



CARTA DE AUTORIZACIÓN

CÓDIGO

AP-BIB-FO-06

VERSIÓN

1

VIGENCIA

2014

PÁGINA

1 de 1

Neiva, 24 de febrero del 2020

Señores

CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD SURCOLOMBIANA

Ciudad

El (Los) suscrito(s):

Jesús Santiago Tovar Cuenca, con C.C. No. 1082215737, Jhorkman Dariel Bernal Herrera, con C.C. No. 1079607488, Autor(es) de la tesis y/o trabajo de grado titulado Sistema de recomendador de ofertas educativas basado en la clasificación de perfiles cognitivos presentado y aprobado en el año 2020 como requisito para optar al título de Ingeniero de software; Autorizo (amos) al CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN de la Universidad Surcolombiana para que, con fines académicos, muestre al país y el exterior la producción intelectual de la Universidad Surcolombiana, a través de la visibilidad de su contenido de la siguiente manera:

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

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

EL AUTOR/ESTUDIANTE: Jesús Santiago Tovar

EL AUTOR/ESTUDIANTE: Jhorkman Bernal Herrera

Firma:

Firma:

Vigilada Mineducación



TÍTULO COMPLETO DEL TRABAJO: Sistema recomendador de ofertas educativas basado en la clasificación de perfiles cognitivos

AUTOR O AUTORES:

Primero y Segundo Apellido	Primero y Segundo Nombre
Tovar Cuenca	Jesús Santiago
Bernal Herrera	Jhorkman Dariel

DIRECTOR Y CODIRECTOR TESIS:

Primero y Segundo Apellido	Primero y Segundo Nombre
Triana Tacuma	Eurípides

ASESOR (ES):

Primero y Segundo Apellido	Primero y Segundo Nombre
Castro Silva	Juan Antonio
Medina Rojas	Ferley

PARA OPTAR AL TÍTULO DE: Ingeniero de Software

FACULTAD: Ingeniería

PROGRAMA O POSGRADO:

CIUDAD: Neiva **AÑO DE PRESENTACIÓN:** 2020 **NÚMERO DE PÁGINAS:** 97

TIPO DE ILUSTRACIONES (Marcar con una X):

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

Vigilada Mineducación

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



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

MATERIAL ANEXO:

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

PALABRAS CLAVES EN ESPAÑOL E INGLÉS:

<u>Español</u>	<u>Inglés</u>	<u>Español</u>	<u>Inglés</u>
1. Test de inteligencias	Intelligence test	6. _____	_____
2. Recomendaciones	Recommendations	7. _____	_____
3. Ofertas educativas	Educational offers	8. _____	_____
4. Microservicios	Microservices	9. _____	_____
5. Programa	Software	10. _____	_____

RESUMEN DEL CONTENIDO: (Máximo 250 palabras)

Se desarrolló un ecosistema de aplicaciones de múltiples oferentes, en la que los usuarios puedan buscar y filtrar ofertas educativas ofrecidas por diversas entidades o instructores, donde los estudiantes obtengan facilidad en la búsqueda con un banco de ofertas para pregrado, postgrados, tecnologías, técnicos, curso, entre otros. Brindando además un test cognitivo que ayuda a las personas a determinar con sus cualidades que le es más fácil estudiar, retornando múltiples ofertas educativas según su test de guía vocacional, apoyado también de un sistema de recomendación ya que son herramientas que generan sugerencias sobre un determinado objeto de estudio, a partir de las preferencias y opiniones dadas por otros usuarios o por su mismo contenido. Implementando esta propuesta, se pretende disminuir la tasa de deserción en la educación por parte de los estudiantes, permitiéndoles desarrollar al máximo sus habilidades académicas y que éstos estén preparados para afrontar las demandas en el mercado laboral.



ABSTRACT: (Máximo 250 palabras)

An ecosystem of multi-vendor applications was developed, in which users can search and filter educational offers offered by various entities or instructors, where students obtain ease in searching with a bank of offers for undergraduate, postgraduate, technologies, technicians, course, among others. Also providing a cognitive test that helps people determine with their qualities that it is easier to study, returning multiple educational offers according to their vocational guidance test, also supported by a recommendation system since they are tools that generate suggestions on a given object of study, based on the preferences and opinions given by other users or by the same content. Implementing this proposal is intended to reduce the dropout rate in education by students, allowing them to fully develop their academic skills and that they are prepared to front-facing the demands in the labor market.

APROBACION DE LA TESIS

Nombre Presidente Jurado: Fernando Rojas Rojas

Firma:

Nombre Jurado: Juan Antonio Castro Silva

Firma:

Nombre Jurado: Ferley Medina Rojas

Firma:



UNIVERSIDAD
SURCOLOMBIANA

FACULTAD DE INGENIERÍA

PROGRAMA INGENIERÍA DE SOFTWARE

LÍNEA DE INVESTIGACIÓN DE DESARROLLO DE SOFTWARE

PROYECTO DE GRADO:

**SISTEMA RECOMENDADOR DE OFERTAS
EDUCATIVAS BASADO EN LA CLASIFICACION DE
PERFILES COGNITIVOS**

Proyecto presentado por Jhorkman Dariel Bernal Herrera y Jesús Santiago Tovar
Cuenca para optar el título de Ingeniero de Software de la Universidad
Surcolombiana.

Director:

Eurípides Triana

Tabla de Contenido

INTRODUCCIÓN.....	11
JUSTIFICACIÓN	13
PROBLEMA	15
OBJETIVOS.....	16
OBJETIVO GENERAL.....	16
OBJETIVOS ESPECÍFICOS	16
ALCANCES Y LIMITACIONES.....	17
MARCO TEÓRICO.....	18
INTELIGENCIA	18
<i>Concepto de inteligencia.....</i>	<i>18</i>
INTELIGENCIAS MÚLTIPLES.....	18
<i>Inteligencia Lingüística.....</i>	<i>21</i>
<i>Inteligencia lógico-matemática.....</i>	<i>21</i>
<i>Inteligencia Musical.....</i>	<i>22</i>
<i>Inteligencia Visual-Espacial.....</i>	<i>23</i>
<i>Inteligencia corporal-cinestésica.....</i>	<i>23</i>
<i>Inteligencia Interpersonal.....</i>	<i>24</i>
<i>Inteligencia Intrapersonal.....</i>	<i>24</i>
<i>Inteligencia Naturalista.....</i>	<i>25</i>
SISTEMAS DE RECOMENDACIÓN	25
<i>Capturar la información.....</i>	<i>26</i>
<i>Filtrado basado en contenido.....</i>	<i>27</i>
<i>Filtrado colaborativo.....</i>	<i>29</i>

METODOLOGIA DE CASCADA	35
METODOLOGÍA AGILE.....	36
<i>Manifiesto Agile</i>	37
ESTADO DEL ARTE	39
APLICACIONES BASADAS EN LAS INTELIGENCIAS MÚLTIPLES DE HOWARD GARDNER.....	39
APLICACIÓN Y ENFOQUE SISTEMAS DE RECOMENDACIÓN	39
ESTADÍSTICAS SIMILARWEB - PAGINAS NACIONALES.....	40
<i>Performance del sitio Web.</i>	40
CONCLUSIONES DEL ESTADO DEL ARTE	¡ERROR! MARCADOR NO DEFINIDO.
METODOLOGÍA	43
DESARROLLO ÁGIL Y EN CASCADA	43
<i>Requisitos del sistema</i>	45
CLOUD	75
<i>Flexibilidad.</i>	75
<i>Recuperación de desastres</i>	75
<i>Actualizaciones automáticas de software</i>	75
<i>Gastos de capital</i>	75
<i>Mayor colaboración</i>	75
<i>Flexibilidad laboral.</i>	76
<i>Control de documentos</i>	76
<i>Seguridad.</i>	76
<i>Competitividad</i>	76
TECNOLOGÍAS UTILIZADAS.....	76
GESTIÓN DE TAREAS.....	48
DISEÑO.....	50
CASOS DE USO.....	50
MODELO DE BASE DATOS	56

DESCRIPCIÓN DE REQUISITOS FUNCIONALES 60

DISEÑO WEB 60

Paneles de administración..... 64

IMPLEMENTACIÓN..... **70**

 DESARROLLO DE SISTEMA DE RECOMENDACIÓN 70

Construcción de modelo simple de popularidad y filtrado colaborativo usando Turicreate 70

CRONOGRAMA **77**

 DETALLE ACTIVIDADES 78

PRESUPUESTO..... **82**

CONCLUSIONES..... **86**

APÉNDICE ¡ERROR! MARCADOR NO DEFINIDO.

