiciones de salida			
Ninguna			
Caminos de Excepción /Flujos alternativos			
<ul style="list-style-type: none"> - Un curador solo puede gestionar los permisos de gestión de sus auxiliares. - Un administrador puede gestionar todos los permisos de gestión de los curadores y los auxiliares del sistema. 			

7.1.13. Diagramas de flujo. Se realizaron los diagramas de flujo de los diferentes casos de uso, estos se pueden encontrar en el apéndice A.

Figura 14. Diagrama entidad-relación.

7.2.1. Diccionario de datos. En la siguiente tabla se puede encontrar una descripción de cada uno de los atributos de las entidades filósofo mostrada en el diagrama entidad relación. La descripción de los atributos de las demás entidades se puede encontrar en el apéndice B.

Tabla 31. Descripción de los atributos de filósofo.

Filósofo		
Atributo	Tipo de dato	Descripción
Id	INT	Clave única para el registro de cada filósofo
Nombre	TEXT	Nombres del filósofo
Sexo	TEXT	Se define si el filósofo es hombre o mujer
Tema	TEXT	Define el tema del perfil del filósofo
Fecha_nacimiento	TEXT	Fecha de nacimiento del filósofo
Fecha_muerte	TEXT	Fecha de fallecimiento del filósofo en caso de que corresponda
Ciudad_nacimiento	TEXT	Ciudad de nacimiento del filósofo
Pais_nacimiento	TEXT	País de nacimiento del filósofo
Nacionalidad	TEXT	Nacionalidad del filósofo
Biografía	TEXT	El texto correspondiente a la biografía del filósofo
Image_url	VARCHAR (255)	Ruta en el servidor de la imagen de perfil

7.3. Arquitectura Cliente-Servidor de tres capas

La arquitectura cliente-servidor propone la división de los procesos necesarios para el funcionamiento de un software en dos o más capas que interactúan entre sí. Generalmente la capa de usuario es la encargada de realizar una o más peticiones a la capa de servidor, la cual procesa la solicitud, consulta con la base de datos y devuelve la información necesaria. Esta separación de funciones le proporciona al software mayor seguridad, reparto de la carga de procesamiento de

datos ya que los componentes desarrollados se pueden repartir en varios servidores, facilidad de mantenimiento, diversidad de lenguajes de programación para su implementación y accesos más eficientes a los datos.



Figura 15. Capas de la arquitectura Cliente-Servidor. Adaptado de (Marini, 2012)

Las capas propuestas se pueden observar en la figura 15 son:

- **Lógica de presentación o capa de usuario:** Esta capa es la encargada de interactuar con el usuario generalmente por medio de una interfaz gráfica registrando las peticiones realizadas y mostrando los resultados obtenidos.
- **Lógica de negocio o capa de servidores de aplicación:** Es la encargada de asegurar la integridad de las transacciones de las operaciones necesarias para el funcionamiento del software. Recibe las solicitudes de la capa de presentación y consulta a la capa de lógica de datos para devolver la información necesaria. También separa las reglas del negocio y su lógica de las interfaces de usuario por lo que pueden existir distintas versiones de la interfaz interactuando con esta capa.
- **Lógica de datos o capa de servidores de bases de datos:** Esta capa se encarga de garantizar la integridad referencial de los datos almacenados y la gestión de las

transacciones. Usualmente las funciones de esta capa son realizadas por un SGBD (sistema de gestión de base de datos) como MySQL, Oracle, etc.

Esta arquitectura se puede ver implementada en la mayoría de las aplicaciones web que se pueden encontrar en internet, las cuales aprovechan las funcionalidades, patrones de diseño y estructuras propuestas por distintos frameworks para cumplir las tareas de las tres capas nombradas anteriormente. Las tareas de la capa de presentación o también llamada Frontend generalmente son realizadas por frameworks escritos en JavaScript como ReactJs, Vue.js, Angular, etc. La lógica del negocio es implementada en lenguajes como Laravel, Django, Noves, etc. Y la lógica de datos es implementada por los SGBD, como ya se había mencionado anteriormente (Marini, 2012).

La implementación de la arquitectura planteada para este proyecto se puede ver en la figura 16 en la cual el framework Vue.js cumple las funciones de la capa de usuario (Frontend) y Laravel cumple las funciones de la capa de servidores de aplicación y de base de datos (Backend). En la figura se muestra a MySQL como responsable de las funciones de la capa de servidores de base de datos, pero todavía dentro del framework Laravel, esto se debe a que en ningún momento del desarrollo se interactúa directamente con MySQL ya que Laravel posee un ORM (Object-Relational mapping) o mapeo objeto-relacional que interactúa con MySQL y se encarga de la gestión de las tablas de la base de datos.

La comunicación entre el Frontend y el Backend se realiza por medio de métodos HTTP y de respuestas en formato JSON. La autenticación de usuario se hace por medio de tokens con el paquete Passport proporcionado por Laravel, de esa manera no se mantiene un estado de sesión entre solicitudes.

Se utilizó el servidor Nginx como servidor principal escuchando el puerto 80 y sirve como reverse proxy para el servidor Apache que escucha el puerto 8080 en el cual funciona el Backend con el framework Laravel.

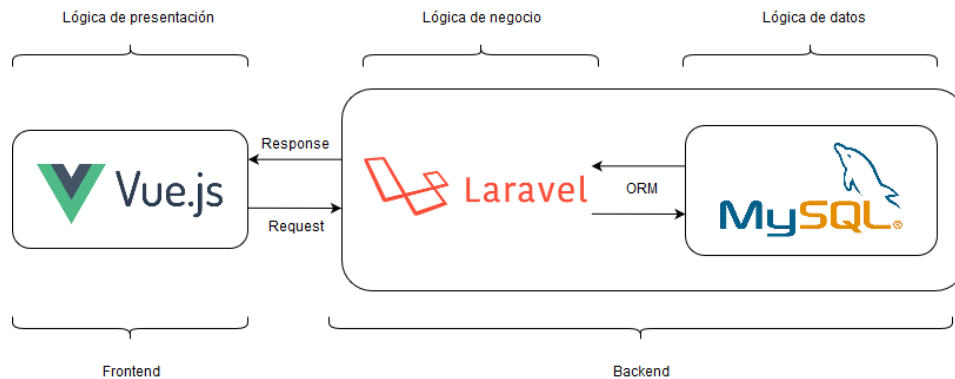


Figura 16. Implementación de la arquitectura.

7.3.1. Frontend. De acuerdo con la arquitectura y el diseño planteado se procedió a realizar la implementación correspondiente a la lógica de presentación mediante el framework Vue.js. Como se puede observar en la siguiente figura, el código fuente de esta capa corresponde al directorio delfic-frontend y contiene siete directorios principales, cada uno se explica a continuación.

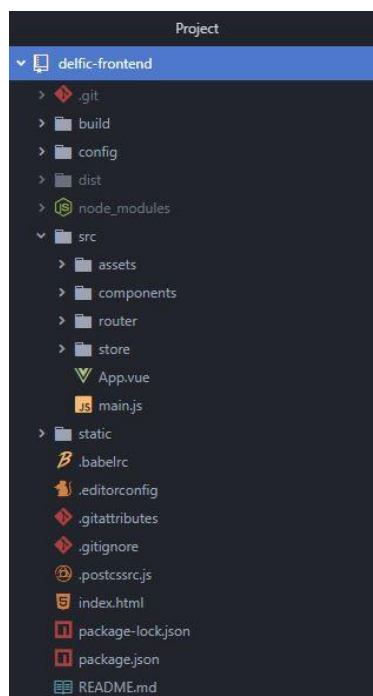


Figura 17. Estructura del código fuente del Frontend.

- `config`: Aquí se encuentra toda la configuración básica del framework.
- `dist`: Este directorio contiene los archivos que servirán como ejecutables en el entorno de producción.
- `src`: Contiene los subdirectorios con los que se define el funcionamiento y la estructura de la plataforma, también contiene el archivo `/App.vue` que corresponde al componente principal de la plataforma dentro del cual se encuentran todos los demás componentes.
- `assets`: Este subdirectorio ubicado dentro del directorio `/src` contiene los recursos estáticos que necesita la plataforma para cargar su estado inicial. Contiene principalmente imágenes que corresponden a los íconos, temas, imágenes de ejemplo y cabeceras de algunas secciones.
- `components`: Ubicado dentro del directorio `/src`, contiene los componentes que definen la estructura de la plataforma.
- `router`: En él se encuentra el archivo `/index.js` que maneja todas las secciones de la plataforma por medio de las URL's.
- `store`: En este se encuentra el archivo `/store.js` que se encarga de almacenar la información dinámica de la plataforma y realizar consultas a la API del Backend para traer nuevos datos.

Los componentes que se encuentran dentro del directorio `/components` están organizados de la siguiente manera:

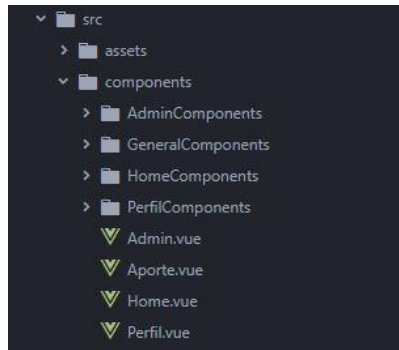


Figura 18. Estructura del directorio /components

Se puede observar que existen cuatro componentes principales, cada uno de ellos corresponde a una de las cuatro páginas principales de la plataforma: Página principal, Perfil de filósofo, Aporte y Panel de administración. Para cada uno de los componentes principales se creó un directorio que contiene todos los subcomponentes necesarios para su funcionamiento, a excepción del componente Aporte.vue ya que los subcomponentes que este requiere son los mismos que los del componente Perfil.vue. En las siguientes figuras se pueden observar todos los componentes de los directorios mostrados en la figura anterior: /AdminComponents, /GeneralComponents, /HomeComponents y /PerfilComponents.

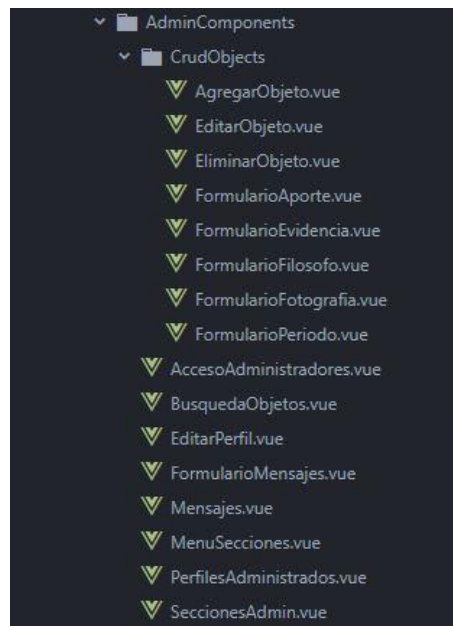
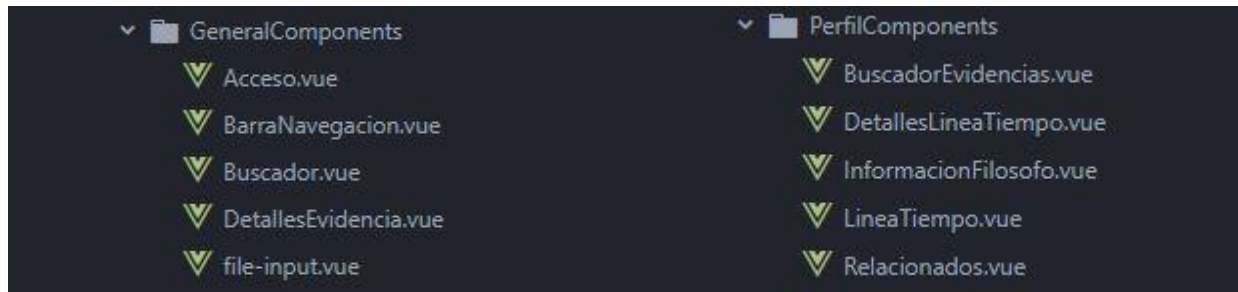
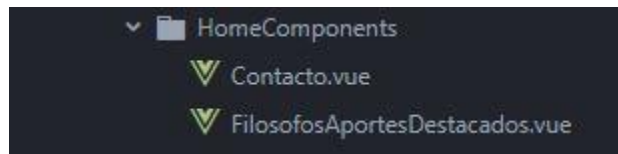


Figura 19. Componentes /AdminComponents.*Figura 20.* Componentes /GeneralComponents y /PerfilComponents.*Figura 21.* Componentes /HomeComponents.

Los componentes son los elementos más importantes que ofrece Vue para el desarrollo de las aplicaciones, estos elementos HTML permiten encapsular código reutilizable y se componen principalmente de tres partes o tres etiquetas:

- **Template:** Esta etiqueta contiene la plantilla que define la estructura del componente, está escrita en HTML y cada uno de los elementos permite agregar atributos específicos del framework web que añaden funcionalidades y operaciones lógicas como if, else, for, etc.
- **Style:** Esta se define el estilo de los elementos agregados en la etiqueta template, está escrita en lenguaje CSS.
- **Script:** En esta etiqueta se definen mediante el lenguaje JavaScript los atributos principales del componente que lo hacen dinámico. Se destacan dos atributos principales: data, en los cuales se definen los datos que van a ser utilizados de manera

reactiva por la plantilla, y methods, en el cual se definen los métodos propios del componente.

En la siguiente figura se puede observar la estructura básica de un componente de Vue.



```
componente.vue
<template>
  <div class="contenedor">
    <p>Texto</p>
  </div>
</template>

<script>
  export default {
    name: 'Componente',
    components: {},
    props: {},
    computed: {},
    data () { return {}},
    methods: {}
  }
</script>

<style scoped>
</style>
```

Figura 22. Estructura básica de un componente.

7.3.2. Backend. La implementación del Backend que corresponde a la capa de lógica de negocio y de datos se realizó mediante una API REST con el framework Laravel, en lenguaje PHP. Esta se encargará de recibir las peticiones realizadas por el Frontend y entregar los respectivos resultados en formato JSON. A continuación, se explicará la estructura de los archivos del código fuente y después la estructura de los endpoints de la API.

En la siguiente figura se muestran los directorios del código fuente, los directorios más relevantes son siete.

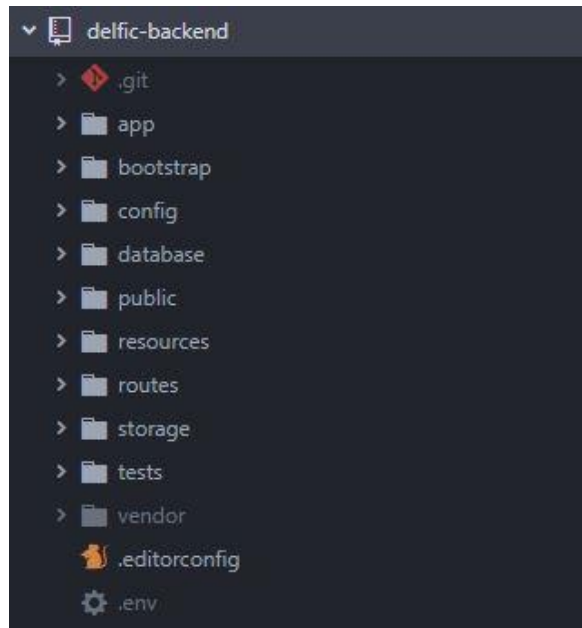


Figura 23. Estructura del Backend de la plataforma.

- **app:** En este directorio se encuentran los modelos de las tablas de la base de datos, cada modelo provee una interfaz ORM (Object-Relational mapping) que es usada por Laravel para interactuar con la base de datos MySQL (se puede observar esta interacción en el diagrama de arquitectura que se encuentra en la sección 7.3). En la siguiente figura se pueden observar otros subdirectorios importantes.

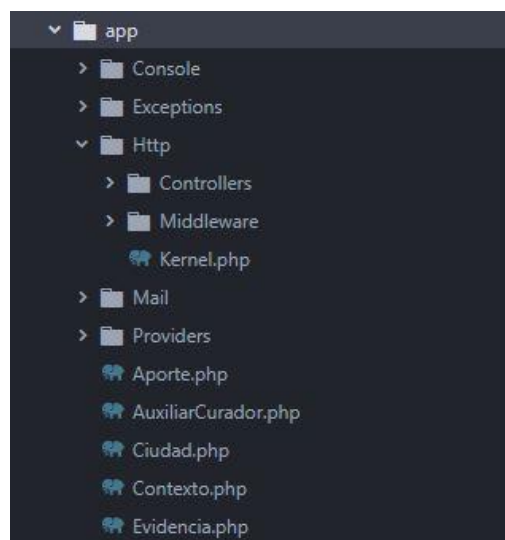


Figura 24. Estructura del directorio /app.

- Http/Controllers: En este directorio se encuentran los archivos que cumplen la funcionalidad de controladores en el patrón MVC usado por Laravel, estos se encargan de recibir las solicitudes realizadas por el Frontend, realizar las respectivas operaciones (inserción, eliminación, edición y consultas) con el ORM en la base de datos de MySQL y finalmente comunicar los resultados obtenidos mediante JSON al Frontend.
- Http/Mail: Se encuentran los archivos que utilizan algunos controladores para enviar información a los usuarios por medio de un correo electrónico. El correo electrónico utilizado es *delficcol@gmail.com* y fue creado únicamente para el uso de la plataforma.
- config: Contiene archivos con las configuraciones básicas del framework Laravel como la conexión con la base de datos, funciones de hashing, autenticación de usuarios, manejo de archivos, caché y el mecanismo de Intercambio de recursos de origen cruzado o también llamado CORS por sus siglas en inglés Cross-origin resource sharing.
- database/migrations: Contiene los archivos que modifican las tablas de la base de datos, tienen el nombre de migraciones y se encargan de representar lo mostrado en los diagramas ER (Entidad-Relación) en forma de tablas, evitando de esa manera interactuar directamente con MySQL.
- routes: Dentro de este directorio se encuentra el archivo *api.php* que toma las peticiones HTTP realizadas por el Frontend y se las comunica a sus respectivos controladores dependiendo del endpoint especificado. En la siguiente figura se puede observar la interacción entre el Frontend, el router, los controladores y el modelo mediante el patrón MVC.

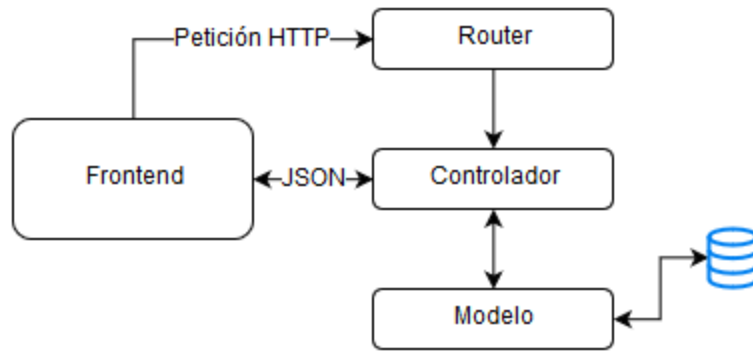


Figura 25. Frontend y API mediante el patrón MVC.

- storage: En este directorio se encuentran los archivos que corresponden a las imágenes y documentos de los objetos principales de la plataforma: filósofos, periodos de tiempo, fotografías, aportes y evidencias.

Los Endpoints son las direcciones URL a las cuales se comunica el Frontend para obtener cierta información, estas están divididas en privadas y públicas y contienen una de las cuatro posibles peticiones HTTP (GET, POST, PATCH, DELETE). En la siguiente tabla se explican con más detalle.

Tabla 32. Endpoints de la API del Backend

Publicas			
RUTA	HTTP	BODY	RESULT
/login	POST	JSON string	Devuelve token de acceso
/register	POST	JSON string	Envía solicitud de registro
/recoverPass	POST	JSON string	Envía mensaje de recuperación
/getCuradores	GET	Vacío	Devuelve una lista de usuarios curadores
/getFilosofo	GET	JSON string	Devuelve la información de un filósofo
/getFilosofosAportesDestacados	GET	Vacío	Devuelve los aportes y filósofos destacados
/busqueda	GET	JSON string	Devuelve un filósofo o aporte que corresponda a la búsqueda realizada
/busquedaObjectAdmin	GET	JSON string	Devuelve un filósofo, periodo, filosofía, aporte o evidencia que corresponda a la búsqueda realizada
/getAporte	GET	JSON string	Devuelve la información de un aporte

/getRelacionadosAporte	GET	JSON string	Devuelve los filósofos o aportes relacionados a un aporte
/getEvidencias	GET	JSON string	Devuelve las evidencias correspondientes a la búsqueda realizada
/postMensaje	POST	JSON string	Crea un nuevo mensaje
Privadas			
RUTA	HTTP	BODY	RESULT
/logout	POST	Vacío	Elimina el token
/user	GET	Vacío	Devuelve la información de usuario
/países	GET	Vacío	Devuelve una lista de países
/ciudades	GET	JSON string	Devuelve las ciudades correspondientes a un país
/searchFilosofo	GET	JSON string	Devuelve la información de un filósofo
/postFilosofo	POST	JSON string	Crea un filósofo
/patchFilosofo	PATCH	JSON string	Edita un filósofo
/deleteFilosofo	DELETE	JSON string	Elimina un filósofo
/postAporte	POST	JSON string	Crea un aporte
/patchAporte	PATCH	JSON string	Edita un aporte
/deleteAporte	DELETE	JSON string	Elimina un aporte
/getDataRelacionadosAporte	GET	JSON string	Devuelve los filósofos o aportes relacionados a un aporte
/postEvidencia	POST	JSON string	Crea una evidencia
/patchEvidencia	PATCH	JSON string	Edita una evidencia
/deleteEvidencia	DELETE	JSON string	Elimina una evidencia
/getDataParticipantes	GET	JSON string	Devuelve los participantes de una evidencia
/postPeriodo	POST	JSON string	Crea un periodo de tiempo
/deletePeriodo	DELETE	JSON string	Elimina un periodo
/patchPeriodo	PATCH	JSON string	Edita un periodo
/postFotografia	POST	JSON string	Crear una fotografía
/patchFotografia	PATCH	JSON string	Edita una fotografía
/deleteFotografia	DELETE	JSON string	Elimina una fotografía
/getAdminFilosofos	GET	JSON string	Devuelve los filósofos correspondientes a un usuario
/getBasicInfoAdmin	GET	JSON string	Devuelve la información básica de un usuario
/updateUser	PATCH	JSON string	Edita un usuario
/changeAccessAdmin	PATCH	JSON string	Cambia el permiso de acceso de un usuario
/deleteAdminFilosofo	DELETE	JSON string	Elimina un permiso de un usuario sobre un filósofo
/addPermisoFilosofo	POST	JSON string	Agrega un permiso de un usuario sobre un filósofo

/getPermisosAdmin	GET	JSON string	Devuelve los permisos sobre filósofos que tiene un usuario
/getSolicitudes	GET	JSON string	Devuelve las solicitudes de acceso correspondientes a un usuario
/acceptSolicitud	PATCH	JSON string	Cambia el estado de una solicitud a aceptada y cambia el permiso de acceso del filósofo correspondiente
/rejectSolicitud	DELETE	JSON string	Cambia el estado de una solicitud a rechazada y elimina al usuario correspondiente
/getMensajesCurador	GET	JSON string	Obtiene los mensajes de contacto a un curador
/getMensajesAdmin	GET	JSON string	Obtiene los mensajes de contacto a administradores
/sendMessage	POST	JSON string	Envía un mensaje por correo electrónico, agrega la respuesta al mensaje correspondiente y cambia el estado del mensaje a resuelto.
/addSeen	POST	JSON string	Cambia el estado del mensaje a leído.