REFERENCIAS **93**

Lista de Tablas

Tabla 1. Calificación de oferta por usuario	31
Tabla 2. Clasificación de ofertas por calificación	34
Tabla 3. Comparativa de implementación	44
Tabla 4. Tecnologías a utilizar	76
Tabla 5. Caso de uso para inicio de registro de usuario.....	45
Tabla 6. Caso de uso para inicio de sesión de usuario.....	45
Tabla 7. Caso de uso para registro de oferta.	46
Tabla 8. Caso de uso para publicación de ofertas	46
Tabla 9. Caso de uso para modificación de información personal o de las ofertas	46
Tabla 10 Principales actividades realizadas.....	77
Tabla 11. Presupuesto.....	82

Lista de Figuras

Figura 51. Ilustración de filtrado basado en contenido. Fuente Analyticsvidhya	27
Figura 52. Ejemplo de recomendación basada entre usuarios. Fuente Analyticsvidhya	30
Figura 13. Website Performance entre Estudialo.co y SenaSofia Fuente: SimilarWeb	40
Figura 14. Visitas totales Fuente: SimilarWeb.....	41
Figura 15. Porcentaje de trafico entrante, desktop vs. Móvil.....	41
Figura 16. Interés en el tiempo, línea roja SenaSofia, azul estudialo.co Fuente: Google	42
Figura 17. Desglose comparado por subregión.....	42
Figura 18. Waterfall model vs Agile. (Lotz, 2013)	44
Figura 19. Dashboard de proyectos de Asana. (Asana.com, 2017).....	49
Figura 20. Registro de usuarios.....	51
Figura 21. Diagrama UML: Login y Registro de instituciones educativas.....	51
Figura 22. Diagrama de estado: login.....	53
Figura 23. Diagrama UML: Búsqueda de ofertas educativas	54
Figura 24. Diagrama UML: DasBoard Admin.....	54
Figura 25. Diagrama UML: Dashboard estudiante	55
Figura 26. Diagrama UML: Dashboard instructor	55
Figura 27. Diagrama UML: test de inteligencias	56
Figura 28. Modelo relacional para la seguridad	57
Figura 29. Modelo relacional para la administración de las ofertas	58
Figura 30. Modelo relacional para inscripción de ofertas.....	58
Figura 31. ER Información usuarios e intereses.....	59
Figura 32. Página de inicio	60
Figura 33. Ofertas descartadas(pago)	61
Figura 34. Ofertas clasificadas según su la inteligencia predominante	61

Figura 35. Ofertas más vistas por los usuarios del sistema.....	62
Figura 36. Formulario de login del sistema.	62
Figura 37. Formulario para el registro de usuarios.....	63
Figura 38. Buscador de ofertas.....	63
Figura 39. Pie de pagina	64
Figura 40. Página de perfil de usuario	65
Figura 41. Gestión de ofertas	65
Figura 42. Administración de instituciones	66
Figura 43. Manejo de publicidad.....	66
Figura 44. Administración de usuarios	67
Figura 45. Formulario de datos personales.	67
Figura 46. Detalle de la oferta.....	68
Figura 47. Diseño test de inteligencias.	68
Figura 48. Fin test de inteligencias.....	69
Figura 49. Resultados test de inteligencias.	69
Figura 50. Estructura de despliegue de la aplicación	70

Lista de Ecuaciones

Ecuación 1	28
Ecuación 2	28
Ecuación 3	29
Ecuación 4	30
Ecuación 5	32
Ecuación 6	32
Ecuación 7	33
Ecuación 8	34
Ecuación 9	34
Ecuación 10.....	35

Lista de Códigos

Código 1. Archivo de entrenamiento del modelo de recomendación.....	71
Código 2. Objeto JSON que emite spring boot.....	71
Código 3. Configuración Flask para el microservicio de recomendación	72
Código 4. Objeto JSON de ofertas recomendadas	73
Código 5	74

Apéndice

- A. Caso de Uso - Registro de Usuarios **¡Error! Marcador no definido.**
- B. Caso de Uso - Diagrama UML: Login y Registro de instituciones educativas **¡Error! Marcador no definido.**
- C. Diagrama de estado: login **¡Error! Marcador no definido.**
- D. Caso de Uso - Diagrama UML: Búsqueda de ofertas educativas **¡Error! Marcador no definido.**
- E. Caso de Uso - Diagrama UML: DasBoard Admin **¡Error! Marcador no definido.**
- F. Caso de Uso - Diagrama UML: Dashboard estudiante **¡Error! Marcador no definido.**
- G. Caso de Uso - Diagrama UML: Dashboard instructor **¡Error! Marcador no definido.**
- H. Caso de Uso - Diagrama UML: test de inteligencias **¡Error! Marcador no definido.**
- I. Modelo relacional para la seguridad **¡Error! Marcador no definido.**
- J. Modelo relacional para la administración de las ofertas **¡Error! Marcador no definido.**
- K. Modelo relacional para inscripción de ofertas **¡Error! Marcador no definido.**
- L. ER Información usuarios e intereses **¡Error! Marcador no definido.**
- M. Cronograma de actividades Pt. 1 **¡Error! Marcador no definido.**
- N. Cronograma de actividades Pt. 2 **¡Error! Marcador no definido.**

Introducción

Actualmente existe un beneficio considerable por la capacidad del internet para dar soporte a potentes formas de aprendizaje y comunidades de práctica digitalmente dispersas. Así, internet se considera una poderosa herramienta que permite el aprendizaje a través de actividades acreditadas y de la interacción entre personas y ambientes sociales más amplios (Selwyn, N. 2013).

En el entorno de aprendizaje han surgido diversas plataformas educativas específicas en algún área que pretenden complementar el conocimiento y satisfacer las necesidades de los estudiantes. Diversos sistemas implementados se relacionan con brindar ofertas educativas que son instruidas por las mismas instituciones, a nivel nacional encontramos Sena Sofía Plus, politécnico de Colombia, Ucc.edu, Estúdialo, Universia.net, entre otras y las que se destacan a nivel internacional son Coursera, Udemy, Funiber, Google, Tutellus, entre otras “Plataformas educativas que ofrecen formación online” (Educación 3.0, 2020). Sin embargo, ninguno de éstos portales se enfoca en brindar una prueba de múltiples inteligencias a los estudiantes que desean conocer sus fortalezas y debilidades educativas.

Este proyecto pretende desarrollar una plataforma de múltiples oferentes en la que los usuarios puedan buscar y filtrar ofertas educativas ofrecidas por diversas entidades o instructores, donde los estudiantes obtengan facilidad en la búsqueda con un banco de ofertas para pregrado, postgrados, tecnologías, técnicos, curso, entre otros. Brindando además un test cognitivo que ayuda a las personas a determinar con sus cualidades que le es más fácil estudiar, retornando

múltiples ofertas educativas según su test de guía vocacional, apoyado también de un sistema de recomendación ya que son herramientas que generan sugerencias sobre un determinado objeto de estudio, a partir de las preferencias y opiniones dadas por otros usuarios o por su mismo contenido (Beltrán Páez Germán, 2015).

Implementando esta propuesta, disminuirá la tasa de deserción en la educación por parte de los estudiantes, permitiéndoles desarrollar al máximo sus habilidades académicas y que éstos estén preparados para afrontar las demandas en el mercado laboral.

Justificación

“La educación es un derecho de la persona y un servicio público que tiene una función social, con ella se busca el acceso al conocimiento, a la ciencia, a la técnica y a los demás bienes y valores de la cultura”. Art. 67 (2008) de la Constitución Política de Colombia.

Existen diversas plataformas de recursos educativos, tales como Sena Sofía, UCC, Edx, entre otros, los cuales brindan apoyo al estudiante mediante la búsqueda, aprendizaje o enseñanza virtual y presencial de ofertas educativas. Sin embargo, uno de los problemas de estas plataformas es que solo ofrecen ofertas pertenecientes a la misma institución y no se preocupan por saber si el estudiante tiene el perfil cognitivo para afrontar ese reto. Por ende, el estudiante debe hacer búsquedas de ofertas o recursos educativos en diferentes plataformas, encontrándose con un gran número de resultados, lo cual conlleva a una pérdida de tiempo y a no encontrar un recurso educativo que se adapte adecuadamente a su perfil vocacional.