7.4. Vistas de usuario

Se realizaron en total cuatro propuestas para la interfaz del sistema, principalmente de la sección de las líneas de tiempo debido a que esta sección es la que presenta a los usuarios la información más relevante. Estas propuestas fueron presentadas en diferentes reuniones a la directora del proyecto y a los filósofos expertos (mostrados en la sección 7.1) que se encargarán de gestionar la información de los filósofos que se incluirán en la plataforma. Cada propuesta fue evaluada individualmente y con las sugerencias y recomendaciones resultantes se elaboró una siguiente propuesta, siguiendo así la metodología planteada teniendo en cuenta que cada propuesta fue considerada como un prototipo. A continuación, se presenta una captura de pantalla de la sección en la que se muestra la información más relevante de la plataforma, de cada uno de los prototipos realizados.

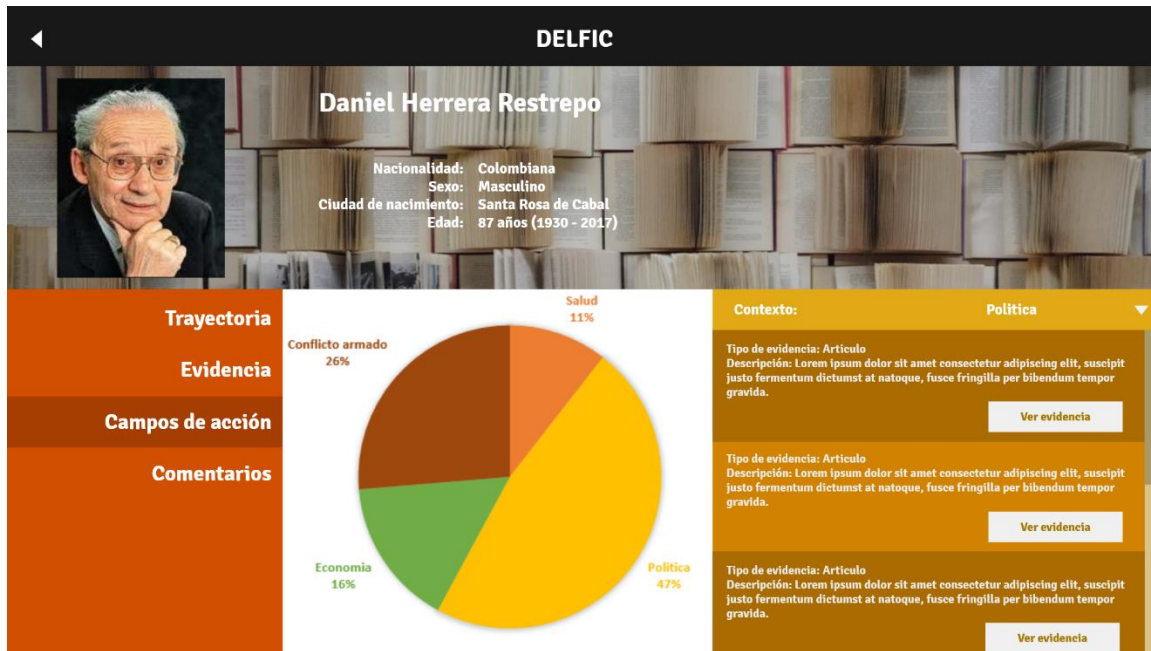


Figura 26. Prototipo 1, sección de campos de acción.

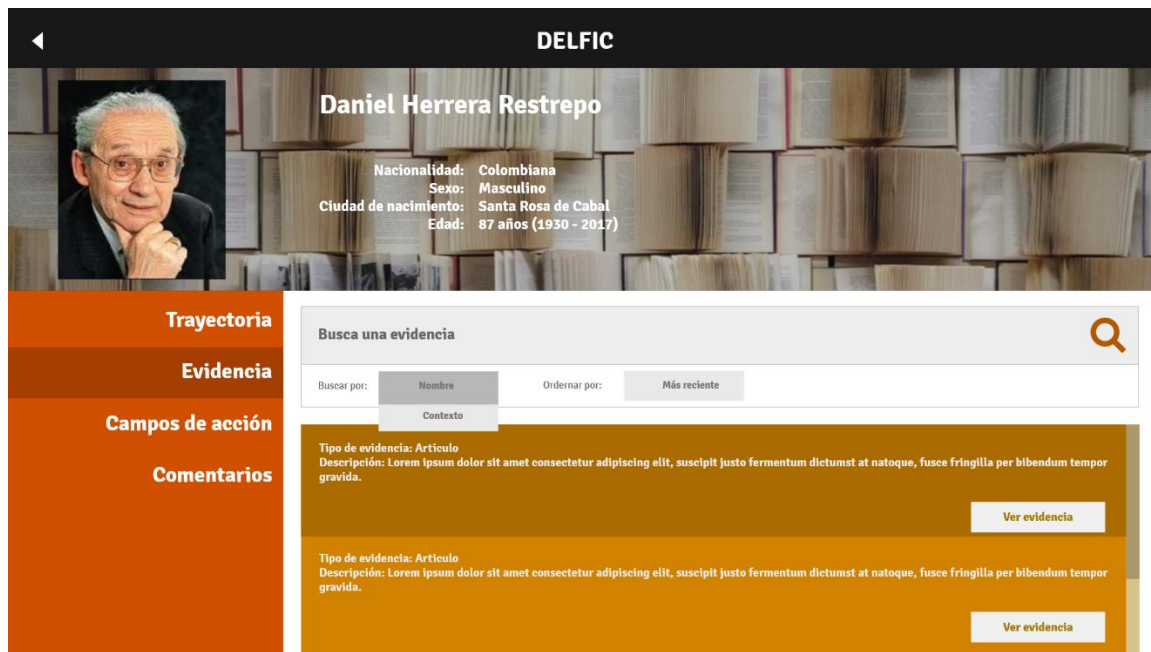


Figura 27. Prototipo 1, sección de evidencias.

En las figuras 17 y 18 se puede apreciar que la información más relevante de la plataforma, que corresponde a las evidencias y aportes de los filósofos se presenta mediante gráficos y listados de información, esto cambiaría en los siguientes prototipos por una o varias líneas de tiempo.

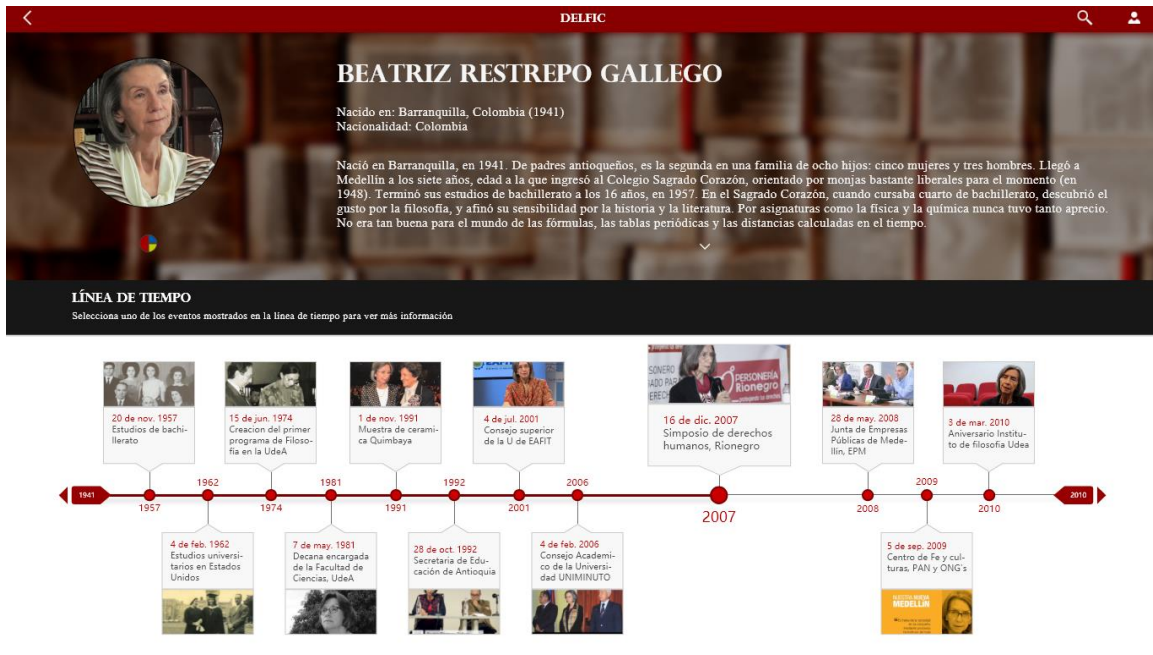


Figura 28. Prototipo 2, sección línea de tiempo.

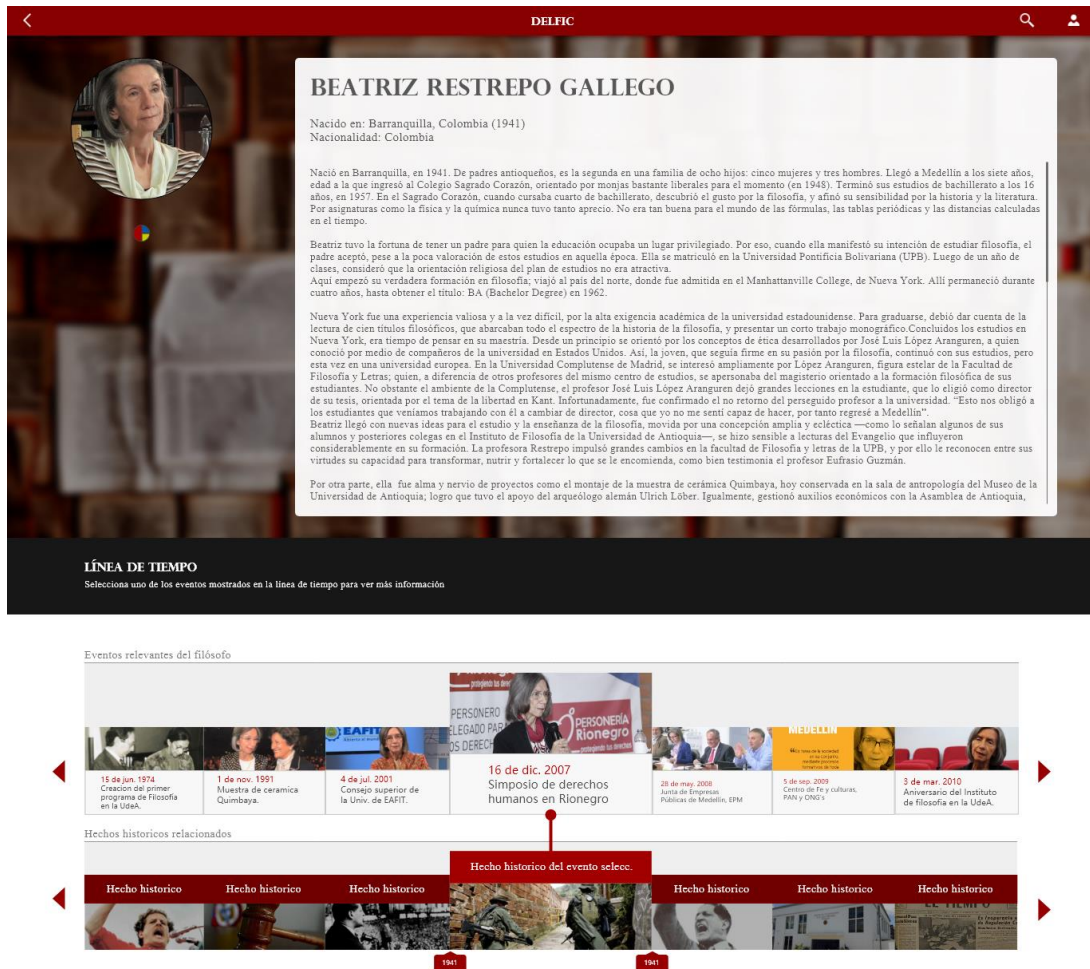


Figura 29. Prototipo 3, sección línea de tiempo.



Figura 30. Prototipo final, sección línea de tiempo.

En el apéndice C se pueden encontrar en más detalle toda la interfaz del prototipo final.

7.5. Pruebas

Con el fin de obtener un prototipo final con la menor cantidad de errores posibles en todas sus funcionalidades e interfaces se realizaron pruebas de diferentes tipos para verificar que distintas características del prototipo funcionaran correctamente. Pressman en su libro Ingeniería del Software Un enfoque práctico (Pressman, 2010), presenta nueve tipos de pruebas para aplicaciones

web o webapps, estas examinan una o más de las siguientes de calidad: contenido, función, estructura, usabilidad, navegabilidad, rendimiento, compatibilidad interoperabilidad y seguridad. El plan de pruebas presentado en este documento contiene cinco de las nueve pruebas propuestas por Pressman, son las siguientes: Prueba de Contenido, Prueba de Compatibilidad, Prueba de Navegación, Prueba de Función y Prueba de seguridad.

7.5.1. Prueba de contenido. El objetivo de esta prueba es encontrar errores sintácticos, semánticos, errores tipográficos menores, información incorrecta que no corresponda con lo planeado para determinada sección de una página o alguna violación de la propiedad intelectual en alguna de las imágenes usadas.

Se identificaron cuatro páginas principales: Home, Perfil, Aporte y Admin, a cada una de ellas se le realizó una revisión de la gramática y de las imágenes para identificar cualquier error, los resultados se muestran a continuación:

Tabla 33. *Resultados de la prueba de contenido.*

Página	Sección	Resultado
Home	Barra de navegación	Correcto
	Cabecera principal	Correcto
	Importancia de los filósofos	Errores de sintaxis, faltan algunas tildes
	Buscar filósofos	Correcto
	Contacto	Correcto
	Login	Correcto
	Registro	Correcto
	Recuperar contraseña	Correcto
	Búsqueda	Correcto
Perfil	Barra de navegación	Correcto
	Información del filósofo	Correcto
	Línea de tiempo	Error semántico en la descripción de la sección
	Detalles de la línea de tiempo	Error de sintaxis
	Aportes y filósofos relacionados	Correcto
	Buscador de evidencias	Correcto
Aporte	Barra de navegación	Correcto
	Información del aporte	Correcto
	Aportes y filósofos relacionados	Correcto
Admin	Barra de navegación	Correcto
	Menú lateral de opciones	Error de sintaxis, falta una tilde

Perfiles administrados	Error semántico en la descripción de los periodos de tiempo, faltan comas que den un sentido correcto a la oración
Mensajes	Error semántico en la descripción de los mensajes de contacto a administradores
Acceso a administradores	Correcto
Editar perfil de usuario	Error de sintaxis, falta una tilde
Búsqueda de objetos	Error de sintaxis, falta una tilde

Según los resultados mostrados en la tabla anterior, se encontraron algunos errores de sintaxis y semánticos los cuales fueron solucionados de manera satisfactoria.

7.5.2. Pruebas de compatibilidad. Debido a que las aplicaciones web son accesibles en internet para todo el mundo, es necesario verificar que todas las interfaces y el rendimiento del sistema funcionen correctamente en cada uno de los navegadores más usados en la actualidad y para distintos tamaños de pantalla.

Los navegadores y tamaños de pantalla seleccionados para la prueba se pueden observar en la siguiente tabla junto a los resultados de la verificación.

Tabla 34. Resultados de la prueba de compatibilidad.

		Tamaños de pantalla		
		Escritorio	Tablet	Dispositivo móvil
Navegadores	Chrome	✓	✓	✓
	Firefox	✓	✓	✓
	Edge	✓	✓	✓

Los resultados de la prueba fueron correctos, aunque se puede realizar una observación respecto al navegador Edge, algunas propiedades CSS no se muestran correctamente como por ejemplo *clip-path: polygon()* pero esto no afecta significativamente la interfaz del sistema.

7.5.3. Prueba de navegación. El Frontend permite al usuario navegar por todas las interfaces proporcionadas, por lo cual es necesario verificar que los redireccionamientos y los vínculos de

navegación funcionan correctamente. Para lo anterior se identificaron ciertas acciones principales que los usuarios comúnmente realizarían dentro del sistema, estas se pueden observar en la siguiente tabla junto a los resultados obtenidos.

Tabla 35. Resultados de la prueba de navegación

	Caso de prueba	Resultado esperado	Resultado obtenido
Home	1. Entrar a la página principal	Redirecciona a la página principal	Redirecciona a la página principal
	2. Ingresar una URL no válida, ejemplo: domain.com/aleatorio	Redirecciona a la página principal	El navegador se queda en blanco
	3. Seleccionar un filósofo destacado	Redirecciona a la página del perfil	Redirecciona a la página del perfil
	4. Seleccionar un aporte destacado	Redirecciona a la página del aporte	Redirecciona a la página del aporte
	5. Entrar al panel de administración desde la barra de navegación después de haberse autenticado	Redirecciona al panel de administración	Redirecciona al panel de administración
Perfil	1. Entrar al perfil usando una URL válida, ejemplo: domain.com/perfil/{filosofo_id}	Redirecciona al perfil del filósofo correspondiente	Redirecciona al perfil del filósofo correspondiente
	2. Entrar al perfil usando una URL no válida, ejemplo: domain.com/perfil/aleatorio	Redirecciona a una página de error, dónde se puede devolver a la página principal	Redirecciona a una página de error
	3. Entrar al panel de administración desde la barra de navegación después de haberse autenticado	Redirecciona al panel de administración	Redirecciona al panel de administración
	4. Seleccionar cerrar sesión desde la barra de navegación	Redirecciona a la página principal	Redirecciona a la página principal
	5. Seleccionar el logo de la barra de navegación	Redirecciona a la página principal	Redirecciona a la página principal
Aporte	1. Entrar al aporte usando una URL válida, ejemplo: domain.com/aporte/{aporte_id}	Redirecciona al perfil del aporte correspondiente	Redirecciona al perfil del aporte correspondiente
	2. Entrar al perfil usando una URL no válida, ejemplo: domain.com/aporte/aleatorio	Redirecciona a una página de error, dónde se puede devolver a la página principal	Redirecciona a una página de error
	3. Entrar al panel de administración desde la barra de navegación después de haberse autenticado	Redirecciona al panel de administración	Redirecciona al panel de administración
	4. Seleccionar cerrar sesión desde la barra de navegación	Redirecciona a la página principal	Redirecciona a la página principal
	5. Seleccionar el logo de la barra de navegación	Redirecciona a la página principal	Redirecciona a la página principal
Admin	1. Ingresa al panel de administración sin haberse autenticado	Redirecciona a la página principal	Redirecciona a la página principal

2.	Ingresar al panel de administración después de haberse autenticado	Redirecciona al panel de administración	Redirecciona al panel de administración
3.	Seleccionar cerrar sesión desde la barra de navegación	Redirecciona a la página principal	Redirecciona a la página principal
4.	Ingresar al panel de administración usando URL no válida, ejemplo: domain.com/admin/aleatorio	Redirecciona a la página principal	El navegador se queda en blanco
5.	Seleccionar el logo de la barra de navegación	Redirecciona a la página principal	Redirecciona a la página principal
6.	Seleccionar ‘ver’ un filósofo	Redirecciona al perfil del filósofo	Redirecciona al perfil del filósofo
7.	Seleccionar ‘ver’ un aporte	Redirecciona al perfil del aporte	Redirecciona al perfil del aporte

De los resultados obtenidos se puede observar que el sistema no redireccionaba a una página de error o la página principal al ingresar una URL que no correspondía en las páginas Home y Admin. Se realizaron las respectivas correcciones en el archivo de configuración de rutas del Frontend ubicado en *delfic-frontend/src/router/index.js*.