Según un artículo de la revista Semana titulado “La mayoría de universitarios no terminan la carrera que empiezan” (Semana, 2017), habla sobre la deserción de los estudiantes entre programas académicos y el tiempo valioso que pierden estudiando algo que no van a necesitar. Pero, que, además, indican que la deserción es una cuestión de mala elección de los recursos educativos o programas elegidos y que esto se debe a que, en ocasiones, estos no consideran sus aptitudes o cualidades como un factor a la hora de elegir su perfil vocacional, si no que van por la opción que está de moda o la de mejores ingresos.

Para finalizar, en el artículo titulado “Estudio del razonamiento lógico-matemático desde el modelo de las inteligencias múltiples (Ferrándiz, C., Bermejo, R., Sainz, M., Ferrando, M., & Prieto, M. D., 2008), se concluye que, según investigaciones realizadas en estudiantes, aplicando la teoría de inteligencias múltiples de Howard Gardner, dio resultados positivos en su aprendizaje. Por lo tanto, es factible que jóvenes con la realización de un test de inteligencias múltiples logren en conjunto ayudar al estudiante de una manera más asertiva en la elección de su perfil vocacional o recursos que le sean más fáciles de estudiar según sus capacidades cognitivas, cualidades y/o gustos. Sumado a esto se desea añadir un sistema de recomendaciones de filtro colaborativo y basado en contenido los cuales ayudan a generar sugerencias sobre un determinado objeto de estudio, a partir de las preferencias y opiniones dadas por otros usuarios (Beltrán Páez Germán, 2015), los cuales agilizan de manera considerable la elección de determinado recurso (Betancur, 2009).

Problema

¿Cómo se puede ayudar al estudiantado a elegir de manera más asertiva su carrera profesional de acuerdo a su perfil cognitivo?

Se desarrollará un ecosistema de aplicaciones llamado Educare, el cual está basado en la clasificación de perfiles cognitivos del usuario mediante un test de inteligencias múltiples apoyado en la teoría de Howard Gardner y un sistema de recomendaciones que implementa un algoritmo basado en contenido y de filtro colaborativo, que permite localizar recursos educativos adecuados al usuario, con ellos se logra que el usuario tenga un mejor acercamiento a las carreras o cursos que se acomodan a su perfil vocacional mediante sus capacidades, cualidades y/o gustos.

Objetivos

Objetivo General

Desarrollar una plataforma de ofertas educativas de múltiples oferentes, apoyado en el test de inteligencias múltiples de Howard Gardner e integrado junto a un sistema de recomendación, la cual ayudará a orientar a los estudiantes en la elección de su perfil vocacional.

Objetivos Específicos

- Implementar el test de inteligencias múltiples de Howard Gardner, mediante un algoritmo de clasificación y promediado relacionado con cada fase de inteligencia, mostrando al estudiante cuál es la más predominante para posteriormente ofrecerle ofertas basada en su resultado.
- Desarrollar un modelo de recomendación mediante el uso de algoritmos de filtro colaborativo y basado en contenido, retornando ofertas educativas dependiendo del perfil cognitivo del usuario e historial de búsqueda.
- Implementar un ecosistema de aplicaciones basada en la arquitectura de Micro-servicios con Angular, Java y Python, en busca de optimizar el desarrollo, mejorando el rendimiento y la escalabilidad del sistema a futuro.

Alcances y limitaciones

Esta plataforma inicialmente estará disponible en el país de Colombia, para todas las personas interesadas en la búsqueda de nuevos conocimientos aptos para su perfil cognitivo o vocacional, pretendiendo conectar estudiantes con instituciones educativas y/o educadores, haciendo de esta una plataforma de múltiples oferentes.

Una de las limitaciones de este proyecto puede abarcar en la falta de actualización de registros en las ofertas ofrecidas por educadores o instituciones en el país. Además, generar un lapso de tiempo requerido para una mayor recolección de datos con el fin de hacer un reentrenamiento del modelo recomendador con mayor precisión.

Marco Teórico

Inteligencia

Concepto de inteligencia

La inteligencia es el término global mediante el cual, se describe una propiedad de la mente en la que se relacionan habilidades tales como: las capacidades del pensamiento abstracto, el entendimiento, la comunicación, el raciocinio, el aprendizaje, la planificación y la solución de problemas. Olea (2013).

Además, como lo menciona Prieto (2014), el desarrollo de la inteligencia no es equivalente al desarrollo del potencial humano, es decir, no son garantías de que la persona posea valores que humanizan, se está dejando de lado la parte emocional, los sentimientos, lo afectivo y la dimensión espiritual y ética que ha de impregnar el pensamiento, el sentimiento y la acción, orientada por valores que dan sentido a la vida.

El diccionario de la Real Academia Española de la lengua define la inteligencia (del latín *intellegentia*), entre otras acepciones como la "capacidad para entender o comprender" y como la "capacidad para resolver problemas".⁵ La inteligencia parece estar ligada a otras funciones mentales como la percepción o capacidad de recibir información y la memoria, o capacidad de almacenamiento.

Inteligencias múltiples

En 1979, la fundación Bernard Van Leer (grupo filantrópico irlandés) se acerca a la Universidad de Harvard y le pide a un grupo de investigadores, entre ellos Howard Gardner, que investiguen acerca del potencial humano. Así se llevó a cabo el "Proyecto Cero". Sus estudios culminaron en 1983 con su libro "Frames of mind" (Estructuras de la mente), en el que, desafiando esa creencia tan generalizada, formuló

una nueva teoría de la inteligencia, que supuso un cambio muy profundo en cuanto a idea y la concepción que se tenía de ella.

Como punto de partida, “la Teoría de las Inteligencias Múltiples” tenía dos corrientes muy influyentes: por un lado, la teoría de Jean Piaget, que consideraba que el pensamiento humano intentaba alcanzar el ideal del pensamiento científico; y por el otro, la concepción predominante de inteligencia, que la ligaba a la habilidad de responder a test, poniendo en juego habilidades lógico - matemáticas y lingüísticas.

Durante más de veinte años, ha hecho sucesivos reajustes y reformulaciones de su teoría y ha alentado para que florezcan nuevas propuestas. Gardner definió a la Inteligencia como “la habilidad para resolver problemas o enfrentar situaciones conflictivas de manera coherente e ingeniosa y para diseñar o crear un producto valorado, al menos, por una cultura o comunidad”. De acuerdo a esta definición la inteligencia cambia, ya que a través de los años lo que las distintas culturas valoran, también se modifica.

La teoría de las inteligencias múltiples se basa en los descubrimientos neurológicos realizados en Harvard y otras universidades de Estados Unidos, que cambiaron las líneas del conocimiento neuro - lógico sobre la mente humana y cuestionaron procesos anteriormente existentes para explicar sistemas neurales que involucran la memoria, el aprendizaje, la conciencia, las emociones y las inteligencias en general. Las repercusiones del desarrollo científico por el conocimiento del cerebro son muy significativas para la medicina, pero también para la educación, ya que provee nuevas bases para la comprensión del proceso de aprendizaje y, además, posibilita el desarrollo de estímulos para las inteligencias y trata los disturbios ligados a la atención, la creatividad y la memoria.

Prieto menciona algunas de las características de la teoría de las inteligencias múltiples. Sus características esenciales son:

- Se rechaza la concepción estándar, según la cual existe un solo tipo de inteligencia que, a su vez, deriva en una visión unidimensional acerca de la forma de evaluarla y no se plantea ni se visualiza la diversidad que existe entre los alumnos ni se les presta atención a la misma. Parte de una perspectiva expresada en una visión polifacética de la inteligencia que elucida los

mecanismos cerebrales que subyacen en las inteligencias que tiene cada individuo.

- Para cada tipo de inteligencia, el cerebro posee distintos mecanismos y operaciones que son identificables. Cada inteligencia constituye un sistema computacional basado en neuronas, que se activa a partir de ciertos tipos de información presentada en forma interna o externa.
- Cada una de las inteligencias es neurológicamente autónoma y relativamente independiente de las otras en su funcionamiento, pero cuando se aplican a un campo o a una disciplina, trabajan siempre concertadas. Cualquiera se puede aplicar a cualquier campo.
- Cada uno de los diferentes tipos de inteligencia necesita alcanzar un mínimo de bagaje intelectual, “competencias básicas” según Gardner, para funcionar adecuadamente. La inteligencia es una capacidad o destreza que se puede desarrollar.
- Todos tenemos una combinación de los tipos de inteligencia, con diferentes grados de desarrollo: una o dos preponderantes y otra u otras más débiles.
- Cada una de estas inteligencias se desarrolla en cada individuo de distinto modo y en diferentes niveles.
- Hay diferentes maneras de ser inteligente dentro de cada tipo de inteligencia.

Algunas de las conclusiones de Prieto son que, al aplicar las teorías de las inteligencias múltiples en la práctica pedagógica, las capacidades y habilidades y la combinación de las diferentes inteligencias son distintas en cada individuo y no todos los alumnos tienen los mismos centros de interés ni aprenden de la misma manera.

A continuación, Prieto da una descripción de las diferentes inteligencias:

Inteligencia Lingüística.

Se relaciona con la capacidad para usar las palabras de forma efectiva, sea de manera oral o de manera escrita, con el fin de comunicarse y expresar el propio pensamiento y darle un sentido al mundo mediante el lenguaje. Esta inteligencia incluye la habilidad de manipular la sintaxis o estructura del lenguaje, la fonética o sonidos del lenguaje, la semántica o significados del lenguaje y las dimensiones pragmáticas o usos prácticos del lenguaje. Las áreas cerebrales del sistema lingüístico tienen su asiento preferencial en el lóbulo frontal y temporal del hemisferio izquierdo, con dos importantes áreas como son las de Broca y la de Wernicke. La prosodia, o sea el contenido no verbal de la palabra (tono, frecuencia, volumen, ritmo, etc.), sería responsabilidad principal del hemisferio derecho. Es un tipo de inteligencia que utiliza ambos hemisferios. Esta inteligencia es la que permite un mayor y más fácil desarrollo de estrategias: narraciones orales de cuentos e historias; torrente de ideas; grabaciones de las propias palabras; entrevistas; diario personal; publicaciones como el diario de la escuela, correspondencia escolar, entre otras, son actividades que estimulan el desarrollo lingüístico. Ocupaciones características: bibliotecario, archivista, escritor, locutor de radio o televisión, periodista, abogado, secretario, dactilógrafo, docente de lengua, poetas, o simplemente personas para quienes la lengua es importante para el ejercicio de su profesión. Personas famosas: Jorge Luis Borges, Goethe, Julio Cortázar, Isabel Allende, entre otros.

Inteligencia lógico-matemática.

Es la capacidad para usar los números de manera efectiva y razonar adecuadamente. Esta inteligencia incluye la sensibilidad a los esquemas y relaciones lógicas, las afirmaciones y las proposiciones, las funciones y otras abstracciones relacionadas. Los tipos de procesos que se usan al servicio de esta inteligencia incluyen: la categorización, la clasificación, la inferencia, la generalización, el cálculo y la demostración de hipótesis.

Es el tipo de inteligencia más compleja en cuanto a la estructuración. Según Gardner, se expresa a través de cuatro competencias y habilidades:

- Habilidad para poder manejar una cadena de razonamientos en la forma de supuestos, proposiciones y conclusiones.
- Capacidad para darse cuenta de que las relaciones entre los elementos de una cadena de razonamientos de este tipo determinan el valor de éstas.
- Poder de abstracción: en lógica consiste en una operación de elaboración conceptual y en matemática es un proceso que comienza con el concepto numérico, pasa luego al concepto de dimensión variable y llega en su nivel más alto a la función de sus variables.
- Actitud crítica: consiste en que un hecho puede ser aceptado cuando ha sido posible su verificación empírica.

Inteligencia Musical.

La capacidad para reconocer, apreciar y producir ritmos, tonos, timbres y acordes de voces y/o instrumentos, percibir, discriminar, transformar y expresar las formas musicales. Para Gardner se expresa a través de tres competencias básicas: un sentido para los tonos (frecuencias), un sentido para el ritmo y un sentido para las tonalidades. Estas habilidades o competencias permiten comunicar, comprender y crear los significados de los sonidos.

La música es un lenguaje que tiene sus reglas, su gramática, etc. Es sonido autosuficiente y organizado, regido por reglas de armonía y contrapunto. Podemos presuponer que los procesos que se requieren para la actividad musical son de distinto tipo:

- Visuales: para la lectura de la notación musical, donde ésta no solo se presenta de manera secuencial, sino también con una forma y en un contexto espacial que interviene para dar significado a esa notación.
- Auditivos: permiten apreciar la belleza y estructura de una composición musical mediante la percepción y comprensión de las melodías, los timbres, los ritmos y la armonía que constituyen un

proceso acústico.

- Kinésicos: para la ejecución musical es necesaria una coordinación motora de altísima complejidad.
- Funciones cognitivas de tipo ejecutivo: para el desarrollo de las piezas musicales.
- Activaciones de circuitos afectivos: para explicar las activaciones emocionales que produce la música.

Inteligencia Visual-Espacial.

Habilidad para percibir de manera exacta el mundo visual-espacial y de ejecutar transformaciones sobre esas percepciones. Esta inteligencia incluye la sensibilidad al color, la línea, la forma, el espacio y las relaciones que existen entre estos elementos. La capacidad de visualizar, de representar de manera gráfica ideas visuales o espaciales y de orientarse de una manera adecuada en una matriz espacial. Otra competencia es la de reconocer el mismo objeto en diferentes circunstancias. Por último, la de anticiparse a las consecuencias de los cambios espaciales, que tan desarrollada tienen, por ejemplo, los maestros de ajedrez. Grandes flujos de información permiten que esto se realice con ubicación diferente en el cerebro en las regiones posteriores del hemisferio derecho: una localizada en la parte dorsal, que procesa lo relacionado con el espacio y otra ventral, relacionada con los objetos.

Inteligencia corporal-cinestésica.

Capacidad para utilizar todo el cuerpo para expresar ideas y sentimientos y la facilidad en el uso de las propias manos para transformar o producir cosas. Esta inteligencia incluye habilidades físicas específicas como la coordinación, el equilibrio, la destreza, la fuerza, la flexibilidad y la velocidad, así como las capacidades auto perceptivas, las táctiles y la percepción de medidas y volúmenes. Es la inteligencia del movimiento, la expresión y el lenguaje corporal, se trata de la sensibilidad que tiene una persona para manifestarse a través de un lenguaje no verbal. Tiene dos competencias básicas: en primer lugar, el

control de los movimientos corporales propios que posee el individuo; en segundo lugar, el tratamiento adecuado del manejo de objetos, expresado en destrezas y habilidades manuales para realizar actividades detalladas y de pequeñas dimensiones.

Inteligencia Interpersonal.

La capacidad de percibir y establecer distinciones en los estados de ánimo, las intenciones, las motivaciones y los sentimientos de otras personas. Se expresa hacia el exterior de la persona. Es la capacidad para relacionarse con otras personas y comprender sus sentimientos, sus formas de pensar, sentir y actuar, detectando sus motivaciones, preferencias e intenciones. Se expresa también en la capacidad para comunicarse con la gente y manejar los conflictos, gracias a una adecuada evaluación del manejo de emociones propias y ajenas. La inteligencia interpersonal nos hace capaces de sintonizar con otras personas y de manejar los desacuerdos antes de que se conviertan en rupturas insalvables. Se expresa en la capacidad empática que permite comprender el estado de ánimo de los demás y considerar al otro en su realidad de otro. Facilita la creación de un clima que valora la pluralidad y la diversidad como un hecho positivo. Permite asumir el punto de vista de los otros, es decir, ver las cosas desde la perspectiva de los demás. Un factor importante es el carisma, entendido como la habilidad de atraer a personas o grupos.

Inteligencia Intrapersonal.

Es la inteligencia del autoconocimiento, de la autoestima y de la capacidad de automotivación. Se expresa hacia el interior de la persona. Se refiere a poseer conocimientos sobre uno mismo. Implica la reflexibilidad del propio espíritu. Los individuos dotados de esta inteligencia tienden a saber lo que pueden hacer o no, lo cual les ayuda a tomar decisiones eficaces y eficientes sobre sus vidas. La metacognición es un proceso indispensable para el desarrollo de la inteligencia intrapersonal.

Se trata de la aptitud para el conocimiento introspectivo de uno mismo, que permite el análisis y el

manejo de las propias emociones, los sentimientos, intereses, capacidades y motivos. Esta inteligencia permite que los individuos, al observar sus estados y procesos neurocognitivos, tanto a nivel cognitivo como afectivo, estén en mejores condiciones para orientar sus comportamientos.

Inteligencia Naturalista.