7.5.4. Pruebas de función. Las pruebas de función o también llamada de componente, se enfoca en un conjunto de pruebas que verifican el correcto funcionamiento de cada una de las funcionalidades del sistema. La metodología aplicada en esta prueba es la de caja negra, en la cual se prueban datos de entrada y se identifican los datos de salida sin conocer el funcionamiento interno del componente, los pasos serán los siguientes:

- Identificación de las funcionalidades principales a partir de los casos de uso planteados anteriormente.
- Desarrollar los casos de pruebas para cada una de las funcionalidades identificadas.
- Realizar los casos de prueba tabulando los datos de entrada y los datos de salida.

Los resultados de la prueba fueron correctos y se encuentran detallados en el apéndice D.

7.5.5. Pruebas de seguridad. Debido a que la información principalmente esta almacenada en el Backend y es accesible por medio de la API, se debe verificar cualquier posible vulnerabilidad en ella. Las principales posibles vulnerabilidades que se identificaron fueron las siguientes:

- **Rutas privadas expuestas al público:** Las rutas privadas de la API, es decir, las rutas que necesitan token de autenticación por parte de un usuario están protegidas por un middleware de autenticación que se encarga de verificar que la solicitud contenga un token y que este corresponda a un usuario que se haya autenticado recientemente. Las rutas que están protegidas se pueden observar en la tabla de rutas de la sección 7.3.2. Las solicitudes se realizaron con la herramienta Insomnia y los resultados de la verificación se pueden observar en la siguiente tabla:

Tabla 36. *Resultados prueba de seguridad*

RUTA	Resultado de la solicitud	
	Solicitud sin token de autenticación	Solicitud con token de autenticación
/logout	unauthenticated	ok
/user	unauthenticated	ok
/paises	unauthenticated	ok
/ciudades	unauthenticated	ok
/searchFilosofo	unauthenticated	ok
/postFilosofo	unauthenticated	ok
/patchFilosofo	unauthenticated	ok
/deleteFilosofo	unauthenticated	ok
/postAporte	unauthenticated	ok
/patchAporte	unauthenticated	ok
/deleteAporte	unauthenticated	ok
/getDataRelacionadosAporte	unauthenticated	ok
/postEvidencia	unauthenticated	ok
/patchEvidencia	unauthenticated	ok
/deleteEvidencia	unauthenticated	ok
/getDataParticipantes	unauthenticated	ok
/postPeriodo	unauthenticated	ok
/deletePeriodo	unauthenticated	ok
/patchPeriodo	unauthenticated	ok
/postFotografia	unauthenticated	ok

/patchFotografia	unauthenticated	ok
/deleteFotografia	unauthenticated	ok
/getAdminFilosofos	unauthenticated	ok
/getBasicInfoAdmin	unauthenticated	ok
/updateUser	unauthenticated	ok
/changeAccessAdmin	unauthenticated	ok
/deleteAdminFilosofo	unauthenticated	ok
/addPermisoFilosofo	unauthenticated	ok
/getPermisosAdmin	unauthenticated	ok
/getSolicitudes	unauthenticated	ok
/acceptSolicitud	unauthenticated	ok
/rejectSolicitud	unauthenticated	ok
/getMensajesCurador	unauthenticated	ok
/getMensajesAdmin	unauthenticated	ok
/sendMessage	unauthenticated	ok
/addSeen	unauthenticated	ok

Se puede observar que sin token de autenticación el servidor devuelve como respuesta el mensaje unauthenticated, que corresponde al resultado esperado ya que la ruta está protegida por el middleware.

- Cifrado de seguridad en la comunicación entre el Frontend y el Backend: Para que la comunicación entre el Frontend y el Backend sea completamente segura se debe cifrar la comunicación, esto generalmente se hace por medio del cifrado de seguridad de la capa de transporte (TLS) que es un estándar para mantener la conexión privada. Para la implementación de este proyecto no se agregó esta medida de seguridad en el servidor de Backend debido a que se requieren más recursos económicos y porque el entorno virtual desarrollado sigue en estado de prototipo así que por el momento no es necesario.
- Accesibilidad al servidor por alguno de los puertos: Para acceder a la configuración del servidor se habilitó el puerto 22, esta se hace de manera remota por medio de herramientas como Putty para el sistema operativo Windows o por medio de la terminal de consola para sistemas operativos Linux. La autenticación para ingresar a la

configuración del servidor por medio de ssh (Secure Shell o Protocolo de administración remota) se hace por medio de claves ssh privadas y públicas, aumentando así la seguridad del servidor. La autenticación funciona de la siguiente manera:

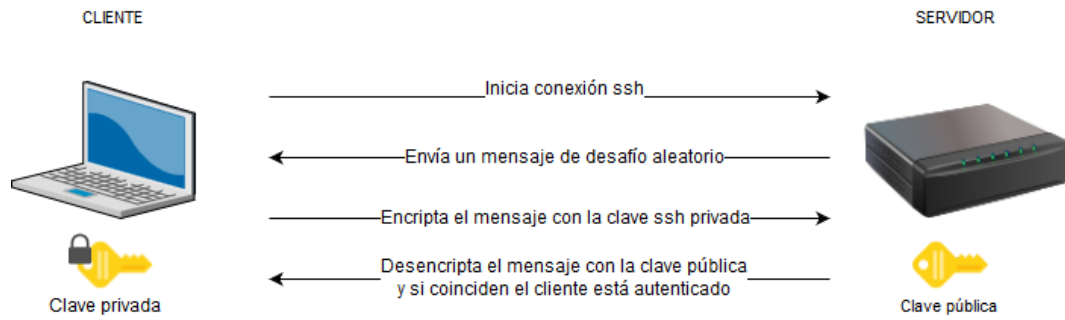


Figura 31. Autenticación ssh con clave privada y pública.

8. Conclusiones

- El objetivo general planteado se cumplió en su totalidad, ya que se logró desarrollar el entorno virtual en el cual se permite visualizar los aportes, evidencias y periodos de tiempo de los filósofos colombianos y sus relaciones con otros objetos, mediante una interfaz intuitiva y adaptable.
- Se realizaron múltiples reuniones que permitieron obtener los requerimientos y la estructura de datos del entorno virtual.
- Se implementó una arquitectura cliente-servidor de tres capas que permite la escalabilidad del sistema y facilidad en posibles futuras modificaciones en el Frontend y el Backend.
- Se desarrolló un panel de administración que permite a los usuarios del sistema gestionar fácilmente la información correspondiente a cada uno de los filósofos.
- El entorno virtual se instaló correctamente en un VPS (Virtual Private Server) de DigitalOcean para fines de evaluación y pruebas.
- Se diseñó un plan de pruebas y se aplicó correctamente permitiendo corregir errores existentes y validando las funcionalidades del entorno virtual.
- El desarrollo de este proyecto ayuda significativamente a la comunidad filosófica proveyendo una herramienta moderna que permite a los usuarios obtener recursos bibliográficos para fines de estudio e informarse del estado actual de la filosofía colombiana.

9. Recomendaciones

- Agregar funcionalidades donde se promueva la participación de los visitantes, como foros y comentarios en aportes, evidencias y los perfiles de los filósofos. También una sección de estadísticas dónde se pueda encontrar información relacionada a todos los filósofos que se encuentran en la plataforma y que permita ver un panorama general del estado actual de la filosofía.
- Incluir un módulo que permita la comunicación entre los diferentes tipos de usuario que existen en el entorno virtual.
- Mejorar la seguridad del servidor agregando un certificado SSL que permita establecer una conexión segura entre el cliente y el servidor.
- Proponer el desarrollo de otros entornos virtuales que permitan a los visitantes encontrar información sobre el estado actual del país en otros temas como política, deporte, etc.

Referencias bibliográficas

- Acosta García, D. F., & Torres Gaona, S. J. (2016). *Herramienta Web para la construcción colectiva de una biblioteca digital de recursos electrónicos sobre clásicos de la filosofía (tesis de pregrado)*. Bucaramanga, Santander: Universidad Industrial de Santander.
- Bahit, E. (2011). *Poo y mvc en php. El paradigma de la Programación*.
- Banco de la República de Colombia. (1997). *Banrepcultural*. Recuperado el 25 de Octubre de 2018, de Red cultural del Banco de la República de Colombia:
<http://www.banrepcultural.org/>
- Belloch, C. (s.f). *Entornos virtuales*. Recuperado el 24 de Marzo de 2018, de Sitio Web Universidad de Valencia: <https://www.uv.es/bellohc/pedagogia/EVA3.pdf>
- Booch, G. (1994). *Object-Oriented Analysis and Design with Applications, 2a. ed.* Redwood City, CA: Benjamin/Cummings.
- Bravo Contreras, M. C., Pavón Gutiérrez, F., Reyes Ortiz, J. A., & Alcántara Ramírez, R. A. (Febrero de 2018). *Tecnológico Nacional de México*. Recuperado el 26 de Abril de 2018, de Sitio Web Tecnológico Nacional de México:
<http://www.itcelaya.edu.mx/ojs/index.php/pistas/article/viewFile/1173/949>
- Bruegge, B., & H. Dutoit, A. (2002). *Object Oriented Software Engineering*. New Jersey, E.U.A: Prentice Hall Inc.
- Carrillo Garavito, Y. L. (2014). *Archivo digital para filósofo colombiano (tesis de pregrado)*. Bucaramanga, Santander: Universidad Industrial de Santander.
- Castells, P. (2003). *La Web Semántica*. Recuperado el 12 de Noviembre de 2018, de Servidor Moodle de la Facultad de Ingeniería Pontificia Universidad Javeriana Seccional Cali:
<http://cic.puj.edu.co/wiki/lib/exe/fetch.php?media=materias:castells-uclm03.pdf>
- Facultad de humanidades Universidad del Valle. (2006). *Universidad del Valle*. Obtenido de Universidad del Valle, Estanislao Zuleta: <http://catedraestanislao.univalle.edu.co/>
- Fernández Breis, J. T. (Marzo de 2013). *TDX, Tesis doctorales en Xarxa*. Recuperado el 26 de Abril de 2018, de Repositorio digital de tesis doctorales:
<http://www.tdx.cat/bitstream/handle/10803/10921/FernandezBreis.pdf>
- Fouce, J. M. (Octubre de 2001). *Webdianoia*. Recuperado el 27 de Octubre de 2018, de La filosofía en bachillerato: <https://www.webdianoia.com/index.html>
- Fundación Dialnet y Universidad de la Rioja. (s.f.). *Fundación Dialnet*. Recuperado el 26 de Abril de 2018, de Fundación Dialnet: <https://dialnet.unirioja.es/servlet/autor?codigo=69>
- Fundacion para el apoyo a la investigacion del estado de Sao paulo [FAPESP]. (1998). *SciELO*. Recuperado el 25 de Octubre de 2018, de Scientific Electronic Library Online (Biblioteca Científica Electrónica en Línea): <http://www.scielo.org/php/index.php>

- Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript*. Barcelona: Marcombo, S.A.
- Gomez Nossa, S. K., & Hernández Serrano, J. K. (2015). *Sistema de información para administrar objetos bibliográficos digitales*. Bucaramanga, Santander: Universidad Industrial de Santander.
- Herrera Ríos, E. (2011). *Arrancar con HTML5* (1 ed.). México, D.F.: Alfaomega Grupo Editor.
- Hyraldo Trejo, R. (2013). *Uso de los entornos virtuales de aprendizaje*. Recuperado el 29 de Marzo de 2018, de Universidad Estatal a Distancia:
https://www.uned.ac.cr/academica/edutec/memoria/ponencias/hiraldo_162.pdf
- Jacobson, I., Booch, G., & Rumbaugh, J. (2000). *El proceso unificado de desarrollo de software*. Madrid: Pearson Education. S.A.
- Jadhav, M., Sawant, B., & Deshmukh, A. (2015). Single page application using angularjs [Aplicaciones de una sola página usando angularjs]. *International Journal of Computer Science and Information Technologies*, 2876-2879.
- Ley N.º 1030. (22 de junio de 2006). Congreso de la república de Colombia. Colombia.
- Marini, E. (Octubre de 2012). *El modelo Cliente/Servidor*. Recuperado el 28 de Diciembre de 2018, de <https://www.linuxito.com/docs/el-modelo-cliente-servidor.pdf>
- Martínez Villalobos, G., Camacho Sánchez, G., & Biancha Gutiérrez, D. (2010). DISEÑO DE FRAMEWORK WEB PARA EL DESARROLLO DINÁMICO DE APLICACIONES. *Scientia Et Technica*, 178-183.
- Microsoft Corporation. (5 de Abril de 2014). *Implementing the MVVM Pattern Using the Prism Library 5.0 for WPF*. Recuperado el 6 de Diciembre de 2018, de [https://docs.microsoft.com/en-us/previous-versions/msp-n-p/gg405484\(v=pandp.40\)](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/gg405484(v=pandp.40))
- Pressman, R. (2010). *Ingeniería del software Un enfoque práctico* (Séptima Edición ed.). México, D. F.: McGraw Hill.
- Restrepo, R. (30 de marzo de 2011). ¿Dónde están los filósofos? *Revista Arcadia*. Obtenido de <https://www.revistaarcadia.com/impres/filosofia/articulo/donde-estan-filosofos/24577>
- Rouse, M. (2018). *What is MySQL?* Recuperado el 5 de Diciembre de 2018, de <https://searchoracle.techtarget.com/definition/MySQL>
- Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., & Lorensen, W. (1991). *Object-Oriented Modeling and Design*. Englewood Cliffs, NJ: Prentice Hall.
- Rumbaugh, J., Jacobson, I., & Booch, G. (2000). *El lenguaje unificado de modelado. Manual de referencia*. Madrid: Pearson Education. S.A.
- Salinas, M. I. (s.f.). *Pontificia Universidad Católica Argentina*. Recuperado el 27 de 10 de 2018, de Entornos virtuales de aprendizaje en la escuela: tipos, modelo didactico y rol de docente: <http://eduteka.icesi.edu.co/gp/upload/Educaci%C3%B3n%20EVA.pdf>
- Saunier, R. (2014). *Getting started with Laravel 4*. Birmingham, UK: Packt Publishing Ltd.

- Tabares, M. S., Barrera, A. F., Arroyave, J. D., & Pineda, J. D. (2007). Un método para la trazabilidad de requisitos en el proceso unificado de desarrollo. *Revista EIA*, 69-82.
- Universidad Autónoma del Estado de México. (2003). *Red de Revistas Científicas de América Latina y el Caribe, España y Portugal*. Recuperado el 25 de Octubre de 2018, de Portal de revistas de y para ciencias sociales y humanidades: <https://www.redalyc.org/home.oa>
- Universidad de la Rioja. (2001). *Dialnet*. Recuperado el 25 de Octubre de 2018, de Portal bibliográfico Dialnet: <https://dialnet.unirioja.es/>
- Uniwebsidad. (s.f.). *Uniwebsidad - Los mejores recursos para aprender diseño y programación web*. Recuperado el 28 de Noviembre de 2018, de <https://uniwebsidad.com/libros/css/capitulo-1/breve-historia-de-css>
- Weitzenfeld, A. (2005). *Ingeniería de Software Orientada a Objetos con UML, Java e Internet*. México: International Thomson Editores, S.A de C.V.
- You, E. (2014). *VueJs Guide to the progressive JavasCrpt Framework*. Recuperado el 6 de Diciembre de 2018, de <https://vuejs.org/v2/guide/>
- Zulian, E. R. (2010). *Implementación de un framework para el desarrollo de aplicaciones web utilizando patrones de diseño y arquitectura MVC/REST*. Buenos Aires, Argentina: Universidad de Belgrano.

Apéndices

Apéndice A. Diagramas de flujo.

A continuación, se muestran los diagramas de flujo realizados a partir de los casos de uso presentados en el capítulo 7. Se puede observar con más detalle el flujo de acciones que debe tomar un usuario para realizar una operación en el sistema.

- **Buscar información:** Se muestran tres diagramas debido el módulo de búsqueda no es la única manera de buscar información en el entorno virtual, también se puede encontrar información en la página principal y dentro de los perfiles de los filósofos y aportes.

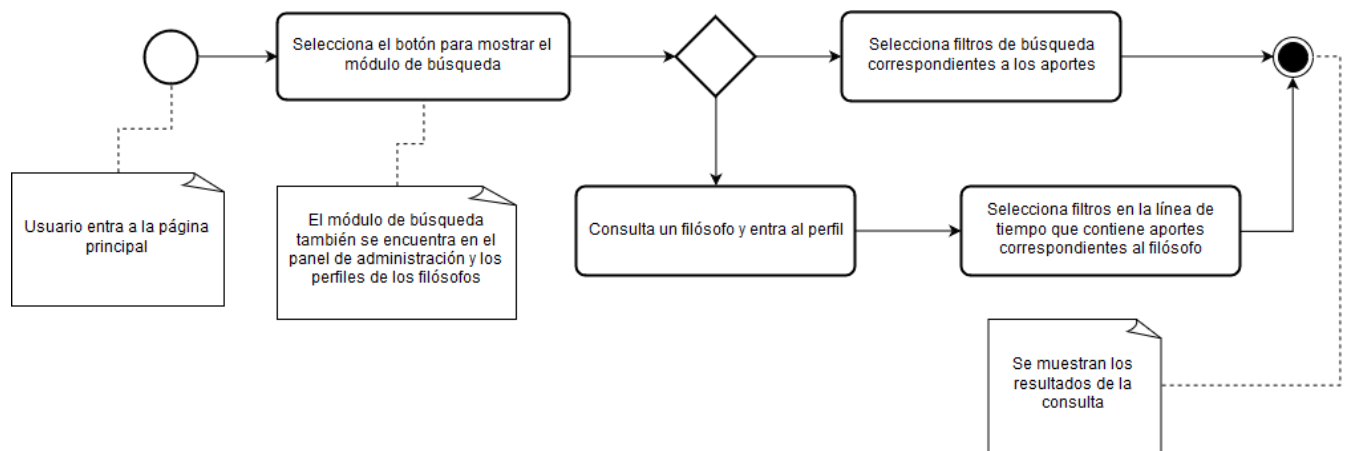


Figura. Diagrama de flujo - Búsqueda de aportes.

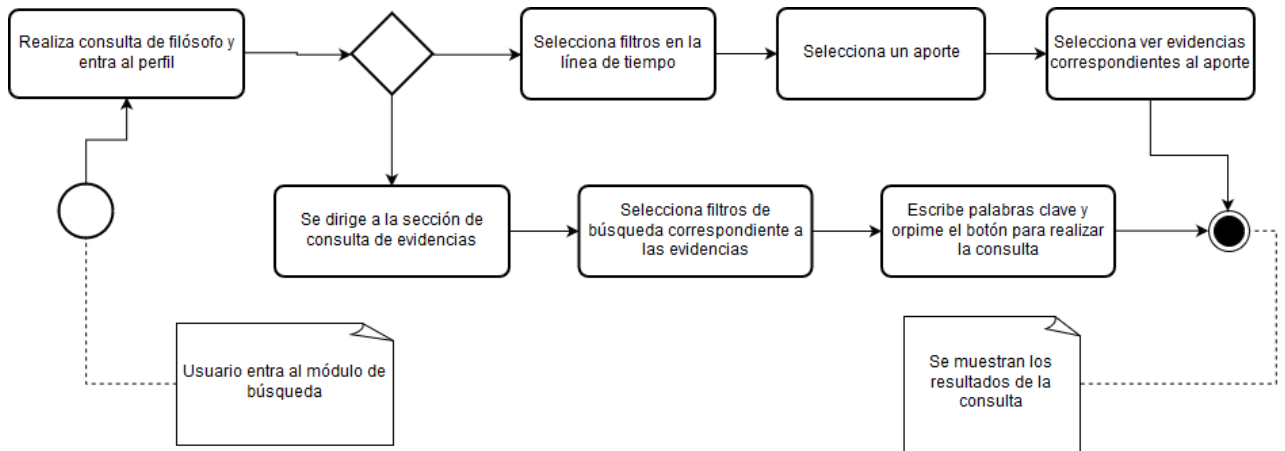


Figura. Diagrama de flujo – Buscar evidencias

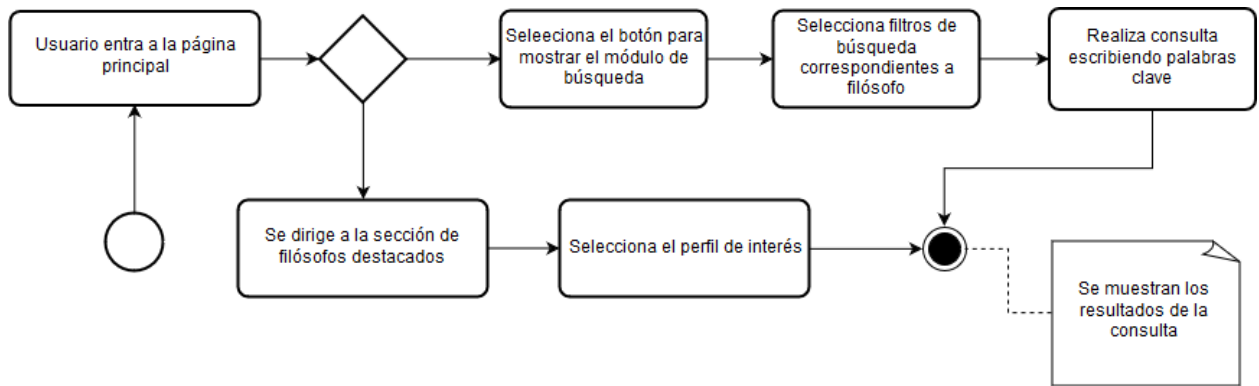


Figura. Diagrama de flujo – Buscar evidencias

- Contactar Administrador/Curador

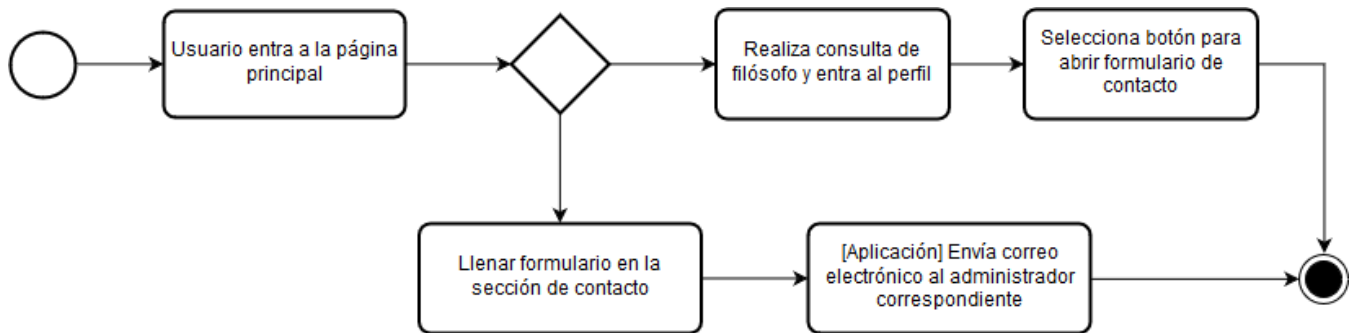


Figura. Diagrama de flujo – Contactar Administrador/Curador

- Enviar solicitud de registro

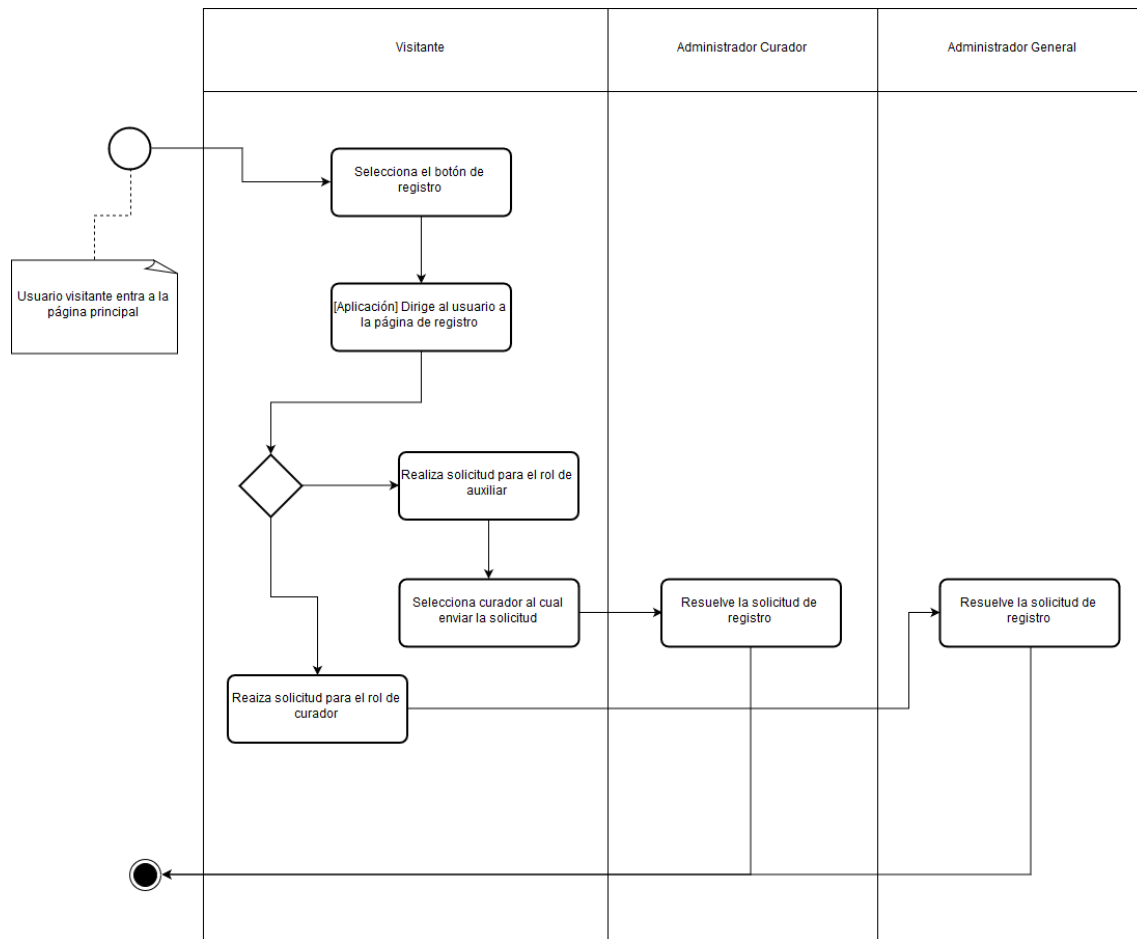


Figura. Diagrama de flujo – Enviar solicitud de registro

- Gestionar Acceso: Se muestran los diagramas de flujo para los dos distintos actores, Administrador y Curador.

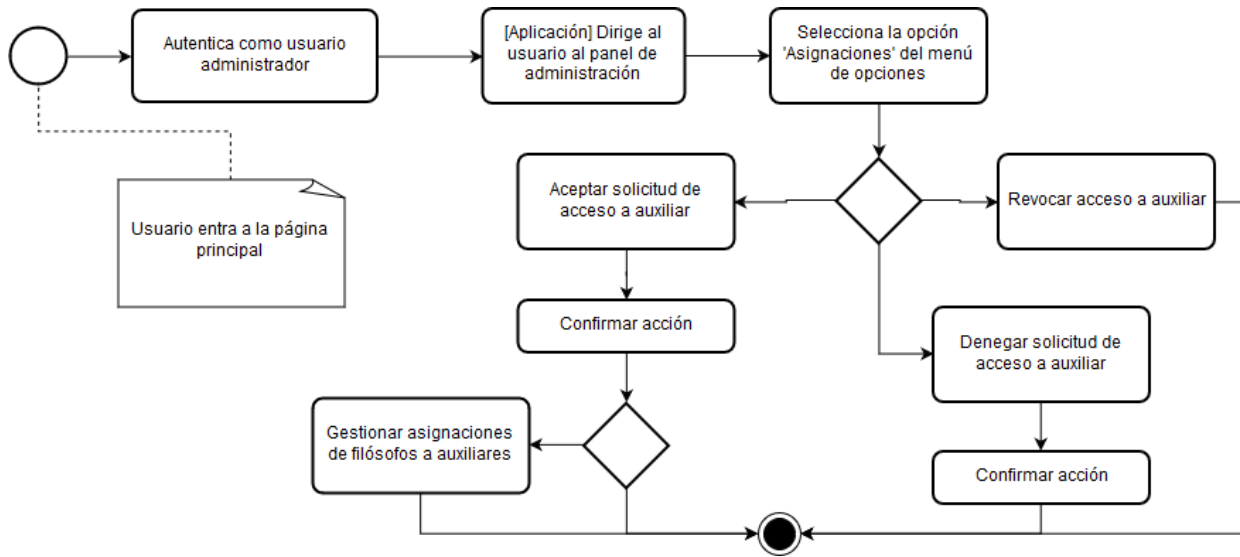


Figura. Diagrama de flujo – Gestionar acceso (Curador)

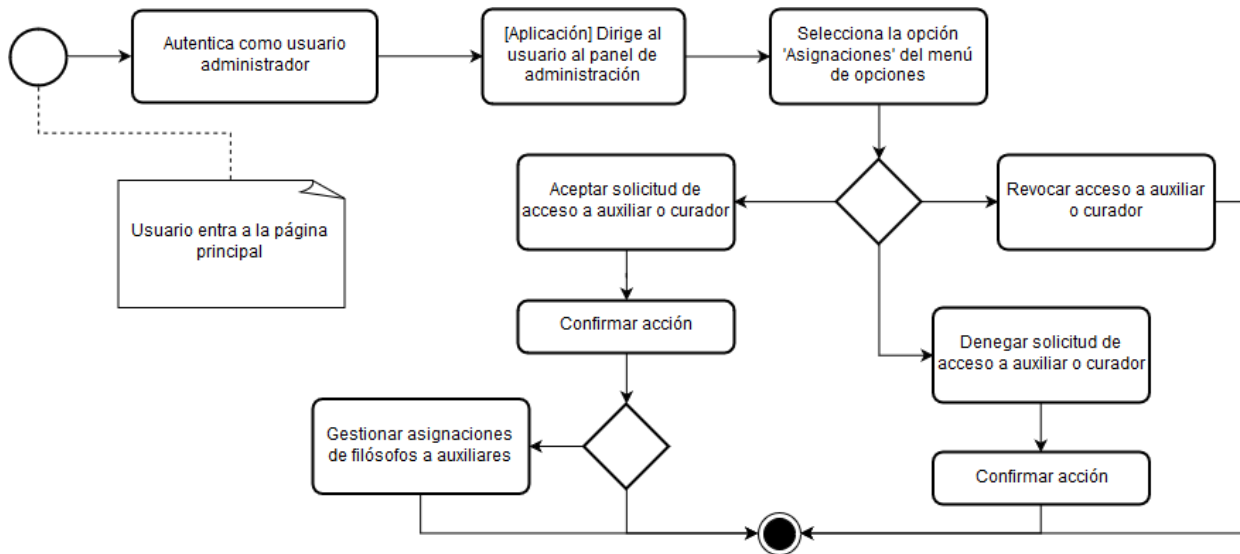


Figura. Diagrama de flujo – Gestionar acceso (Administrador)

- Gestionar aportes

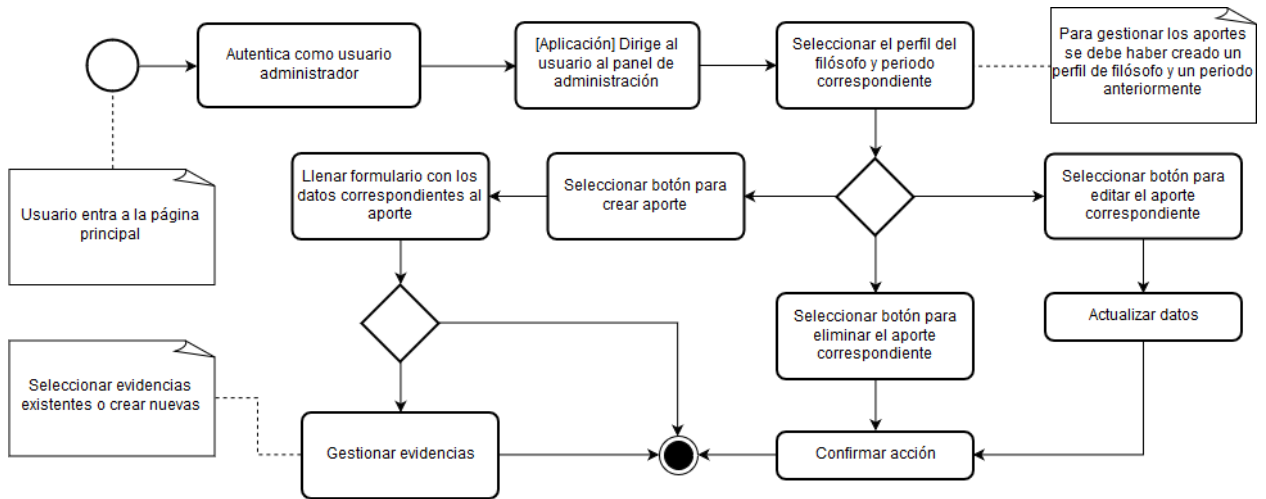


Figura. Diagrama de flujo – Gestionar aportes

- Gestionar evidencias

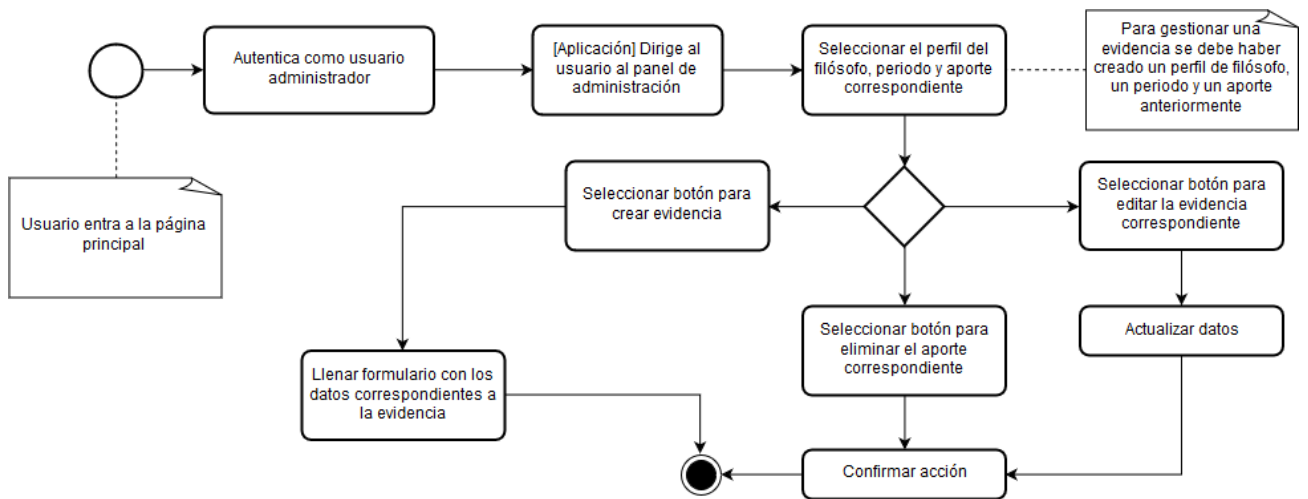


Figura. Diagrama de flujo – Gestionar evidencias

- Gestionar filósofo

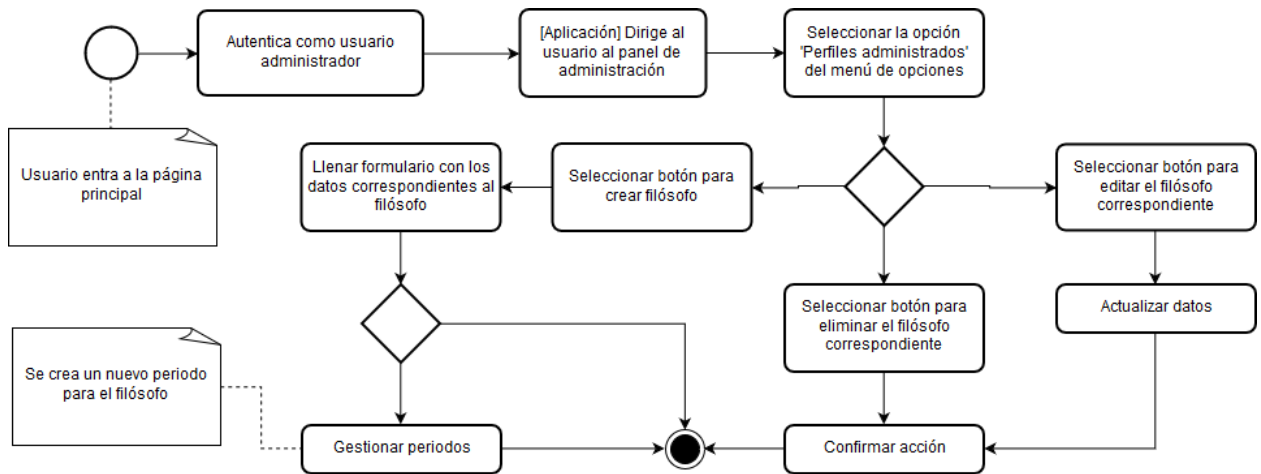


Figura. Diagrama de flujo – Gestionar filósofo

- Gestionar periodo

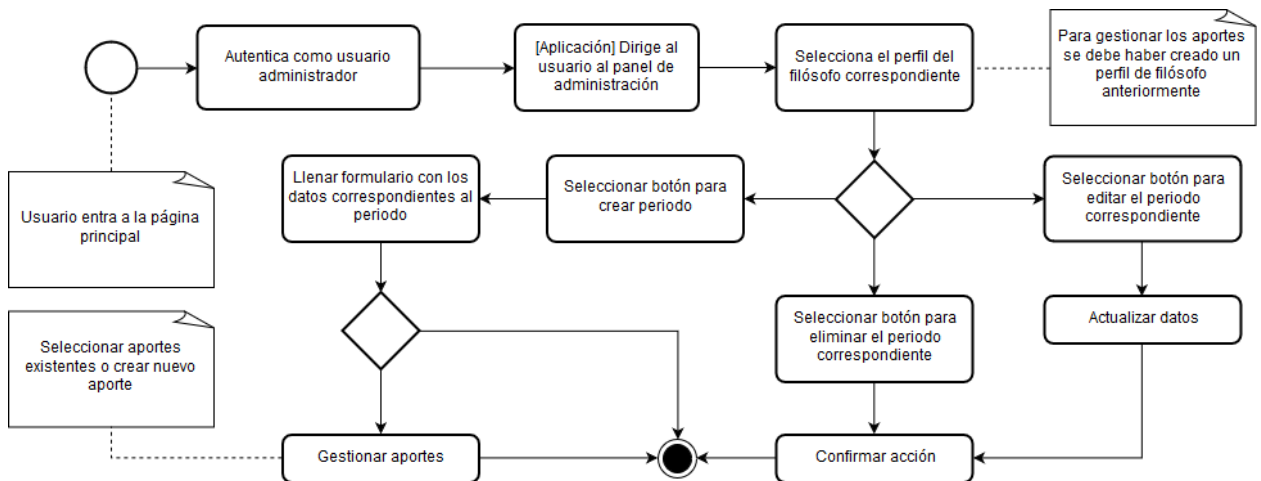


Figura. Diagrama de flujo – Gestionar aporte

- Gestionar asignaciones de filósofos a Curadores/Auxiliares

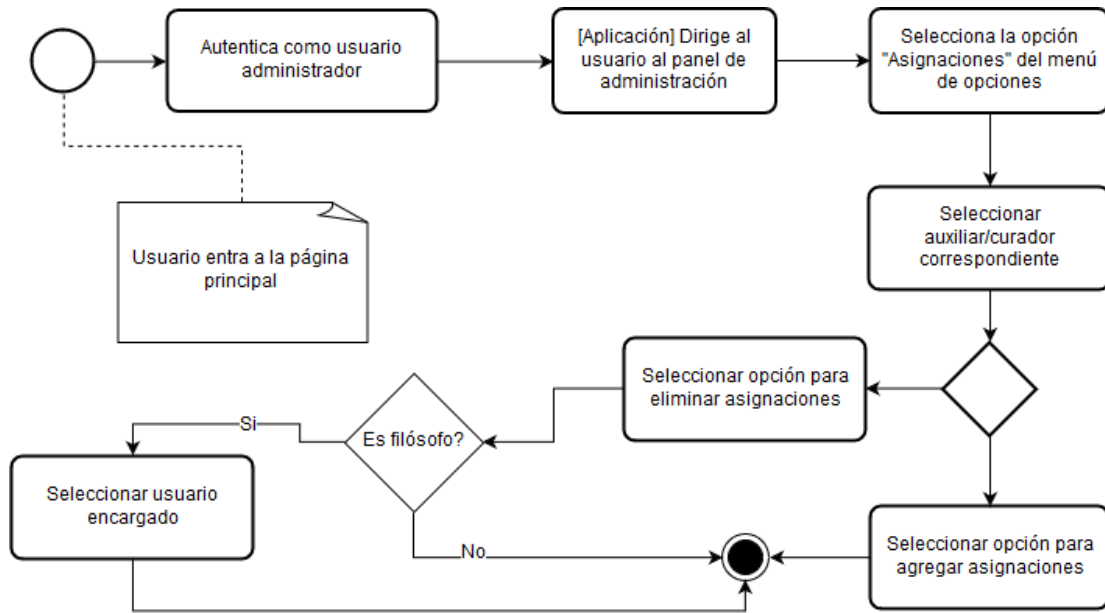


Figura. Diagrama de flujo – Gestionar asignaciones de filósofos a curadores/auxiliares