Es la capacidad para distinguir entre los seres vivos, ya sean plantas o animales. Es un tipo de inteligencia relacionado con el mundo natural, que desarrolla la habilidad para identificar miembros de una misma especie y detectar las diferencias que existen entre ellos. Este tipo de inteligencia está presente en personas que saben observar, estudiar la naturaleza, clasificar elementos del medio ambiente y utilizar estos conocimientos productivamente. Gardner afirma que en la cultura consumista en la que estamos inmersos, los jóvenes aplican este tipo de inteligencia para discriminar tipos de automóviles, estilos de peinados o zapatillas. La atracción por descubrir el mundo natural y la inquietud por develar los misterios de la naturaleza son sus manifestaciones más significativas.

Sistemas de recomendación

Un sistema de recomendación es un sistema inteligente que proporciona a los usuarios una serie de sugerencias personalizadas (recomendaciones) sobre un determinado tipo de elementos (ítems). Los sistemas de recomendación estudian las características de cada usuario y mediante un procesamiento de los datos, encuentra un subconjunto de ítems que pueden resultar de interés para el usuario. Moya (2013).

En 1994 se desarrolló el primer “workshop” en Berkeley donde se vio la utilidad en diversas áreas de los primeros algoritmos simples de este tipo. También se identificaron algunas cuestiones importantes para el desarrollo de estos algoritmos: Escalabilidad, Viabilidad económica, puntuaciones implícitas y explícitas... Uno de los grupos de investigación pioneros en el desarrollo del filtrado colaborativo fue el proyecto GroupLens de la universidad de

Minesota que aún permanece muy activo y que ha proporcionado una gran parte de la base algorítmica de muchos sistemas de recomendación. Fueron los primeros en introducir el filtro colaborativo automático usando un algoritmo de búsqueda de vecinos para proporcionar predicciones en los grupos de noticias de USENET. Galán (2019)

“Un sistema de recomendación filtra los datos usando diferentes algoritmos y recomienda los ítems más relevantes a los usuarios. Primero captura el historial de comportamiento de un cliente y basado en esto, recomienda productos que los usuarios puedan comprar” (Sharma, 2018)

Capturar la información

Los datos se pueden recopilar de dos maneras: explícita e implícitamente. Los datos explícitos son información que se proporciona intencionalmente, es decir, aportes de los usuarios, como calificaciones de ofertas. Los datos implícitos son información que no se proporciona intencionalmente, sino que se recopila de los flujos de datos disponibles, como el historial de búsqueda, los clics, el historial de búsquedas, etc. Para este caso, vamos a tomar el historial de navegación de cada usuario y las calificaciones de las ofertas.

CONTENT-BASED FILTERING

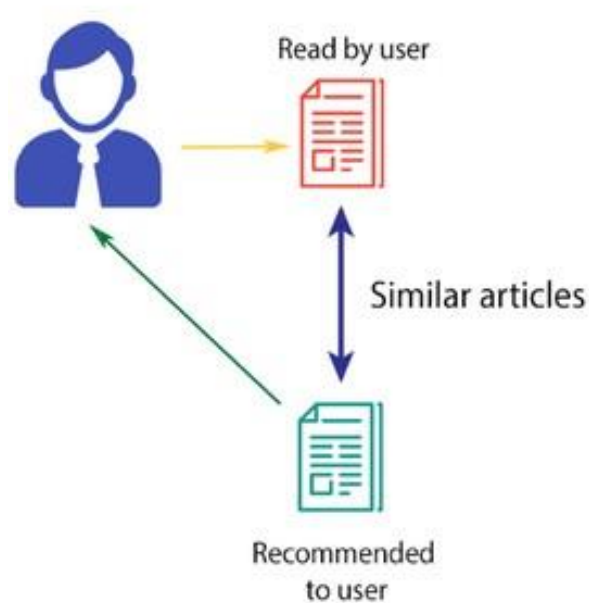


Figura 1. Ilustración de filtrado basado en contenido. Fuente Analyticsvidhya

Filtrado basado en contenido.

Este algoritmo recomienda productos que son similares a los que le han gustado a un usuario en el pasado. Ver figura 51. Por ejemplo, si a una persona le ha dado una buena calificación a enfermería, este algoritmo recomendará ofertas que pertenezcan al mismo género. Pero, ¿cómo entiende el algoritmo de qué género elegir y recomendar ofertas?

Considere el ejemplo de Netflix. Guardan toda la información relacionada con cada usuario en forma de vector. Este vector contiene el comportamiento pasado del usuario, es decir, las películas que le gustaron / no le gustaron y las calificaciones otorgadas por ellos. Este vector se conoce como el vector de perfil. Toda la información relacionada con las películas se almacena en otro vector llamado vector del elemento. El vector del elemento contiene los detalles de cada película, como género, reparto, director, etc.

El algoritmo de filtrado basado en contenido encuentra el coseno del ángulo entre el vector de perfil y el vector del elemento, es decir, la similitud del coseno. Suponga que A es el vector de perfil y B es el vector del elemento, entonces la similitud entre ellos se puede calcular como:

$$\sin(A, B) = \cos(\theta) = \frac{A * B}{\|A\| \|B\|}$$

Ecuación 1

Según el valor del coseno, que oscila entre -1 y 1, las películas se organizan en orden descendente y se utiliza uno de los dos enfoques siguientes para las recomendaciones:

- **Enfoque Top-n:** donde se recomiendan las mejores películas n (Aquí la empresa puede decidir n).
- **Enfoque de escala de calificación:** donde se establece un umbral y se recomiendan todas las películas por encima de ese umbral

Otros métodos que pueden ser utilizados para calcular la similitud son:

- **Distancia Euclidiana:** Los elementos similares se encontrarán muy cercanos entre sí si se trazan en un espacio n-dimensional. Por lo tanto, podemos calcular la distancia entre elementos y, en función de esa distancia, recomendar elementos al usuario. La fórmula para la distancia euclidiana viene dada por:

$$DistanciaEuclidiana = \sqrt{(x_1 - y_1)^2 + \dots + (x_n - y_n)^2}$$

Distancia Euclidiana

- **Correlación Pearson:** Nos dice cuánto están correlacionados dos elementos. Cuanto mayor sea la correlación, más será la similitud. La correlación de Pearson se puede calcular utilizando la siguiente fórmula:

$$\text{sin}(u, v) = \frac{\sum(r_{ui} - \bar{r}_u) \cdot (r_{vi} - \bar{r}_v)}{\sqrt{\sum(r_{ui} - \bar{r}_u)^2} \cdot \sqrt{\sum(r_{vi} - \bar{r}_v)^2}}$$

Correlación de Pearson

Una desventaja importante de este algoritmo es que se limita a recomendar elementos que son del mismo tipo. Nunca recomendará productos que el usuario no haya comprado o le haya gustado en el pasado. Entonces, si un usuario ha visto o le han gustado solo películas de acción en el pasado, el sistema recomendará solo películas de acción. Es una forma muy estrecha de construir un sistema de recomendación.

Para mejorar este tipo de sistema, necesitamos un algoritmo que pueda recomendar elementos no solo en función del contenido, sino también del comportamiento de los usuarios.

Filtrado colaborativo

Si a la persona A está interesado en 3 ofertas educativas, eje: Matemáticas, Física e ingeniería de sistemas y a la persona B le gustan Física, ingeniería de sistemas y mecánica, entonces tienen intereses casi similares. Por lo tanto, se puede decir con cierta certeza que a A le debería gustar mecánica y a B le debería interesar Matemáticas. El algoritmo de filtrado colaborativo utiliza "Comportamiento del usuario" para recomendar elementos. Este es uno de los algoritmos más utilizados en la industria, ya que no depende de ninguna información adicional. Existen diferentes tipos de técnicas de filtrado colaborativas.

Filtrado colaborativo usuario-usuario.

Este algoritmo primero encuentra el puntaje de similitud entre los usuarios. En función de este puntaje de similitud, selecciona a los usuarios más similares y recomienda productos que les gustaron o compraron anteriormente.