- Gestionar información de usuario

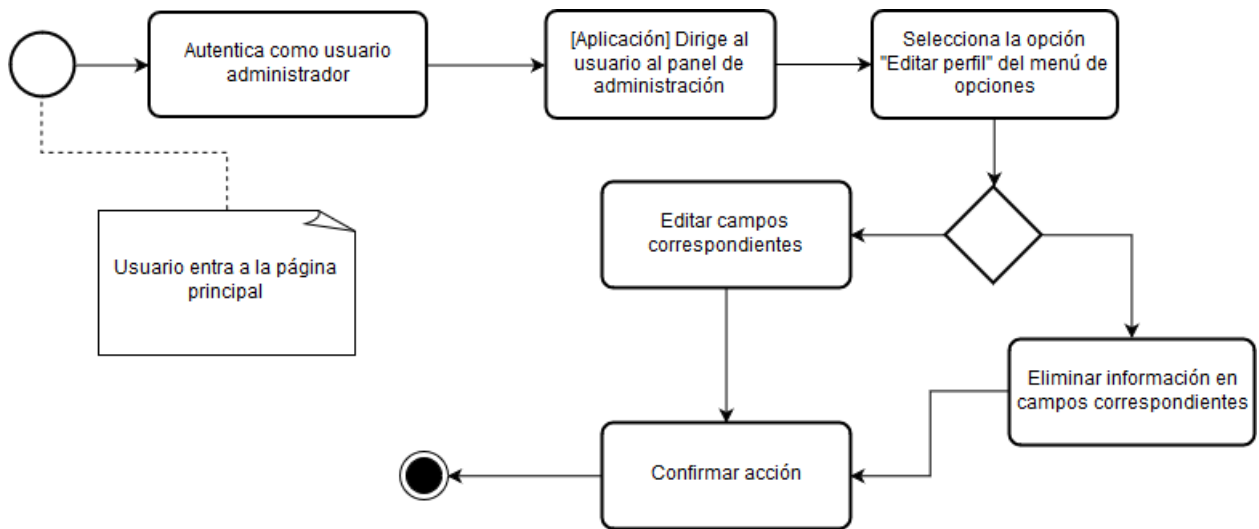


Figura. Diagrama de flujo – Gestionar información de usuario

- Autenticar usuario

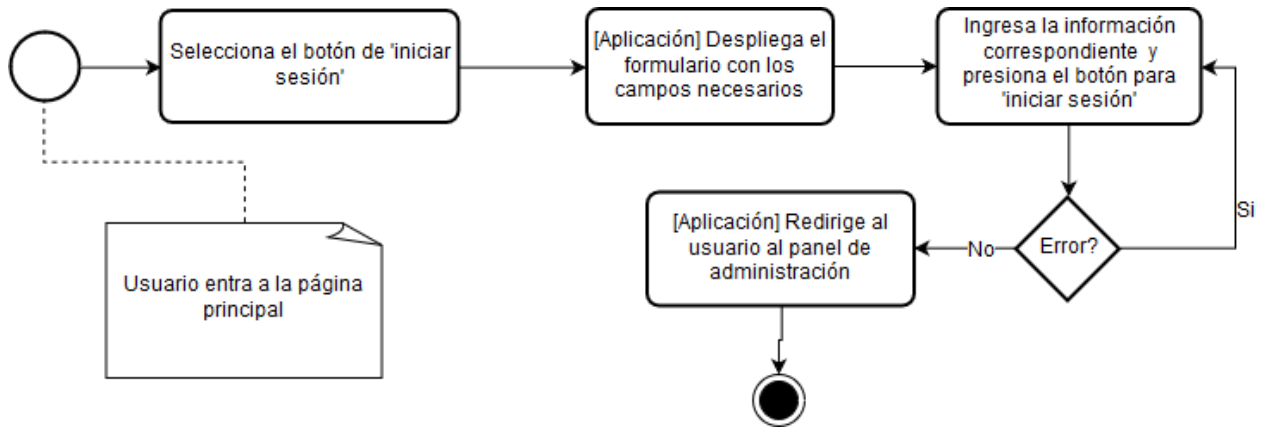


Figura. Diagrama de flujo – Autenticar usuario

- Recuperar contraseña

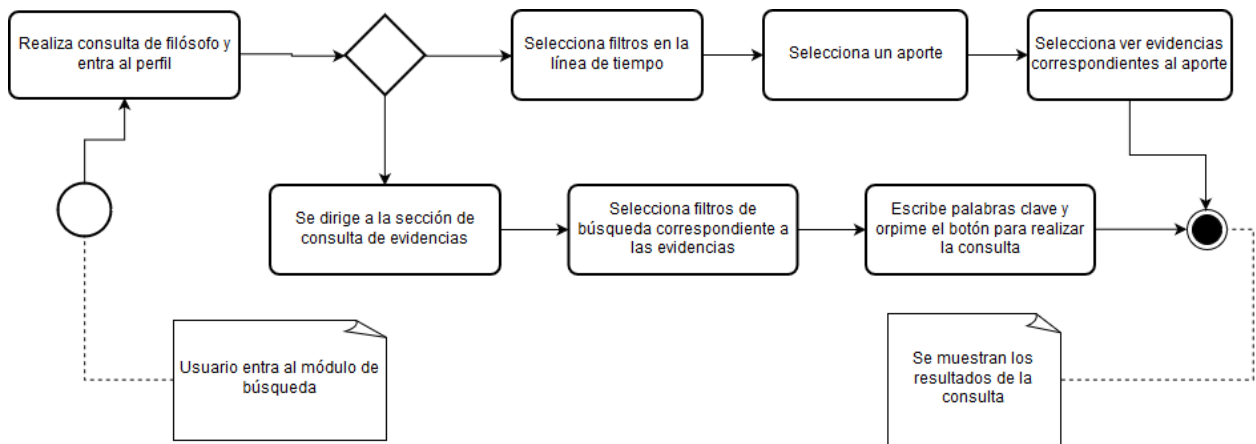


Figura. Diagrama de flujo – Recuperar contraseña

- Gestionar mensajes de contacto administrador/curador

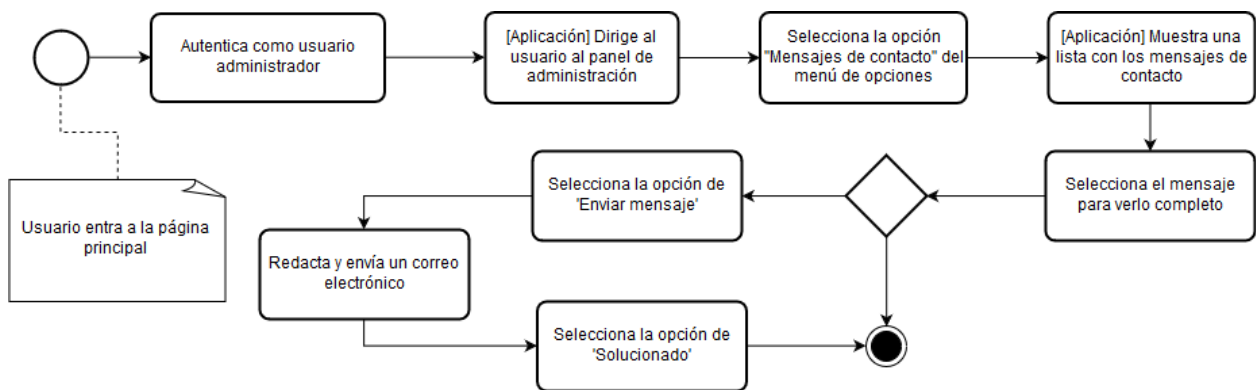


Figura. Diagrama de flujo – Gestionar mensajes de contacto administrador/curador

Apéndice B. Diccionario de datos.

A continuación, se muestra la descripción de los atributos de las entidades mostradas en el diagrama entidad relación presentado en el numeral 7. Las tablas del diagrama que no se muestran es debido a que no contienen algún atributo relevante además de las claves primarias y foráneas.

Tabla. *Filósofo*

Filósofo		
Atributo	Tipo de dato	Descripción
Id	INT	Clave única para el registro de cada filósofo
Nombre	TEXT	Nombres del filósofo
Sexo	TEXT	Se define si el filósofo es hombre o mujer
Tema	TEXT	Define el tema del perfil del filósofo
Fecha_nacimiento	TEXT	Fecha de nacimiento del filósofo
Fecha_muerte	TEXT	Fecha de fallecimiento del filósofo en caso de que corresponda
Ciudad_nacimiento	TEXT	Ciudad de nacimiento del filósofo
Pais_nacimiento	TEXT	País de nacimiento del filósofo
Nacionalidad	TEXT	Nacionalidad del filósofo
Biografía	TEXT	El texto correspondiente a la biografía del filósofo
Image_url	Varchar (255)	Ruta en el servidor de la imagen de perfil

Tabla. *Periodo*

Periodo		
Atributo	Tipo de dato	Descripción
Id	INT	Clave única para el registro de cada periodo
Titulo	TEXT	Título del periodo de tiempo relevante para el filósofo
Descripción	TEXT	Descripción del periodo de tiempo

Fecha_inicio	TEXT	Fecha de inicio del periodo de tiempo
Fecha_final	TEXT	Fecha final del periodo de tiempo
Ciudad	TEXT	Ciudad en la que ocurrió el periodo de tiempo
País	TEXT	País en el que ocurrió el periodo de tiempo
Image_url	Varchar (255)	Ruta en el servidor de la imagen del periodo
Filosofo_id	INT	Clave foránea de la entidad filósofo a la cual está relacionado el periodo de tiempo

Tabla. *Aporte*

Aporte		
Atributo	Tipo de dato	Descripción
Id	INT	Clave única para el registro de cada aporte
Nombre	TEXT	Nombre del aporte
Fecha	TEXT	Fecha en la que se realizó o se documentó el respectivo aporte
Descripción	TEXT	Descripción del aporte
Ciudad	TEXT	Ciudad en la que se realizó el aporte
País	TEXT	País en el que se realizó el aporte
Image_url	Varchar(255)	Ruta en el servidor de la imagen del aporte
Periodo_id	INT	Clave foránea del periodo al que corresponde el aporte
Contexto_id	INT	Clave foránea del contexto en el cual se realizó el aporte
Filosofo_id	INT	Clave foránea del filósofo al que corresponde el aporte

Tabla. *Evidencia*

Evidencia		
Atributo	Tipo de dato	Descripción
Id	INT	Clave única para el registro de cada evidencia

Autor	TEXT	Nombre del autor de la evidencia, se tiene en cuenta que el autor de la evidencia puede que no sea el filósofo
Descripción	TEXT	Descripción de la evidencia
Ciudad	TEXT	Ciudad en la que se creó la evidencia
País	TEXT	País en el que se creó la evidencia
Fecha	TEXT	Fecha en la que se creó la evidencia
Idioma	TEXT	Idioma en la que se encuentra la evidencia
Tipo_evidencia	TEXT	La evidencia puede ser de varios tipos: Imagen, Documento, Audio, Enlace a video de YouTube
Imagen_url	Varchar (255)	Enlace al recurso donde se encuentra alojada la imagen en el servidor, solo se llena este campo en caso de que el tipo de evidencia sea Imagen
Video_url	Varchar (255)	Enlace al video en YouTube donde se encuentra la evidencia, solo se llena este campo en caso de que el tipo de evidencia sea Enlace a video de YouTube
Documento_url	Varchar (255)	Enlace al recurso donde se encuentra alojado el documento en el servidor, solo se llena este campo en caso de que el tipo de evidencia sea Documento
Aporte_id	INT	Clave foránea del aporte al que pertenece la evidencia

Tabla. *Fotografía*

Fotografía		
Atributo	Tipo de dato	Descripción
Id	INT	Clave única para el registro de cada fotografía
Autor	TEXT	Autor de la fotografía
Título	TEXT	Título de la fotografía
Descripción	TEXT	Descripción de la fotografía
Ciudad	TEXT	Ciudad en la que se tomó la fotografía
Fecha	DATE	Fecha en la que se tomó la fotografía
País	TEXT	País en el que se tomó la fotografía
Image_url	Varchar (255)	Enlace al recurso donde se encuentra alojada la imagen de la fotografía en el servidor

Periodo_id	INT	Periodo al que corresponde dicha fotografía
------------	-----	---

Tabla. *Participante externo*

Participante externo		
Atributo	Tipo de dato	Descripción
Id	INT	Clave única para el registro de cada participante externo, (persona que aparece en una evidencia, pero no está dentro del entorno virtual)
Nombre	TEXT	Nombre del participante
Enlace_externo	TEXT	Enlace a la página web del participante externo (Principal red social, Wikipedia, etc.)

Tabla. *Usuario*

Usuario		
Atributo	Tipo de dato	Descripción
Id	INT	Clave única para el registro de cada usuario
Nombre	TEXT	Nombre del usuario
Correo	TEXT	Correo electrónico del usuario
Contraseña	Varchar (255)	Contraseña de acceso del usuario (No se guarda en texto plano, se utiliza un método de encriptación)
Rol	TEXT	Rol de usuario dentro del entorno virtual, puede ser: Auxiliar, Curador o Administrador
Image_url	Varchar (255)	Enlace al recurso donde se encuentra alojada la imagen de perfil de usuario en el servidor
Permiso_acceso	BOOLEAN	Determina si el usuario tiene permiso de acceso al entorno virtual

Tabla. *Solicitud*

Solicitud		
Atributo	Tipo de dato	Descripción
Id	INT	Clave única para el registro de cada solicitud de acceso

Estado	BOOLEAN	Determina si la solicitud ha sido aceptada (True) o si ha sido rechazada (False)
Tipo	Varchar (45)	Determina el tipo de rol de usuario que desea cumplir dentro del entorno virtual
User_id	INT	Clave foránea del usuario que realizó la solicitud

Tabla. *Contexto*

Contexto		
Atributo	Tipo de dato	Descripción
Id	INT	Clave única para el registro de cada contexto social
nombre	TEXT	Nombre del contexto social

Tabla. *Mensaje*

Mensaje		
Atributo	Tipo de dato	Descripción
Id	INT	Clave única para el registro de cada mensaje
Tipo_mensaje	TEXT	Determina si el mensaje es para un administrador o para un curador
Asunto	TEXT	Asunto del mensaje
Contenido	TEXT	Contenido del mensaje
Correo	TEXT	Correo electrónico del autor del mensaje
Fecha	TEXT	Fecha en la que se envió el mensaje

Tabla. *Mensaje_usuario*

Mensaje_usuario		
Atributo	Tipo de dato	Descripción
Id	INT	Clave única para el registro de cada mensaje_usuario
Leído	BOOLEAN	Determina si el mensaje ha sido leído

Resuelto	BOOLEAN	Determina si el mensaje ha sido respondido por un administrador
Respuesta	TEXT	Respuesta del administrador al mensaje
User_id	INT	Clave foránea del usuario que ha interactuado con el mensaje
Mensaje_id	INT	Clave foránea del mensaje

Tabla. *Permiso_filosofo*

Permiso_filosofo		
Atributo	Tipo de dato	Descripción
Id	INT	Clave única para el registro de cada permiso_filosofo
Permiso	BOOLEAN	Determina si el usuario tiene permisos sobre el filosofo
User_id	BOOLEAN	Clave foránea del usuario
Filosofo_id	INT	Clave foránea del filósofo

Apéndice C. Vistas de usuario.

A continuación, se presentan todas las secciones de la interfaz.

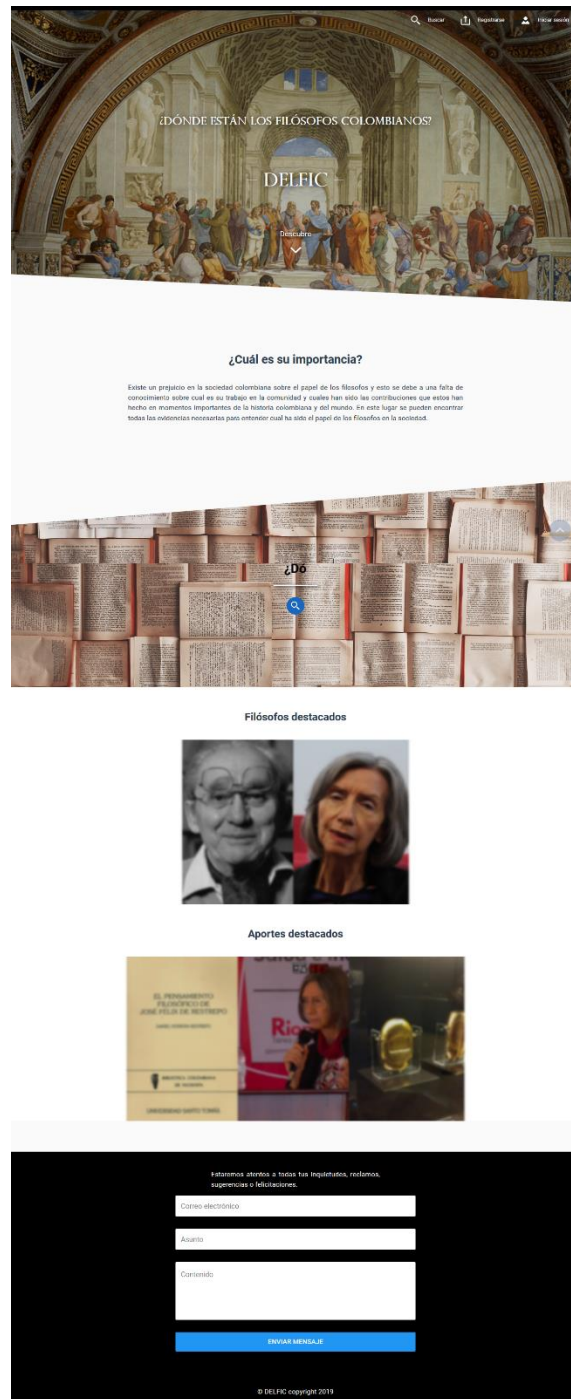


Figura. Página principal de la plataforma.

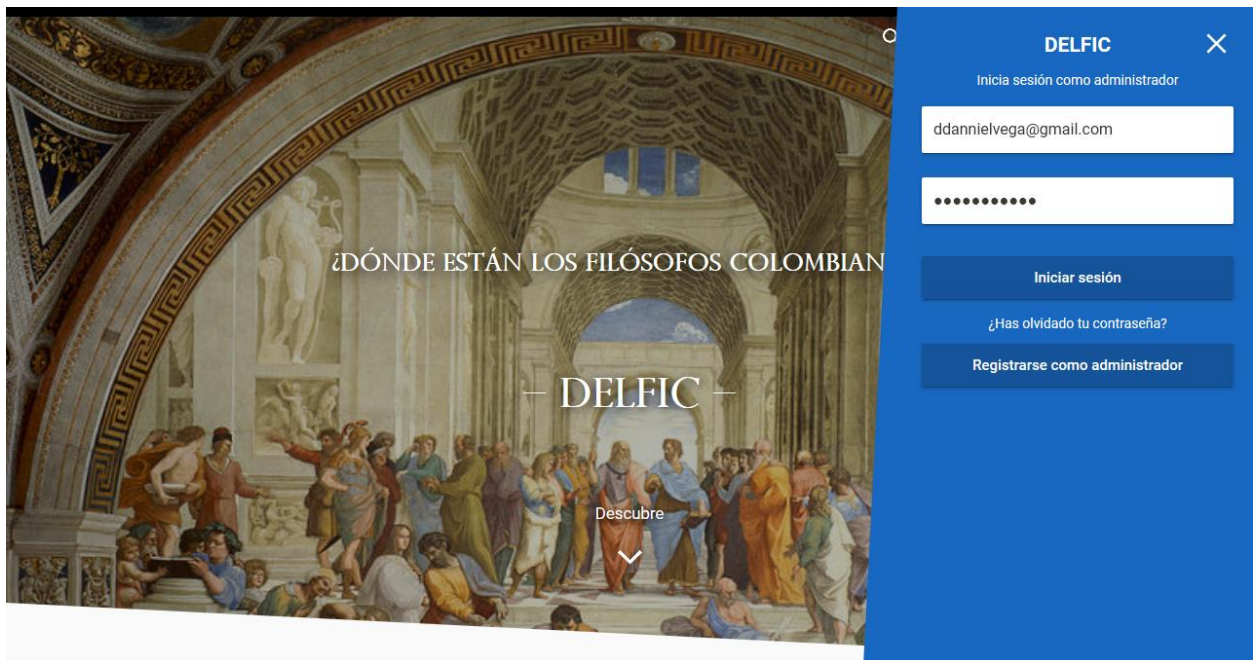


Figura. Login

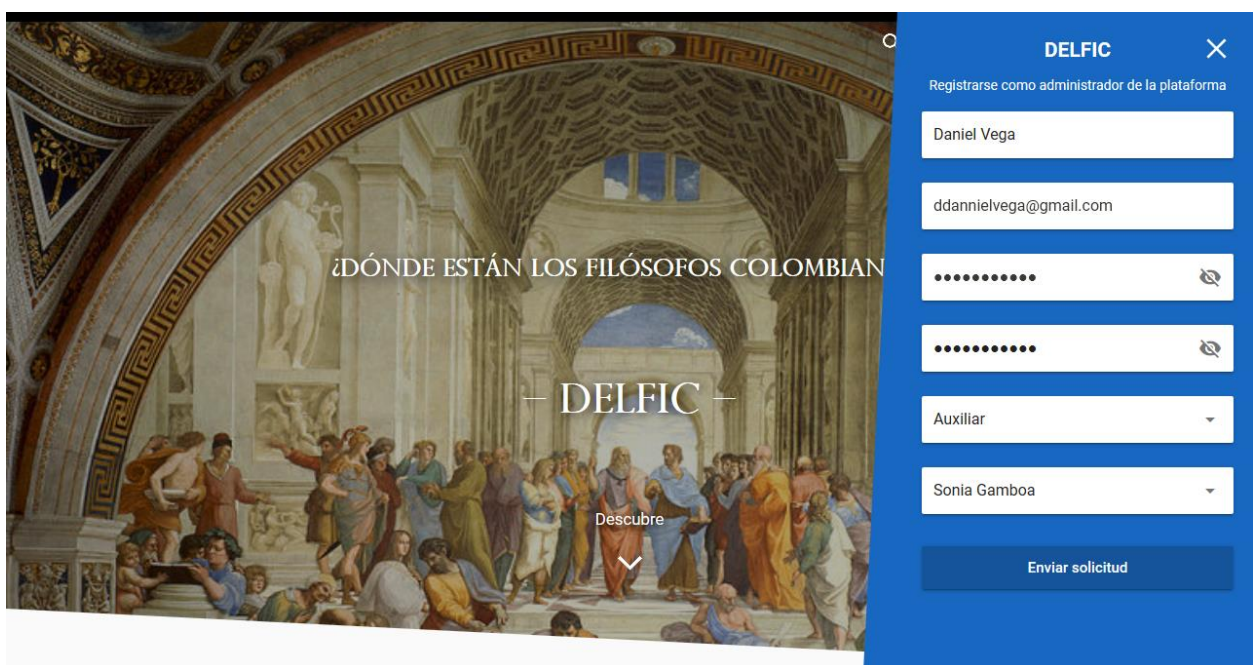


Figura. Registro.