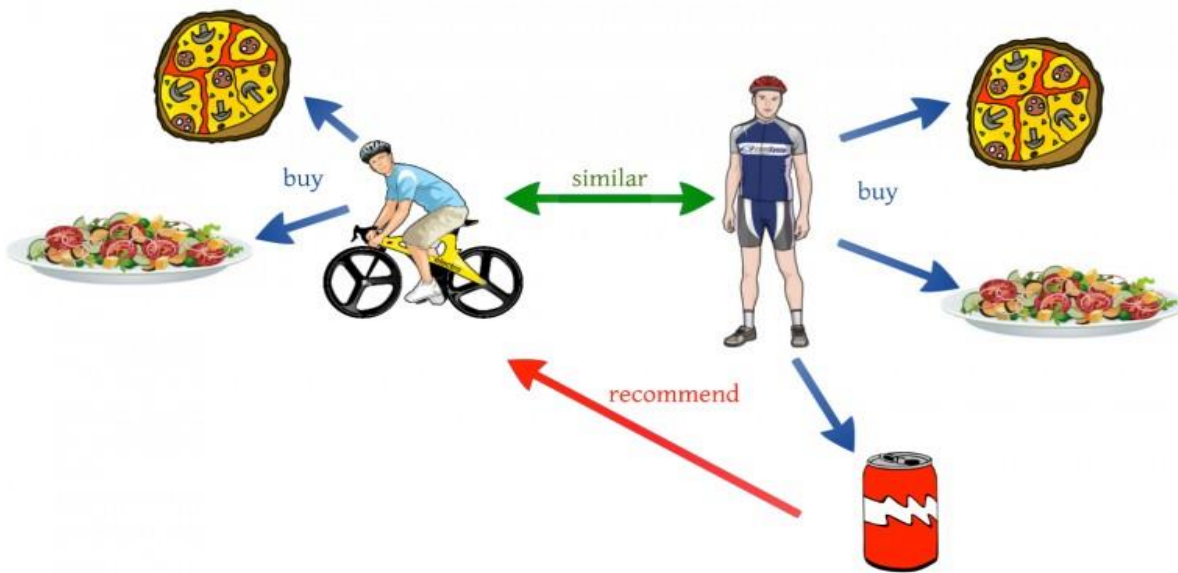


Figura 2. Ejemplo de recomendación basada entre usuarios. Fuente Analyticsvidhya

En términos de las ofertas, este algoritmo encuentra la similitud entre cada usuario en función de las calificaciones que han otorgado previamente a diferentes ofertas. La predicción de un elemento para un usuario u se calcula calculando la suma ponderada de las clasificaciones de usuario otorgadas por otros usuarios a un elemento i .

La predicción $P_{u,i}$ viene dada por:

$$P_{u,i} = \frac{\sum_v (r_{v,i} * s_{u,v})}{\sum_v s_{u,v}}$$

Ecuación 2

Donde:

- $P_{u,i}$ es la predicción de un ítem
- $r_{v,i}$ es la calificación de una oferta dada por un usuario v a una oferta i .
- $s_{u,v}$ es la similitud entre usuarios

Ahora, se tiene las calificaciones para los usuarios en el vector de perfil y con base a eso se predice que predecir las calificaciones para otros usuarios. Se siguen los siguientes pasos para hacerlo:

1. Para las predicciones se necesita la similitud entre el usuario u y v . Podemos hacer uso de la correlación de Pearson.
2. Primero se encuentra los elementos calificados por los usuarios y , en función de las calificaciones, se calcula la correlación entre los usuarios.
3. Las predicciones se pueden calcular utilizando los valores de similitud. Este algoritmo, en primer lugar, calcula la similitud entre cada usuario y luego, en función de cada similitud, calcula las predicciones. Los usuarios que tienen una mayor correlación tenderán a ser similares.
4. Con base a estos valores de predicción, se hacen recomendaciones.

A continuación un ejemplo:

Se considera la matriz de calificación de oferta de usuario:

Tabla 1.
Calificación de oferta por usuario

Usuario/Oferente	x1	x2	x3	x4	x5	Promedio Calificación
A	4	1		4		3
B		4		2	3	3
C		1		4	4	3

Aquí se tiene una matriz de calificación de ofertas por usuario. Para comprender esto de una manera más práctica, se busca la similitud entre los usuarios (A, C) y (B, C) en la tabla anterior. Las ofertas comunes clasificadas por A y C son oferta x2 y x4 y por B y C son las ofertas x2, x4 y x5.

$$r_{A,C} = \frac{[(1 - 3)^2(4 - 3)^2]}{[((1 - 3)^2 + (4 - 3)^2)^{1/2} * ((1 - 3)^2 + (4 - 3)^2)^{1/2}]}$$

$$r_{A,C} = 1$$

Ecuación 3

$$r_{B,C} = \frac{[(4 - 3)(1 - 3) + (4 - 3)(2 - 3) + (3 - 3)(4 - 3)]}{[((4 - 3)^2 + (2 - 3)^2 + (3 - 3)^2)^{1/2} * ((1 - 3)^2 + (4 - 3)^2 + (4 - 3)^2)^{1/2}]}$$

$$r_{B,C} = 0.866$$

Ecuación 4

La correlación entre el usuario A y C es más que la correlación entre B y C. Por lo tanto, los usuarios A y C tienen más similitudes y las ofertas que le gustan al usuario A se recomendarán al usuario C y viceversa.

Este algoritmo consume bastante tiempo, ya que implica calcular la similitud para cada usuario y luego calcular la predicción para cada puntaje de similitud. Una forma de manejar este problema es seleccionar solo unos pocos usuarios (vecinos) en lugar de todos para hacer predicciones, es decir, en lugar de hacer predicciones para todos los valores de similitud, elegimos solo unos pocos valores de similitud. Hay varias formas de seleccionar a los vecinos:

- Seleccionar un umbral de similitud y elija todos los usuarios por encima de ese valor

- Seleccionar al azar los usuarios
- Organizar a los vecinos en orden descendente de su valor de similitud y elija usuarios top-N
- Usar la agrupación para elegir vecinos

Este algoritmo es útil cuando el número de usuarios es menor. No es efectivo cuando hay una gran cantidad de usuarios, ya que tomará mucho tiempo calcular la similitud entre todos los pares de usuarios. Para solucionar este problema, se elabora un filtrado colaborativo elemento-elemento, que es efectivo cuando el número de usuarios es mayor que los elementos recomendados.

Filtrado colaborativo elemento-elemento.

Con este algoritmo, el objetivo es hallar la similitud entre cada par de ítems. En este caso, se halla la similitud de cada par de ofertas y, con base a eso, se recomiendan las ofertas similares que les gustaron a los usuarios en el pasado. Este algoritmo funciona de manera similar al filtrado colaborativo usuario-usuario con solo un pequeño cambio: en lugar de tomar la suma ponderada de las calificaciones de "usuarios-vecinos", se toma la suma ponderada de las calificaciones de "elementos-vecinos". La predicción viene dada por:

$$P_{u,i} = \frac{\sum_N (S_i, N * R_u, N)}{\sum_N (|S_i, N|)}$$

Ecuación 5

Ahora se halla la similitud entre ítems:

$$\sin(i, j) = \cos(\vec{i}, \vec{j}) = \frac{(\vec{i} * \vec{j})}{\|\vec{i}\|_2 * \|\vec{j}\|_2}$$

Ecuación 6

Ahora, como se tiene la similitud entre cada oferta y las clasificaciones, se hacen predicciones y, en función de esas predicciones, se recomiendan ofertas similares. Ej:

Tabla 2.
Clasificación de ofertas por calificación

Usuario/Oferta	x1	x2	x3	x4	x5
A	4	1	2	4	4
B	2	4	4	2	1
C		1		3	4
Prom. Calificación	3	2	3	3	3

Aquí, la calificación media del artículo es el promedio de todas las calificaciones otorgadas a un artículo en particular (comparado con la tabla 11). En lugar de encontrar la similitud usuario-usuario como se mostró anteriormente, se encuentra la similitud ítem-ítem.

Para hacer esto, primero se debe encontrar a esos usuarios que han calificado esos elementos y, en función de las calificaciones, se calcula la similitud entre los elementos. Se halla la similitud entre las ofertas (x1, x4) y (x1, x5). Los usuarios comunes que han calificado ofertas x1 y x4 son A y B, mientras que los usuarios que han calificado ofertas x1 y x5 también son A y B.

$$C_{1,4} = \frac{[(4 - 3)(4 - 3) + (2 - 3)(2 - 3)]}{\left[((4 - 3)^2 + (2 - 3)^2)^{1/2} * ((4 - 3)^2 + (2 - 3)^2)^{1/2} \right]} = 1$$

Ecuación 7

$$C_{1,5} = \frac{[(4 - 3)(4 - 3) + (2 - 3)(1 - 3)]}{\left[((4 - 3)^2 + (2 - 3)^2)^{1/2} * ((4 - 3)^2 + (1 - 3)^2)^{1/2} \right]} = 0.94$$

Ecuación 8

La similitud entre la oferta x1 y x4 es más que la similitud entre la oferta x1 y x5. Entonces, en función de estos valores de similitud, si algún usuario busca la película x1, se le recomendará la oferta x4 y viceversa.