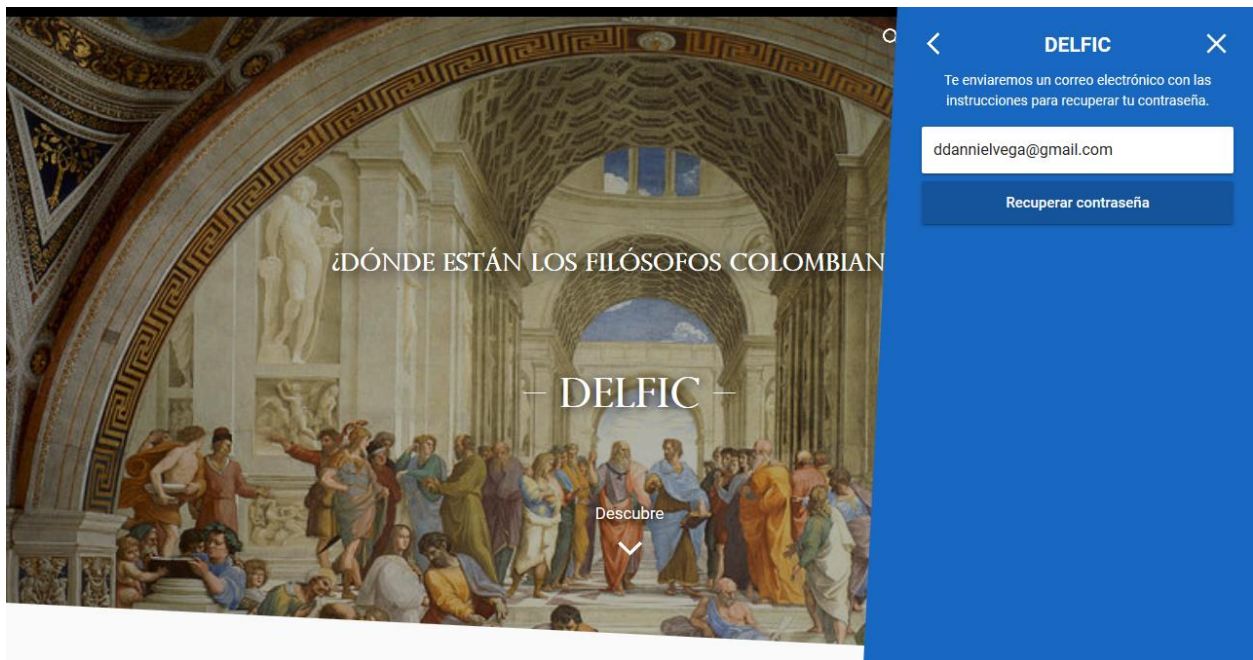


Figura. Recuperar contraseña.

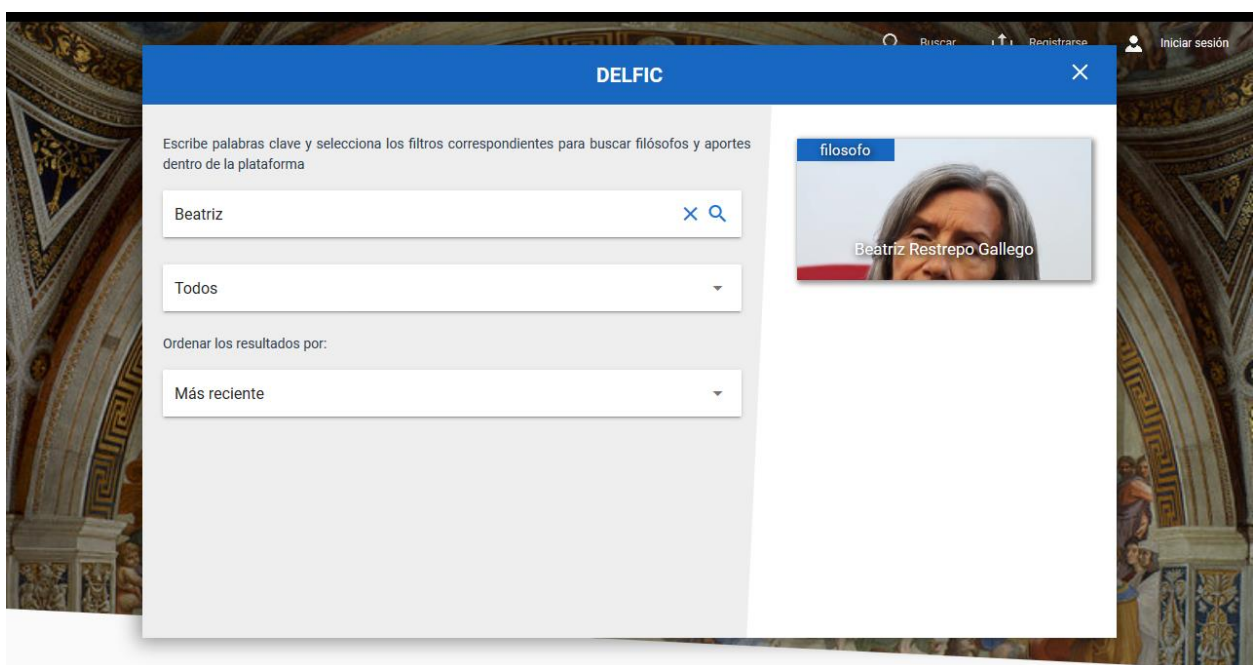



Figura. Buscador de aportes y filósofos.



Beatriz Restrepo Galego
Barranquilla, Colombia (1941)
Historiadora y filósofa


Nació en Barranquilla, en 1941. Su padre antropólogo, es la segunda en una familia de ocho hijos: cinco mujeres y tres hombres. Llegó a Medellín a los seis años, estudió la secundaria en el Colegio Sagrado Corazón, orientado por Martha Beatriz Ibañez para el momento (en 1948). Terminó sus estudios de bachillerato a los 16 años, en 1957. En el Sagrado Corazón estudió la filosofía y la historia, y se graduó en el Colegio de la Inmaculada Concepción de Barranquilla.

Cultura 33% Educación 55% Política 12%


Línea de tiempo
Selecciona uno de los eventos mostrados en la línea de tiempo para ver más información; también puedes filtrar los aportes de la línea de tiempo con los filtros mostrados debajo de este texto. Puedes ver más información sobre los elementos seleccionados (de color azul y ubicados en el centro) haciendo clic sobre ellos. Para moverse en la línea de tiempo usa las flechas al lado de cada una de las líneas de tiempo. Los aportes de color azul corresponden al periodo seleccionado también de color azul.

Año: [dropdown] Contexto: [dropdown] País: [dropdown] Ciudad: [dropdown]


Periodos Relevantes




Estudios universitarios en USA y de maestría en Madrid, UCM
1998 - 2002




Docente y decana en las universidades UPB y Udea respectivamente
2001 - 2002




Empleo en la Gobernación de Antioquia, ONG's y empresas como KPM
2000-Mar-01 | 2012-Jun-16



Filosofía de José Félix de Restrepo
2010-Feb-1
Proyecto de grado




Creación de los primeros programas de filosofía e historia en la Udea
1976-Jun-12



Montaje de la maestra de cosmética
2008-May-17
Quindiyá, Museo de antropología UdeA


Aportes

Detalles de la línea de tiempo
A continuación puedes ver más información sobre los elementos seleccionados de las líneas de tiempo (ubicados en el centro y de color azul). Cada imagen corresponde a una evidencia o fotografía; haz clic en ella para ver más información.



Docente y decana en las universidades UPB y Udea respectivamente

Periodo de tiempo: Docente y decana en las universidades UPB y Udea respectivamente
Medellín - Colombia
Periodo de tiempo: Docente y decana en las universidades UPB y Udea respectivamente




Creación de los primeros programas de filosofía e historia en la Udea

12 Jul 1976
Educación

Aporte: Creación de los primeros programas de filosofía e historia en la Udea
Medellín - Colombia
Creación de los primeros programas de filosofía e historia en la Udea

Relacionados al aporte seleccionado
A continuación se muestran algunos filósofos y aportes relacionados al aporte seleccionado en la línea de tiempo.



Beatriz Restrepo Galego

Evidencias
Selecciona los filtros necesarios para buscar evidencias de algún aporte del filósofo.

Buscar [input] [icon]

Tipo: [dropdown] Año: [dropdown] Idioma: [dropdown]

EDULFI © copyright 2019

Figura. Perfil de filósofo.

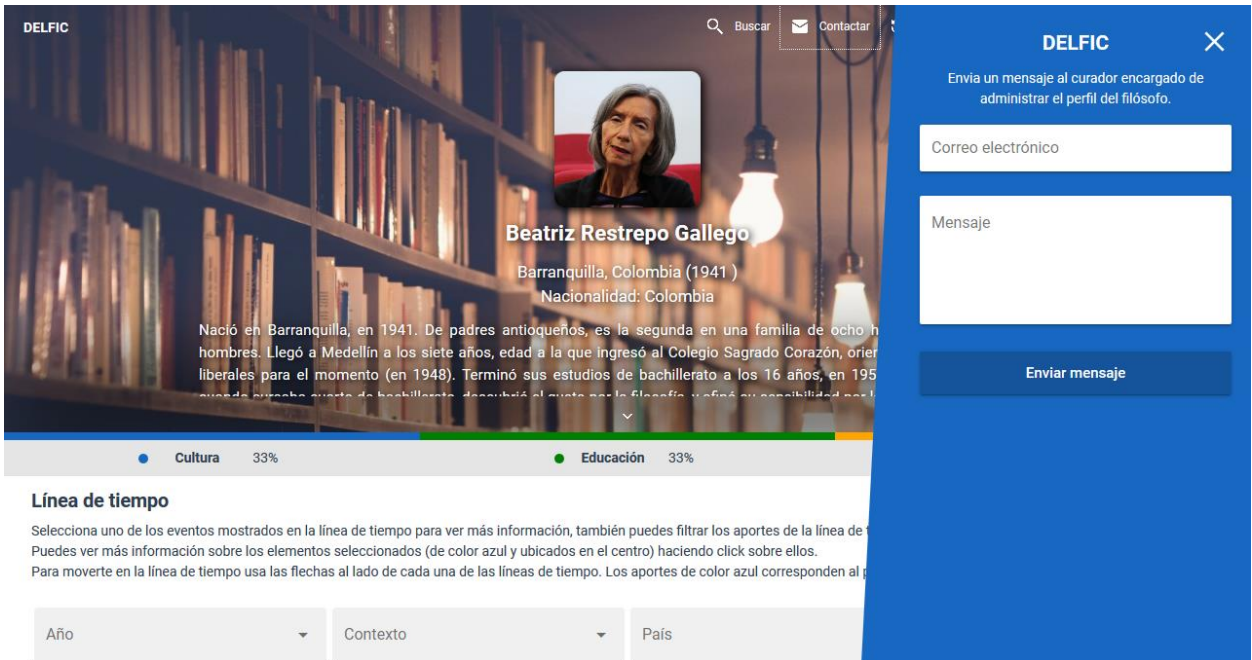


Figura. Contactar curador.



Figura. Ver detalles de fotografía/evidencia.

The screenshot displays the DELFIC web application interface. At the top, a blue navigation bar contains the logo 'DELFIC', a search icon labeled 'Buscar', a user profile icon labeled 'Panel Admin', and a logout icon labeled 'Cerrar sesión'. The main content area is divided into two columns. The left column features a large grey box with a white document icon and a red 'PDF' label, with the text 'Click sobre la imagen para ver más información' above it and 'Proyecto de grado Beatriz Restrepo' below it. Below this is a blue box with the text 'Política', a large number '1', and 'Mar 2010'. The right column contains a white box with the heading 'Aporte: Proyecto de grado' in blue, followed by the text 'Popayán - Colombia' and 'Proyecto de grado'. Below the main content, a section titled 'Relacionados al aporte seleccionado' includes a subtext: 'A continuación se muestran algunos filósofos y aportes relacionados al aporte seleccionado en la línea de tiempo.' Below this is a portrait of Beatriz Restrepo Gallego with her name 'Beatriz Restrepo Gallego' written underneath. A small blue arrow icon is visible in the bottom right corner of the main content area. The footer is a dark grey bar with the text '©DELFIC copyright 2018'.

Figura. Página Aporte.

The screenshot shows the 'Perfiles Administrados' section of the DELFIC administration panel. The left sidebar contains the following menu items: 'Panel de administración', 'Usuario: Daniel Vega C', 'Rol: Administrador', 'Perfiles administrados' (highlighted), 'Mensajes', 'Acceso a administradores', 'Editar perfil de usuario', and 'Búsqueda de objetos'. The main content area is titled 'Filósofos' and includes a search bar, a 'Cerrar sesión' button, and a list of philosopher profiles. The first profile is Beatriz Restrepo Gallego, with a red 'X' icon, a blue pencil icon, and a green plus icon. The second profile is Daniel Herrera Restrepo. Below the list is a '+ ' button. The section is followed by 'Periodos de tiempo | Beatriz Restrepo Gallego', which includes a description and a list of time periods: 'Estudios universitarios en USA y de maestría en Madrid, UCM', 'Docente y decana en las universidades UPB y Udea respectivamente', and 'Empleos en la Gobernación de Antioquia, ONG's y empresas como EPM'. Each period has a red 'X' icon and a right arrow.

Figura. Panel de administración – Perfiles Administrados.

The screenshot shows the 'Búsqueda de objetos' section of the DELFIC administration panel. The left sidebar contains the following menu items: 'Panel de administración', 'Usuario: Daniel Vega C', 'Rol: Administrador', 'Perfiles administrados', 'Mensajes', 'Acceso a administradores', 'Editar perfil de usuario', and 'Búsqueda de objetos' (highlighted). The main content area is titled 'Búsqueda de objetos' and includes a search bar, a 'Cerrar sesión' button, and a list of filters: 'Filósofos', 'Periodos', 'Fotografías', 'Aportes', and 'Evidencias'. Each filter has a blue toggle switch.

Figura. Panel de administración – Perfiles Administrados.

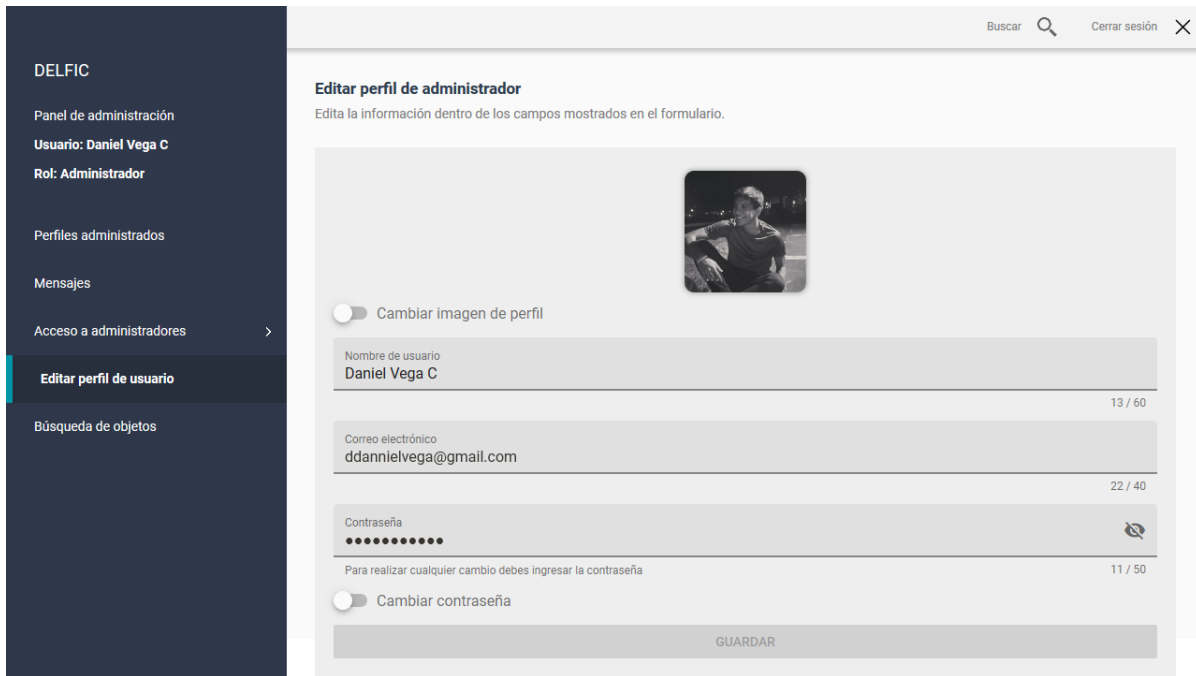


Figura. Panel de administración – Editar perfil de usuario.

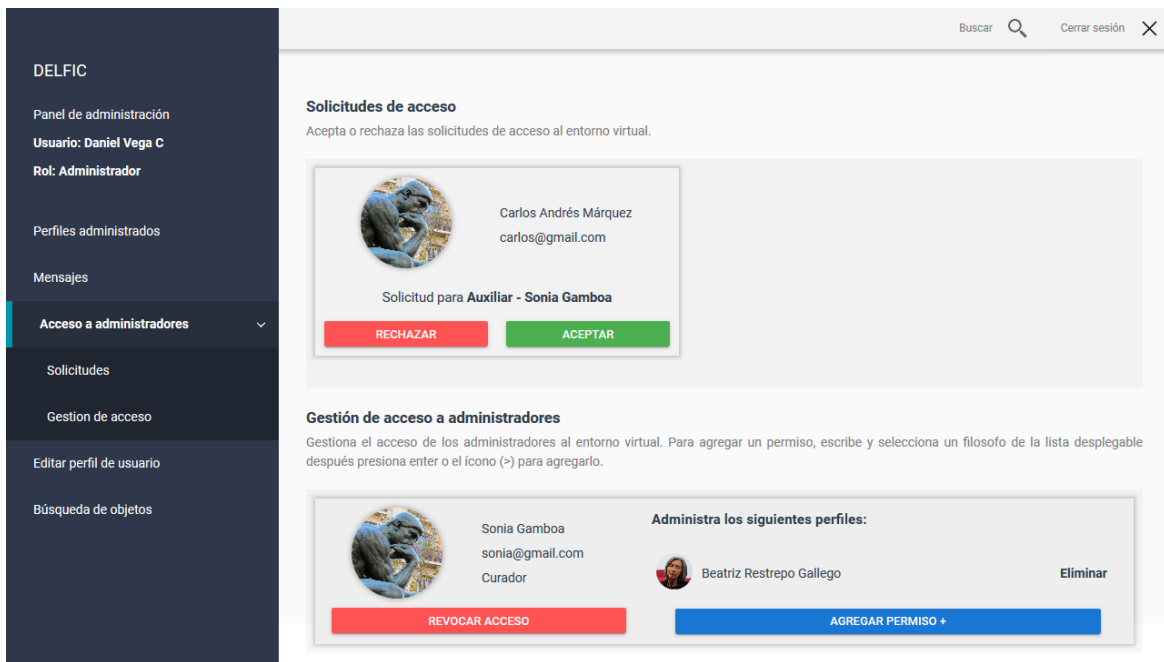


Figura. Panel de administración – Acceso a administradores.

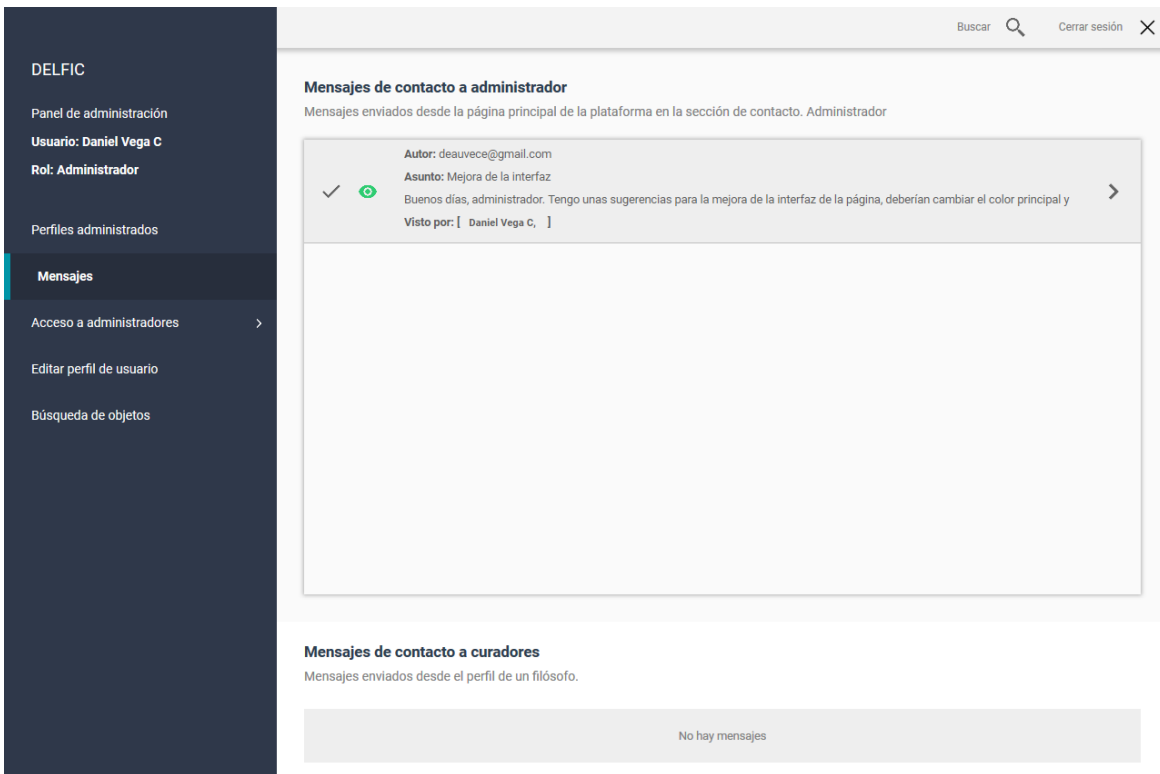


Figura. Panel de administración – Mensajes.

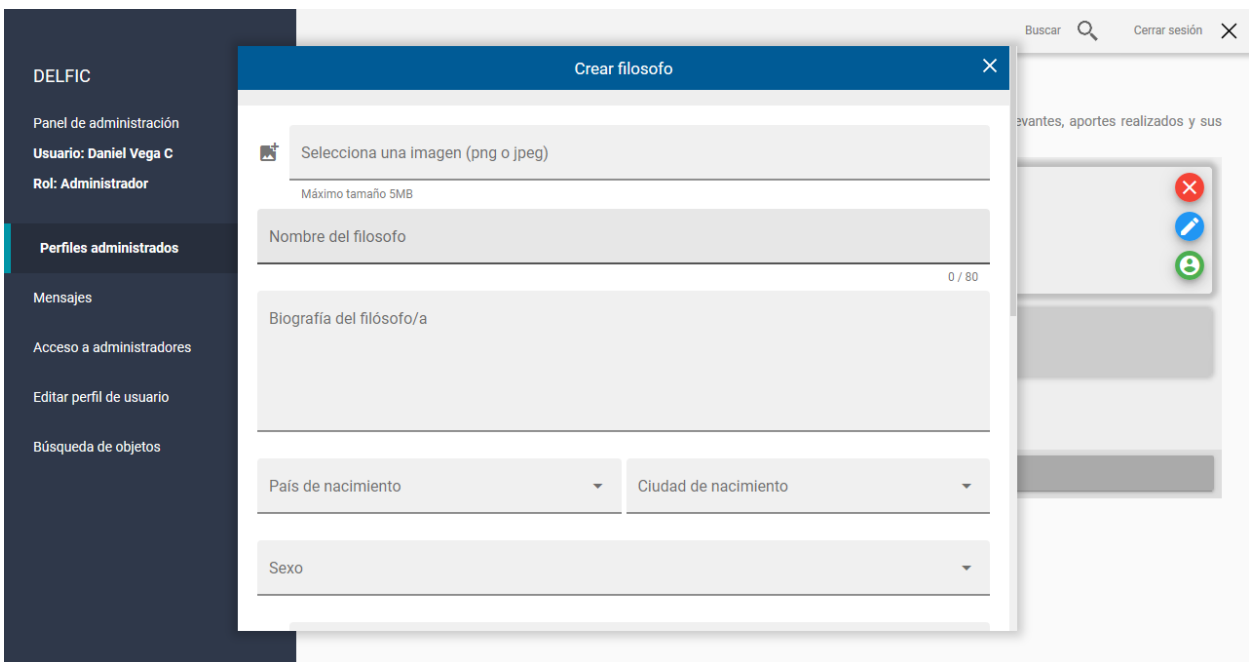


Figura. Panel de administración – Perfiles Administrados – Crear Filósofo.

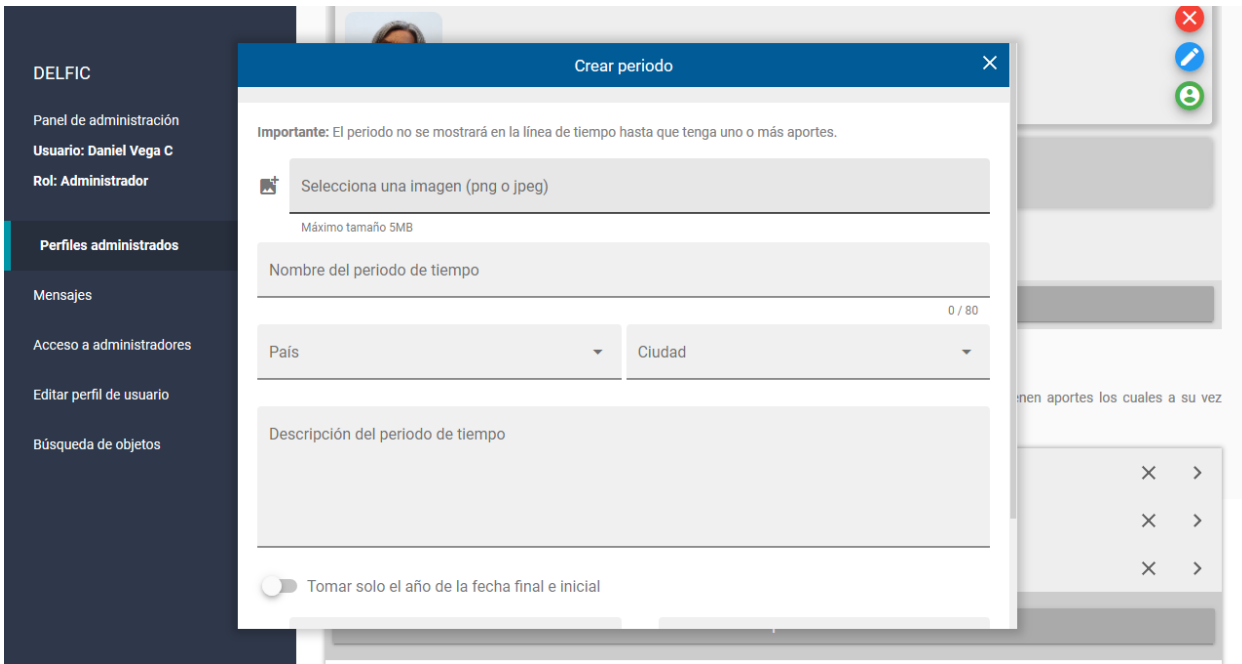


Figura. Panel de administración – Perfiles Administrados – Crear Periodo.

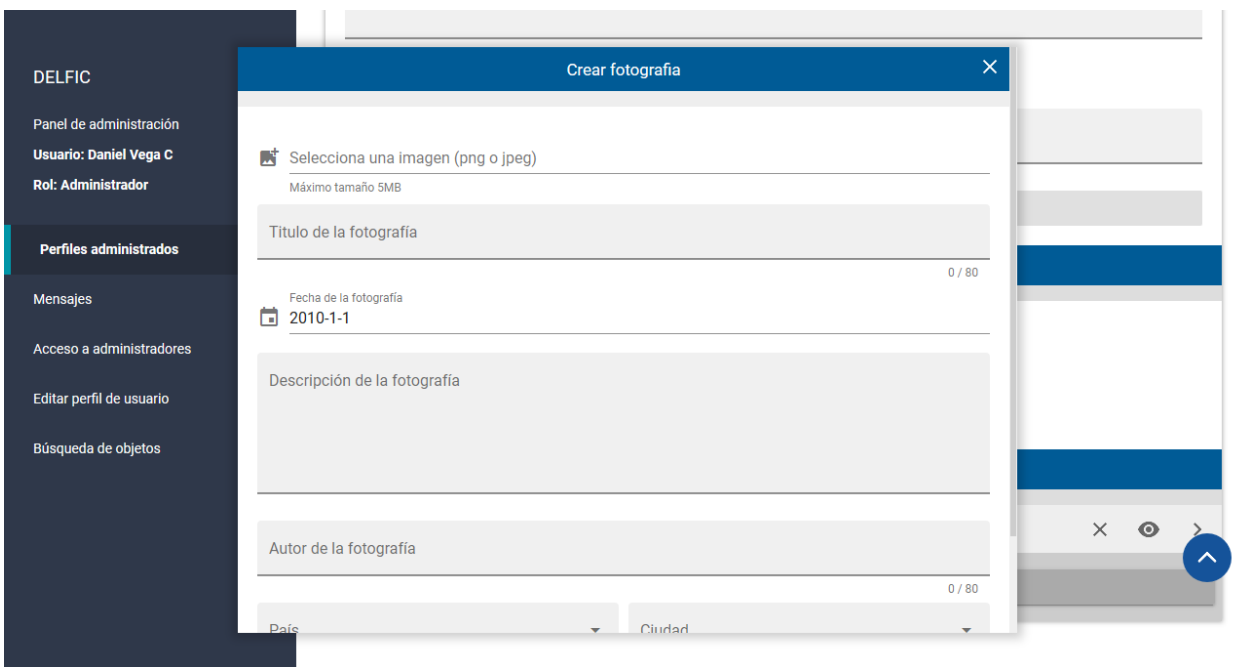


Figura. Panel de administración – Perfiles Administrados – Crear Fotografía.

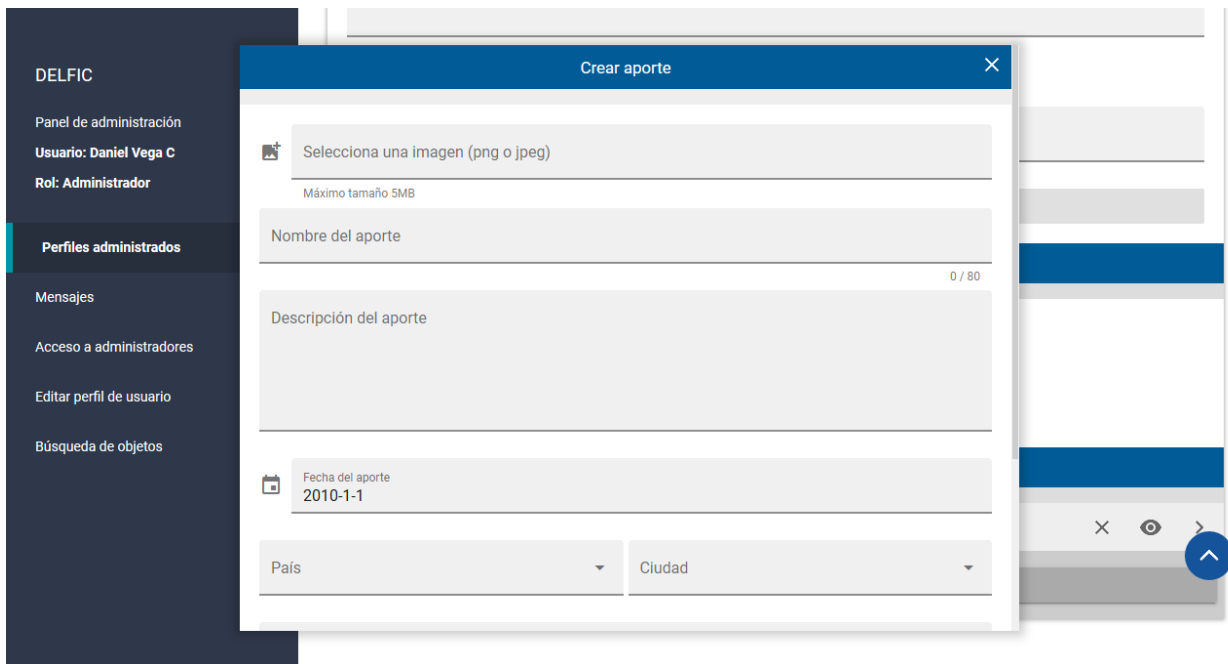


Figura. Panel de administración – Perfiles Administrados – Crear Aporte.

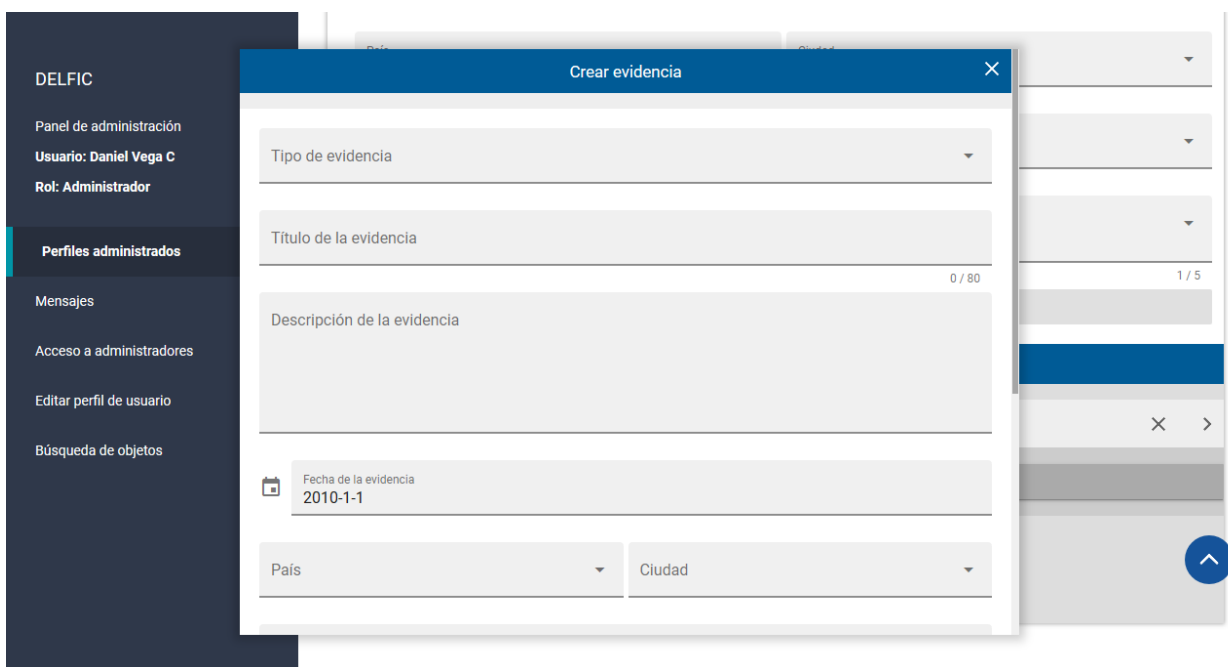


Figura. Panel de administración – Perfiles Administrados – Crear Evidencia.

Buscar 🔍
Cerrar sesión ✕

DELFIC

Panel de administración
 Usuario: Daniel Vega C
 Rol: Administrador

Perfiles administrados

Mensajes


Acceso a administradores >

Editar perfil de usuario

Búsqueda de objetos


Filósofos

Selecciona uno de los filósofos para editar su información y sus objetos relacionados (Periodos de tiempo relevantes, aportes realizados y sus respectivas evidencias).



Beatriz Restrepo Gallego

✕
✎
👤



Daniel Herrera Restrepo

+

Periodos de tiempo | Beatriz Restrepo Gallego

Los periodos de tiempo conforman la línea de tiempo mostrada en el perfil del filósofo estos periodos contienen aportes los cuales a su vez contienen evidencias. Selecciona un periodo de tiempo para editar su información.

< Estudios universitarios en USA y de maestría en Madrid, UCM

Información básica

Importante: El periodo no se mostrará en la línea de tiempo hasta que tenga uno o más aportes.



Cambiar imagen del periodo de tiempo

Nombre del periodo de tiempo
 Estudios universitarios en USA y de maestría en Madrid, UCM 59 / 80

País
 Estados Unidos

Ciudad
 New York

Descripción del periodo de tiempo
 Estudios universitarios en USA y de maestría en Madrid, UCM

Tomar solo el año de la fecha final e inicial

Fecha inicial del periodo
 1998-05-04

Fecha final del periodo
 2002-09-09

GUARDAR

Fotografías



+

Aportes del periodo de tiempo

Proyecto de grado ✕ 👁 >

+

Figura. Panel de administración – Perfiles Administrados – Editar Periodo.

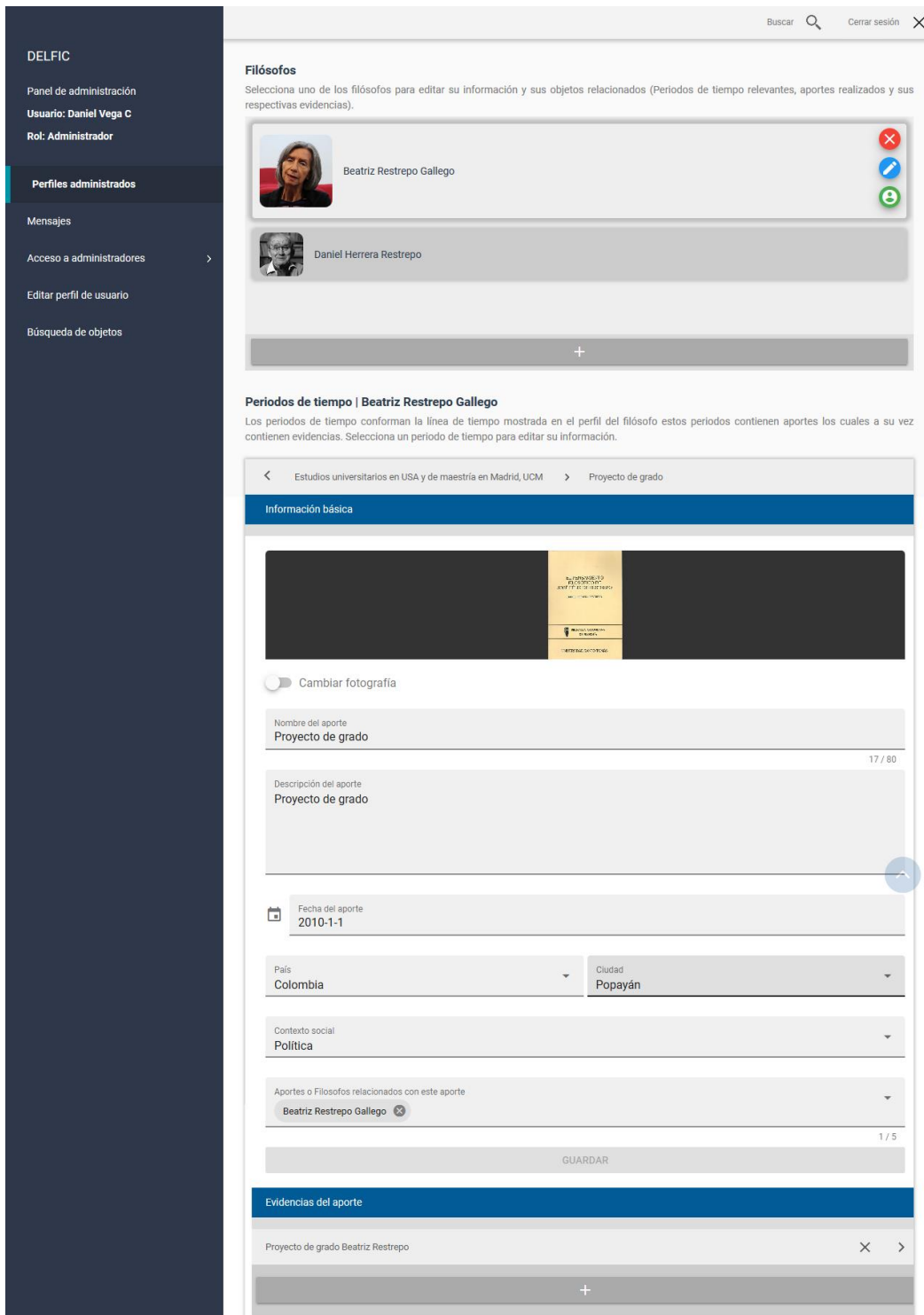


Figura. Panel de administración – Perfiles Administrados – Editar Aporte.

DELFC
Buscar Cerrar sesión

Panel de administración
Usuario: Daniel Vega C
Rol: Administrador

Perfiles administrados

Mensajes

Acceso a administradores >

Editar perfil de usuario

Búsqueda de objetos

Filósofos

Selecciona uno de los filósofos para editar su información y sus objetos relacionados (Periodos de tiempo relevantes, aportes realizados y sus respectivas evidencias).

Beatriz Restrepo Gallego

✕
✎
➕

Daniel Herrera Restrepo

+

Periodos de tiempo | Beatriz Restrepo Gallego

Los periodos de tiempo conforman la línea de tiempo mostrada en el perfil del filósofo estos periodos contienen aportes los cuales a su vez contienen evidencias. Selecciona un periodo de tiempo para editar su información.

< Estudios universitarios en USA y de maestría en Madrid, UCM > Proyecto de grado > Proyecto de grado Beatriz Restrepo

Tipo de evidencia
Documento

documentoProyecto de grado
Beatriz Restrepo2001-05-021.pdf

Cambiar documento

Título de la evidencia
Proyecto de grado Beatriz Restrepo 34 / 80

Descripción de la evidencia
Proyecto de grado Beatriz Restrepo

Fecha de la evidencia
2001-05-02 ↑

País
Colombia

Ciudad
Bello

Autor de la evidencia
Beatriz Restrepo Gallego 24 / 80

Idioma
Inglés

Personas que participan en esta evidencia
Beatriz Restrepo Gallego ✕

Filósofos o participantes externos (Que no se encuentran en la plataforma) que participan en esta evidencia 1 / 10

AGREGAR PARTICIPANTE EXTERNO

GUARDAR

Figura. Panel de administración – Perfiles Administrados – Editar Evidencia.

The image shows a web application interface for 'DELFIC'. On the left is a dark sidebar with navigation options: 'Panel de administración', 'Usuario: Daniel Vega C', 'Rol: Administrador', 'Perfiles administrados' (highlighted), 'Mensajes', 'Acceso a administradores', 'Editar perfil de usuario', and 'Búsqueda de objetos'. The main content area is titled 'Filósofos' and contains a list of philosopher profiles. The first profile is for 'Beatriz Restrepo Gallego'. A modal dialog is open over this profile, with the text: '¿Está seguro de eliminar el filósofo? Toda su información relacionada también se eliminará.' Below the text are two buttons: a green 'Eliminar' button and a red 'Cancelar' button. Below the list of profiles is a '+' sign. At the bottom, there is a section for 'Periodos de tiempo | Beatriz Restrepo Gallego' with a description and a list of time periods, including 'Estudios universitarios en USA y de maestría en Madrid, UCM'.

Figura. Panel de administración – Perfiles Administrados – Eliminar Objeto

Apéndice D. Casos de prueba.