Metodología de cascada

En la Ingeniería de software el desarrollo en cascada, también llamado modelo en cascada, es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior. EcuRed (2020).

Esta metodología está basada en la secuencia rigurosa de una serie de pasos, los cuales son ejecutados una vez el termine anterior, y está estructurada en los siguientes procesos:

- **Ingeniería y Análisis del Sistema:** Debido a que el software es siempre parte de un sistema mayor, el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.
- **Análisis de los requisitos del software:** el proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El ingeniero de software debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.
- **Diseño:** el diseño del software se enfoca en cuatro atributos distintos del programa; la

estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.

- **Codificación:** el diseño debe traducirse en una forma legible para la máquina. Si el diseño se realiza de una manera detallada, la codificación puede realizarse mecánicamente.
- **Pruebas:** una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.
- **Mantenimiento:** el software sufrirá cambios después de que se entrega al cliente. Los cambios ocurrirán debidos a que se haya encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos) o a que el cliente requiera ampliaciones funcionales o del rendimiento.

Metodología Agile

El desarrollo agile se refiere a un grupo de metodologías aplicadas a la creación de software que basa su desarrollo a en un ciclo iterativo, en el cual, las necesidades y soluciones evolucionan junto con la colaboración de los diferentes equipos involucrados en el proyecto. Esaú (2016).

Estas metodologías aparecieron del manifiesto agiles, el cual fue creado en el 2001 por un grupo de CEOs de las principales empresas de software de Utah. Tena (2018)

Manifiesto Agile

La reunión del 2001, de los principales CEOs, dio como resultado 12 principios de metodología ágiles, denominado **El manifiesto agile**, que tiene los siguientes principios:

1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
7. El software funcionando es la medida principal de progreso.
8. Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.

10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia

Estado del arte

En la actualidad, se encuentran muchas páginas que ofrecen ofertas educativas y muchas de estas son portales web de las universidades o institutos de brindan formación académica de pre-grado, postgrados u otros cursos. Algo interesante es que en la mayoría de los sitios de ofertas educativas. Solo hacen publicidad a las ofertas que ellos brindan, como es el caso de Sena Sofía, UCC, entre otras.

Se presenta a continuación una revisión del estado del estado sobre trabajos similares en el contexto educativo, aplicando las inteligencias múltiples de Howard Gardner y sistemas de recomendaciones:

Aplicaciones basadas en las inteligencias múltiples de Howard Gardner

Del mismo contexto educativo, existe una plataforma llamada Estúdialo. Esta es una plataforma de emprendimiento apoyada por el ministerio de tecnologías a través de la plataforma Apps.co, en el cual, han propuesto e implementado el test de inteligencias múltiples de Howard Gardner. Esta plataforma pretende conectar a personas con instituciones educativas, ofreciendo además opciones de financiación y permitiendo filtrar, buscar, orientar al usuario en la elección de mejores ofertas.

Aplicación y enfoque Sistemas de recomendación

Se han propuesto diferentes sistemas de recomendaciones en el contexto educativo. En (Bustos-López et. Al., 2015) se presentó una arquitectura para el desarrollo de sistemas de recomendaciones educativos sin importar el tipo de recomendación generada. Estas son propiedades importantes para crear cualquier sistema recomendación con el propósito de generar información más precisa a los usuarios de acuerdo a sus gustos, preferencias, perfil del usuario y aprendizaje.

Protus 2.1 es un proyecto que describe un módulo de recomendaciones de tutorías llamado

Potrus, el cual se adapta automáticamente a los interés y niveles de conocimiento de los alumnos. Potrus puede reconocer diferentes patrones de aprendizaje analizando los hábitos e intereses de los alumnos. Vesin B., Klačnja-Milićević A., Ivanović M. (2016)

Estadísticas SimilarWeb - Paginas Nacionales

A continuación, se hará una comparativa entre páginas de ofertas educativas colombianas, por un lado, tendremos la página del Servicio Nacional de Aprendizaje, Sena, con su portal web Sena Sofia y por el otro la propuesta de emprendimiento que más se acerca a nuestro proyecto de desarrollo, Estudialo.co

Performance del sitio Web.

En prosecución del análisis de esta página, se mostrará algunas gráficas de ranking a nivel mundial y nacional.

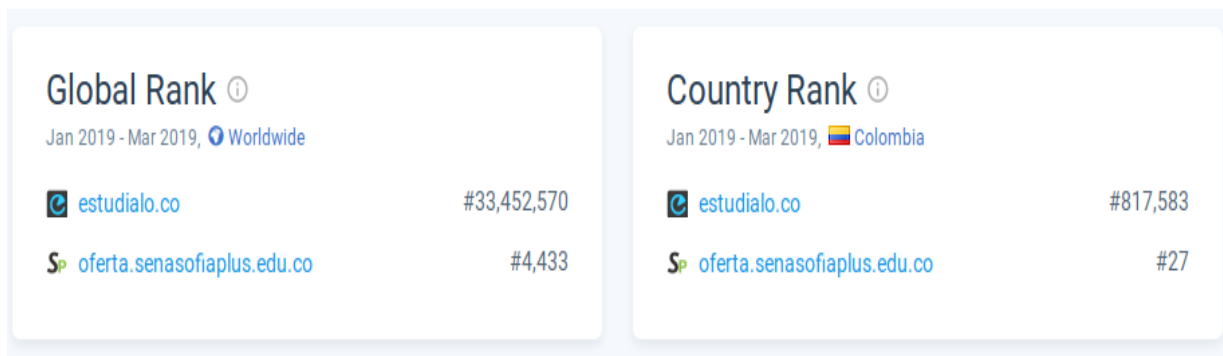


Figura 3. Website Performance entre Estudialo.co y SenaSofia Fuente: SimilarWeb

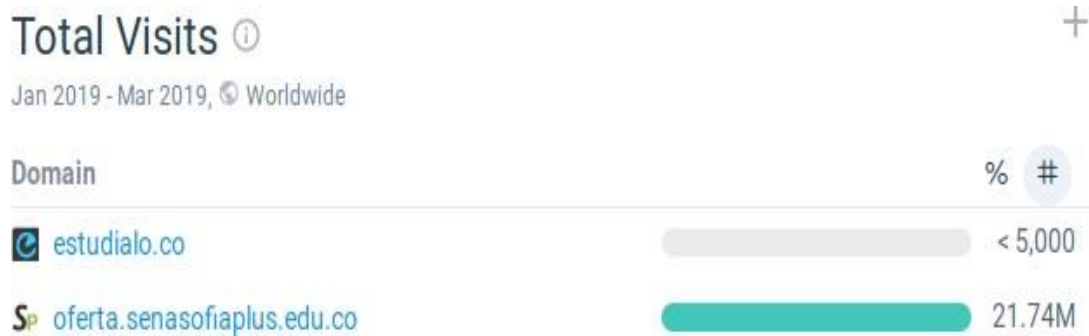


Figura 4. Visitas totales Fuente: SimilarWeb

En la figura 14 se aprecia la suma de todas las visitas (no únicas) por el periodo de tiempo analizado. Las métricas pueden ser vistas como parte de todos los dominios analizados o como un número absoluto

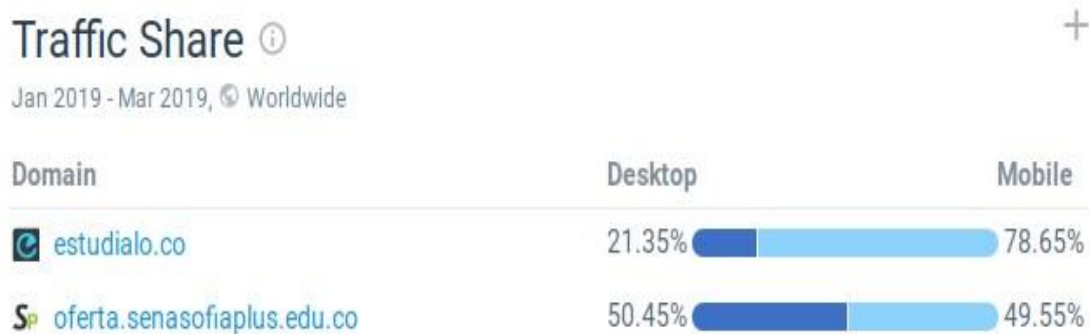
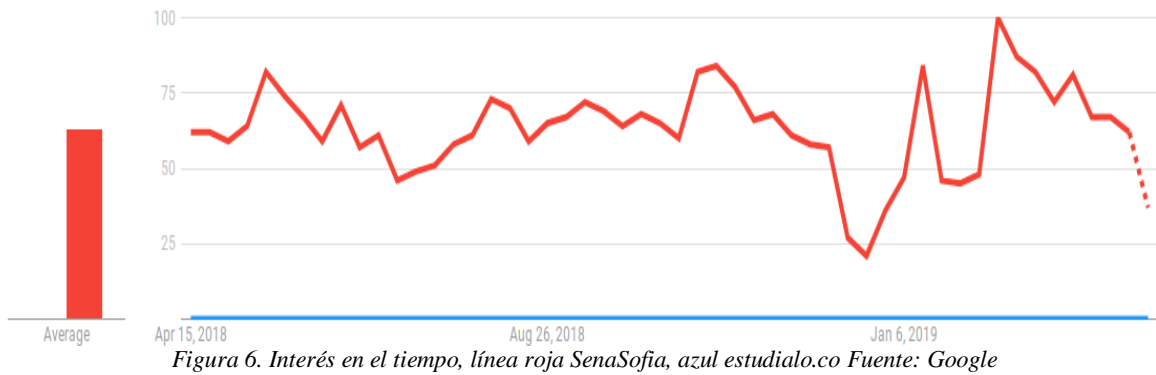


Figura 5. Porcentaje de trafico entrante, desktop vs. Móvil



Los números representan el interés de búsqueda en relación con el punto más alto en el gráfico para la región y el tiempo dados. Un valor de 100 es la popularidad máxima para el término. Un valor de 50 significa que el término es la mitad de popular. Una puntuación de 0 significa que no hubo datos suficientes para este término.

● estudialo ● sena sofia



Figura 7. Desglose comparado por subregión

Metodología

Desarrollo ágil y en cascada

La ingeniería de software ha intentado imitar la administración de tiempo y recursos tal como lo realiza la ingeniería civil utilizando un método de desarrollo llamado Waterfall, en español, cascada. El desarrollo en cascada ha sido el método tradicional utilizado en la industria del software; es un procedimiento estricto de recopilación de datos, producción, prueba y liberación; el cual no cuenta con libertades de modificación durante su ejecución, esto produce problemas de tiempo y dinero porque solo se puede empezar a ejecutar el proyecto una vez se ha recopilado todos los datos de requerimiento y además el cliente solo tiene acceso al software una vez terminado.

Otro modelo de desarrollo es el llamado Agile, en español, ágil. El desarrollo ágil permite dividir el proyecto en secciones más pequeñas, producen prototipos rápidos los cuales el mismo cliente tiene la posibilidad de probar y dar su opinión al respecto, el problema de esto es que requiere una participación mayor del cliente y muchas veces esto no es posible. Además, es muy dependiente del trabajo en equipo y requiere mayores tiempos de discusión y socialización de ideas.

Existen muchas más metodologías en el mercado, hasta ahora no hay una forma absoluta y completamente aceptada por toda la industria.

Para este desarrollo se considera usar la unión de Waterfall y Agile que se conoce como Agile-fall. Esta decisión surge debido a las fortalezas y debilidades que tiene cada método actualmente. En el caso de cascada destaca que es muy estricto a la hora de documentar requerimientos de usuario y la creación de un cronograma de actividades que va de la mano con dichos requerimientos. En el caso de ágil destaca que el desarrollo es muy rápido y va de la

mano con las pruebas, de esta forma se busca encontrar errores inmediatamente en lugar de tener que construir todo el software y luego corregir los errores que se presenten en la última etapa de desarrollo.

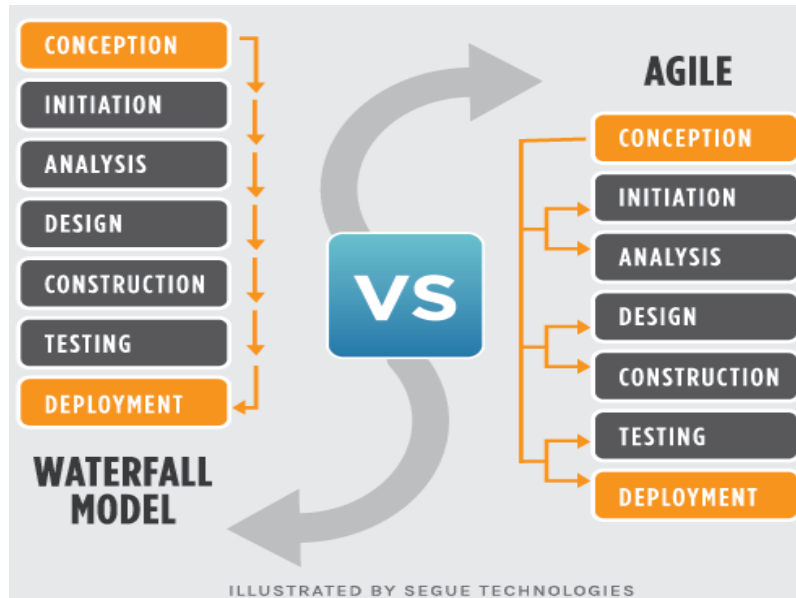


Figura 8. Waterfall model vs Agile. (Lotz, 2013)

Tabla 3.
Comparativa de implementación

Características	Waterfall (Cascada)	Agile (Ágil)
Disponibilidad del cliente	Requiere que el cliente se involucre solo en hitos del proyecto	Prefiere que el cliente siempre esté disponible en todo el proyecto
Alcance y características	Limita los cambios y funciona bien cuando se conoce el alcance del proyecto	Acepta cambios, pero estos tienen un costo adicional para el cliente. Funciona cuando no se conoce el alcance total del proyecto
Características y priorización	Todo o nada. Cascada hace todo lo que el cliente ha pedido	Prioriza las partes más importantes primero para así evitar quedarse sin fondos de desarrollo y al menos producir un éxito parcial
Equipo	La coordinación está limitada a puntos de traspaso	Prefiere equipos de desarrollo pequeños, dedicados, con un alto grado de coordinación y sincronización
Financiación	Reduce riesgos de financiación al firmar contratos de precios por adelantado	Funciona bien en donde los materiales y el tiempo no son fijos, pero incrementa el riesgo en con- tratos de precio por adelantado

Fuente: Lotz (2013)

Requisitos del sistema

A continuación, se presenta los requisitos de alto nivel o funcionales del sistema, los cuales son fundamentales para el correcto funcionamiento y van de acuerdo a lo planteado en los objetivos.

*Tabla 4.
Caso de uso para inicio de registro de usuario.*

Nombre	Registro usuario
Id de Requisito	RF001
Versión	1
Autor	Jhorkman Dariel Bernal Herrera, Jesús Santiago Tovar Cuenca
Fecha	26-02-2018
Descripción	Registro de usuario nuevo
Actores	Invitado, página web
Precondición	El usuario no está registrado en la página
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario deberá proporcionar los datos personales. 2. Se termina el proceso de registro
Flujo alternativo	El usuario no se puede registrar
Postcondición	<ol style="list-style-type: none"> 1. El usuario queda registrado en el sitio web 2. El usuario confirma su cuenta por correo

*Tabla 5.
Caso de uso para inicio de sesión de usuario*

Nombre	Login usuario web
Id de Requisito	RF002
Versión	1
Autor	Jhorkman Dariel Bernal Herrera, Jesús Santiago Tovar Cuenca
Fecha	26-02-2018
Descripción	Inicio de sesión del sitio web
Actores	Usuario registrado, página web
Precondición	El usuario debe estar registrado en la página
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario deberá proveer su nombre de usuario y contraseña 2. Se termina el proceso de ingreso de usuario
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario no puede ingresar 2. El usuario podrá utilizar redes sociales para registrar sus datos 3. La página mostrará un captcha en caso de varios intentos fallidos 4. La cuenta se bloquea temporalmente por seguridad tras varios intentos fallidos

Postcondición	El usuario ingresa al sitio web
----------------------	---------------------------------

Tabla 6.
Caso de uso para registro de oferta.

Nombre	