Tabla. Caso de prueba - Inicio y cierre de sesión

Caso de prueba	Descripción	Datos de entrada	Respuesta	Resultado
Inicio y cierre de sesión	Inicio de sesión con los datos completos	Correo: ddanielvega@gmail.com Contraseña: ddanielvega123	Inicio de sesión válido	Verifica correctamente los datos y devuelve un token de autenticación
	Inicio de sesión con los datos incompletos	Correo: - Contraseña: -	Correo electrónico requerido y contraseña requerida	Ok
	Inicio de sesión con los datos incorrectos	Correo: ddanielvega@gmail.com Contraseña: daniel123	El correo o contraseña son incorrectos	Ok
	Cerrar sesión en cualquier barra de navegación	-	Cierra sesión	Cierra sesión y elimina el token de autenticación

Tabla. Caso de prueba - Registro de usuario

Caso de prueba	Descripción	Datos de entrada	Respuesta	Resultado
Registro de usuario	Registro como administrador con los datos completos	Nombres: Santiago Gómez Correo: santiagogomez@gmail.com Contraseña: santiagogomez123 TipoAdmin: Administrador	Se ha enviado la solicitud de registro	Se ha agregado el usuario a la base de datos y se creó una solicitud
	Registro como administrador con los datos incompletos	Nombres: - Correo: - Contraseña: - TipoAdmin: -	Los campos nombres, correo, contraseña y tipo Admin son requeridos	Ok
	Registro como curador con los datos completos	Nombres: Valentina Rojas Correo: valentinarojas@gmail.com Contraseña: valentinarojas123 TipoAdmin: Curador	Se ha enviado la solicitud de registro	Se ha agregado el usuario a la base de datos y se creó una solicitud
	Registro como curador con los datos incompletos	Nombres: - Correo: - Contraseña: - TipoAdmin: -	Los campos nombres, correo, contraseña y tipo Admin son requeridos	Ok

Registro como auxiliar con los datos completos	Nombres: Daniela Mendoza Correo: danielamendoza@gmail.com Contraseña: danielamendoza123 TipoAdmin: Auxiliar Curador: Valentina Rojas	Se ha enviado la solicitud de registro	Se ha agregado el usuario a la base de datos y se creó una solicitud
Registro como administrador con los datos incompletos	Nombres: - Correo: - Contraseña: - TipoAdmin: -	Los campos nombres, correo, contraseña y tipo Admin son requeridos	Ok

Tabla. Caso de prueba - Recuperar contraseña

Caso de prueba	Descripción	Datos de entrada	Respuesta	Resultado
Recuperar contraseña	Recuperar contraseña con un correo electrónico válido	Correo: ddanielvega@gmail.com	Si existe el correo en la base de datos se enviará la información necesaria para recuperar la cuenta.	Se envió un correo al usuario por medio del correo de la plataforma 'delficcol@gmail.com'
	Inicio de sesión con los datos incompletos	Correo: -	El correo es requerido	Ok

Tabla. Caso de prueba - Contactar administrador

Caso de prueba	Descripción	Datos de entrada	Respuesta	Resultado
	Contactar administrador de la plataforma en la sección de contacto de la página principal con los datos completos	Correo: deauvece@gmail.com Asunto: Prueba Contenido: Contenido de prueba	Mensaje correctamente enviado	Se agregó el mensaje a la base de datos
Contactar administrador	Contactar administrador de la plataforma en la sección de contacto de la página principal con un correo no válido	Correo: deauvece@invalido.com Asunto: Prueba Contenido: Contenido de prueba	Correo electrónico inválido	Ok
	Contactar administrador de la plataforma en la sección de contacto de la página principal con los datos incompletos	Correo: - Asunto: - Contenido: -	Los campos correo, asunto y contenido son requeridos	Ok
	Contactar curador de un filósofo en su perfil con los datos completos	Correo: deauvece@gmail.com	Mensaje correctamente enviado	Se agregó el mensaje a la base de datos

	Contenido: Contenido de prueba		
Contactar curador de un filósofo en su perfil con un correo no válido	Correo: deauvece@adasdf.com Contenido: Contenido de prueba	de	Correo electrónico inválido Ok
Contactar curador de un filósofo en su perfil con los datos incompletos	Correo: - Contenido: -		Los campos correo y contenido son requeridos Ok

Tabla. Caso de prueba - Recuperar contraseña

Caso de prueba	Descripción	Datos de entrada	Respuesta	Resultado
Búsqueda	Realizar búsqueda de filósofo con los datos completos	Query: Beatriz	Aparece Beatriz Restrepo Gallego en los resultados	Ok
	Realizar búsqueda de filósofo con los datos incorrectos	Query: Bea	Debe tener al menos 4 caracteres	Ok
	Realizar búsqueda de aporte con los datos completos	Query: Montaje Contexto: Todos	Aparece Montaje de la muestra de cerámica Quimbaya, Museo de antropología UdeA	Ok
	Realizar búsqueda de aporte sin seleccionar contexto	Query: Montaje Contexto: -	Se queda en blanco	Aparecen errores en la consola, se envía la búsqueda sin seleccionar un contexto
	Realizar búsqueda de aporte con los datos incompletos	Query: - Contexto: -	Escribe algo en el campo de búsqueda	Ok

Tabla. Caso de prueba - Filtro línea de tiempo

Caso de prueba	Descripción	Datos de entrada	Respuesta	Resultado
Filtro línea de tiempo	Filtrar los aportes por año	Año: 2010	Sólo se muestran los aportes correspondientes al año 2010 y sus respectivos periodos de tiempo	Ok
	Filtrar los aportes por contexto	Contexto: Educación	Sólo se muestran los aportes correspondientes al contexto Educación y sus respectivos periodos de tiempo	Ok
	Filtrar los aportes por país	País: Colombia	Sólo se muestran los aportes realizados en el país Colombia y sus respectivos periodos de tiempo	Ok
	Filtrar los aportes por ciudad	Ciudad: Medellín	Sólo se muestran los aportes realizados en la ciudad Medellín y sus respectivos periodos de tiempo	Ok

Tabla. Caso de prueba - Búsqueda de evidencias

Caso de prueba	Descripción	Datos de entrada	Respuesta	Resultado
Búsqueda de evidencias	Realizar búsqueda de evidencia con los datos completos	Query: creación Tipo: Video Año: 1977 Idioma: Español	Aparece Creación de los primeros programas de Filosofía e Historia en la UdeA	Ok
	Realizar búsqueda de evidencia con los datos incorrectos	Query: cre Tipo: Video Año: 1977 Idioma: Español	Debe tener al menos 4 caracteres	Ok

Tabla. Caso de prueba - Búsqueda de objetos en el panel de administración

Caso de prueba	Descripción	Datos de entrada	Respuesta	Resultado
Búsqueda de objetos en el panel de administración	Realizar búsqueda de evidencia con los datos incorrectos	Query: bea	Debe tener al menos 4 caracteres	Ok
	Realizar búsqueda de evidencia con los datos correctos	Query: beatriz	Se muestran los objetos que contienen la palabra beatriz en el nombre o título	Ok
	Se filtran los resultados por filósofos	Query: beatriz Filtro: filósofos	Se muestran los filósofos que contienen la palabra beatriz en el nombre	Ok
	Se filtran los resultados por periodos	Query: beatriz Filtro: periodos	Se muestran los periodos que contienen la palabra beatriz en el título	Ok
	Se filtran los resultados por fotografías	Query: beatriz Filtro: fotografías	Se muestran las fotografías que contienen la palabra beatriz en el título	Ok
	Se filtran los resultados por aportes	Query: beatriz Filtro: aportes	Se muestran los aportes que contienen la palabra beatriz en el nombre	Ok
	Se filtran los resultados por evidencias	Query: beatriz Filtro: evidencias	Se muestran las evidencias que contienen la palabra beatriz en el título	Ok

Tabla. Caso de prueba – Editar perfil de usuario

Caso de prueba	Descripción	Datos de entrada	Respuesta	Resultado
Editar perfil de usuario	Editar el perfil con los datos correctos	Cambiar imagen: Archivo de imagen Nombres: Editar Daniel Correo: deauvece@gmail.com Contraseña: ddanielvega123 Nueva contraseña: daniel123	Se han guardado los cambios.	Se cambiaron los datos correspondientes en la base de datos.

	Repetir nueva contraseña: daniel123	
Editar el perfil con los datos incorrectos	Cambiar imagen: - Nombres: - Correo: - Contraseña: - Nueva contraseña: - Repetir nueva contraseña: -	Se muestra el mensaje 'Este campo es requerido' para cada uno de los campos que hacen falta Ok

Tabla. Caso de prueba – Resolver solicitud de acceso

Caso de prueba	Descripción	Datos de entrada	Respuesta	Resultado
Resolver solicitud de acceso	Se selecciona la opción para aceptar la solicitud de acceso	Seleccionar botón	Se ha aceptado la solicitud	Se cambió el estado de la solicitud y el permiso de acceso del usuario. Pero falta agregar el usuario a la sección de gestión de acceso a administradores
	Se selecciona la opción para rechazar la solicitud de acceso	Seleccionar botón	Se ha rechazado la solicitud	Se cambió el estado de la solicitud y se eliminó al usuario

Tabla. Caso de prueba – Gestionar acceso a usuario

Caso de prueba	Descripción	Datos de entrada	Respuesta	Resultado
Gestionar acceso a usuario	Se selecciona la opción para revocar acceso a uno de los usuarios de la lista	Seleccionar botón	Se ha revocado el acceso al usuario seleccionado	Se cambió el permiso de acceso del usuario en la base de datos
	Se selecciona la opción para garantizar acceso a uno de los usuarios de la lista	Seleccionar botón	Se ha garantizado el acceso al usuario seleccionado	Se cambió el permiso de acceso del usuario en la base de datos

Tabla. Caso de prueba – Gestionar permiso sobre un filósofo a un usuario

Caso de prueba	Descripción	Datos de entrada	Respuesta	Resultado	
Gestionar permiso sobre un filósofo a un usuario	Agregar un permiso sobre un filósofo a un usuario buscándolo en la lista de gestión de acceso y buscando el filósofo en el campo de búsqueda mostrado	Daniel Restrepo	Herrera	Permiso agregado correctamente	Ok
	Se selecciona la opción para garantizar acceso a uno de los usuarios de la lista	Seleccionar botón al lado del filósofo		Permiso eliminado correctamente	Ok

Tabla. Caso de prueba – Responder mensaje de contacto

Caso de prueba	Descripción	Datos de entrada	Respuesta	Resultado
Responder mensaje de contacto	Responder correctamente uno de los mensajes mostrados en la lista de la sección de mensajes, es igual para los dos tipos de mensajes (administrador y curador)	Respuesta: Respuesta prueba	de Se ha enviado la respuesta	Se envió la respuesta por medio del correo 'delficcol@gmail.com', se cambió el estado del mensaje a respondido y se agregó la respuesta al mensaje
	Responder incorrectamente uno de los mensajes mostrados en la lista de la sección de mensajes, es igual para los dos tipos de mensajes (administrador y curador)	Respuesta: re	Debe tener al menos 4 caracteres	Ok

Tabla. Caso de prueba – Gestionar filósofo

Caso de prueba	Descripción	Datos de entrada	Respuesta	Resultado
Gestionar filósofo	Eliminar uno de los filósofos mostrados en la sección de perfiles administrados en el panel de administración, también se encuentra en la sección búsqueda de objetos	Seleccionar botón	Se ha eliminado el filósofo y su información correspondiente	Se eliminó el filósofo y todos los objetos relacionados en la base de datos
	Editar uno de los filósofos mostrados en la sección de perfiles administrados en el panel de administración, también se encuentra en la sección búsqueda de objetos	Nombres: Nombre de prueba Biografía: Biografía de prueba País: Colombia Ciudad: Bucaramanga FechaNacimiento: 1950 FechaFallecimiento: 2000 Nacionalidad: Colombia Tema: 2 Imagen: Archivo de imagen	Se han guardado los cambios	Se actualizó la información correspondiente en la base de datos y en el frontend.
	Editar uno de los filósofos mostrados en la sección de perfiles administrados en el panel de administración con los datos incorrectos	Nombres: - Biografía: - País: - Ciudad: - FechaNacimiento: 1951-1-1 FechaFallecimiento: 1951-1-1 Nacionalidad: - Tema: 1 Imagen: -	Se muestra el mensaje 'Campo requerido' para cada uno de los campos incompletos	Ok
	Agregar un nuevo filósofo	Nombres: Nombre de prueba 2 Biografía: Biografía de prueba 2 País: Argentina Ciudad: Buenos Aires FechaNacimiento: 1980	Se han guardado los cambios	Se agregó el filósofo a la base de datos y se actualizó la información en el frontend.

	FechaFallecimiento: - Nacionalidad: Colombia Tema: 3 Imagen: Archivo de imagen	
Agregar un filósofo con los datos incorrectos	Nombres: - Biografía: - País: - Ciudad: - FechaNacimiento: 1951-1-1 FechaFallecimiento: 1951-1-1 Nacionalidad: - Tema: 1 Imagen: -	Se muestra el mensaje 'Campo requerido' para cada uno de los campos incompletos Ok
Ver el perfil de uno de los filósofos mostrados en el panel de administración	Respuesta: Re	Debe tener al menos 4 caracteres Ok

Tabla. Caso de prueba – Gestionar periodo

Caso de prueba	Descripción	Datos de entrada	Respuesta	Resultado
	Eliminar uno de los periodos correspondientes a un filósofo, también se encuentra en la sección búsqueda de objetos	Seleccionar botón	Se ha eliminado el periodo y su información correspondiente	Se eliminó el periodo y todos los objetos relacionados en la base de datos
	Editar uno de los periodos correspondientes a un filósofo, también se encuentra en la sección búsqueda de objetos	Nombre: Nombre de prueba Descripción: Descripción de prueba País: Colombia Ciudad: Bucaramanga FechaInicial: 2010-4-5 FechaFinal: 2010-4-5 Tomar solo año: Selected Imagen: Archivo de imagen	Se han guardado los cambios	Se actualizó la información correspondiente en la base de datos y en el frontend.
Gestionar periodo	Editar uno de los periodos correspondientes a un filósofo con los datos incorrectos	Nombre: - Descripción: - País: - Ciudad: - FechaInicial: 2010-4-5 FechaFinal: 2010-4-5 Tomar solo año: Unselected Imagen: -	Se muestra el mensaje 'Campo requerido' para cada uno de los campos incompletos	Ok
	Agregar un nuevo periodo en un filósofo con los datos correctos	Nombre: Nombre de prueba 2 Descripción: Descripción de prueba 2 País: Colombia Ciudad: Bucaramanga FechaInicial: 2010-4-5 FechaFinal: 2010-4-5 Tomar solo año: Selected Imagen: Archivo de imagen	Se han guardado los cambios	Se agregó el periodo a la base de datos y se actualizó la información en el frontend.

Agregar un periodo con los datos incorrectos	Nombre: - Descripción: - País: - Ciudad: - FechaInicial: 2010-4-5 FechaFinal: 2010-4-5 Tomar solo año: Unselected Imagen: -	Se muestra el mensaje ‘Campo requerido’ para cada uno de los campos incompletos	Ok
--	--	---	----

Tabla. Caso de prueba – Gestionar fotografía

Caso de prueba	Descripción	Datos de entrada	Respuesta	Resultado
	Eliminar una fotografía correspondiente a un periodo de tiempo, también se encuentra en la sección búsqueda de objetos	Seleccionar botón	Se ha eliminado la fotografía	Se eliminó la fotografía en la base de datos
	Editar una fotografía correspondiente a un periodo, también se encuentra en la sección búsqueda de objetos	Título: Fotografía de prueba Descripción: Descripción de la fotografía Autor: Autor de prueba País: Colombia Ciudad: Bucaramanga Fecha: 2015-7-1 Imagen: Archivo de imagen	Se han guardado los cambios	Se actualizó la información correspondiente en la base de datos y en el frontend.
Gestionar fotografía	Editar una fotografía correspondiente a un periodo con los datos incorrectos	Título: - Descripción: - Autor: - País: - Ciudad: - Fecha: 1951-1-1 Imagen: -	Se muestra el mensaje ‘Campo requerido’ para cada uno de los campos incompletos	Ok
	Agregar una fotografía para un periodo de tiempo con los datos correctos	Título: Fotografía de prueba 2 Descripción: Descripción de la fotografía 2 Autor: Autor de prueba 2 País: Colombia Ciudad: Bucaramanga Fecha: 2015-7-1 Imagen: Archivo de imagen	Se han guardado los cambios	Se agregó la fotografía a la base de datos y se actualizó la información en el frontend.
	Agregar una fotografía para un periodo de tiempo con los datos incorrectos	Título: - Descripción: - Autor: - País: - Ciudad: - Fecha: 1951-1-1 Imagen: -	Se muestra el mensaje ‘Campo requerido’ para cada uno de los campos incompletos	Ok

Tabla. Caso de prueba – Gestionar aporte

Caso de prueba	Descripción	Datos de entrada	Respuesta	Resultado
	Eliminar un aporte correspondiente a un periodo de tiempo, también se encuentra en la sección búsqueda de objetos	Seleccionar botón	Se ha eliminado el aporte	Se eliminó el aporte y sus evidencias en la base de datos
	Editar un aporte correspondiente a un periodo, también se encuentra en la sección búsqueda de objetos	Nombre: Aporte de prueba Descripción: Descripción del aporte Fecha: 2015-7-1 Contexto: Salud País: Colombia Ciudad: Bucaramanga Participantes: Beatriz Restrepo Gallego Imagen: Archivo de imagen	Se guardado los cambios	Se actualizó la información correspondiente en la base de datos y en el frontend.
Gestionar aporte	Editar un aporte correspondiente a un periodo con los datos incorrectos	Nombre: - Descripción: - Fecha: 1951-7-1 Contexto: - País: - Ciudad: - Participantes: - Imagen: -	Se muestra el mensaje 'Campo requerido' para cada uno de los campos incompletos	Ok
	Agregar un aporte para un periodo de tiempo con los datos correctos	Nombre: Aporte de prueba 2 Descripción: Descripción del aporte 2 Fecha: 2015-7-1 Contexto: Salud País: Colombia Ciudad: Bucaramanga Participantes: Beatriz Restrepo Gallego Imagen: Archivo de imagen	Se guardado los cambios	Se agregó el aporte a la base de datos y se actualizó la información en el frontend.
	Agregar un aporte para un periodo de tiempo con los datos incorrectos	Nombre: - Descripción: - Fecha: 1951-7-1 Contexto: - País: - Ciudad: - Participantes: - Imagen: -	Se muestra el mensaje 'Campo requerido' para cada uno de los campos incompletos	Ok

Tabla. Caso de prueba – Gestionar evidencia

Caso de prueba	Descripción	Datos de entrada	Respuesta	Resultado
Gestionar evidencia	Eliminar una evidencia correspondiente a un aporte, también se encuentra en la sección búsqueda de objetos	Seleccionar botón	Se ha eliminado la evidencia	Se eliminó la evidencia en la base de datos

<p>Editar una evidencia de imagen correspondiente a un aporte, también se encuentra en la sección búsqueda de objetos</p>	<p>TipoEvidencia: Imagen Imagen: Archivo de imagen Título: Evidencia imagen Descripción: Descripción de la evidencia Fecha: 2015-7-1 País: Colombia Ciudad: Bucaramanga Idioma: Inglés Participantes: Beatriz Restrepo Gallego</p>	<p>Se guardado han los cambios</p>	<p>Se actualizó la información correspondiente en la base de datos y en el frontend.</p>
<p>Editar una evidencia de documento correspondiente a un aporte, también se encuentra en la sección búsqueda de objetos</p>	<p>TipoEvidencia: Documento Imagen: Archivo PDF Título: Evidencia documento Descripción: Descripción de la evidencia Fecha: 2015-7-1 País: Colombia Ciudad: Bucaramanga Idioma: Inglés Participantes: Beatriz Restrepo Gallego</p>	<p>Se guardado han los cambios</p>	<p>Se actualizó la información correspondiente en la base de datos y en el frontend.</p>
<p>Editar una evidencia de video correspondiente a un aporte, también se encuentra en la sección búsqueda de objetos</p>	<p>TipoEvidencia: Video Imagen: Enlace a video de YouTube Título: Evidencia video Descripción: Descripción de la evidencia Fecha: 2015-7-1 País: Colombia Ciudad: Bucaramanga Idioma: Inglés Participantes: Beatriz Restrepo Gallego</p>	<p>Se guardado han los cambios</p>	<p>Se actualizó la información correspondiente en la base de datos y en el frontend.</p>
<p>Editar una evidencia correspondiente a un aporte con los datos incorrectos</p>	<p>TipoEvidencia: - Título: - Descripción: - Fecha: 1951-1-1 País: - Ciudad: - Idioma: - Participantes: -</p>	<p>Se muestra el mensaje 'Campo requerido' para cada uno de los campos incompletos</p>	<p>Ok</p>
<p>Agregar una evidencia para un aporte con los datos correctos</p>	<p>TipoEvidencia: Video Imagen: Enlace a video de YouTube 2 Título: Evidencia video Descripción: Descripción de la evidencia 2 Fecha: 2015-7-1 País: Colombia Ciudad: Bucaramanga Idioma: Inglés Participantes: Beatriz Restrepo Gallego</p>	<p>Se guardado han los cambios</p>	<p>Se agregó la evidencia a la base de datos y se actualizó la información en el frontend.</p>

Agregar una evidencia para un aporte con los datos incorrectos

TipoEvidencia: -
Título: -
Descripción: -
Fecha: 1951-1-1
País: -
Ciudad: -
Idioma: -
Participantes: -

Se muestra el mensaje 'Campo requerido' para Ok cada uno de los campos incompletos
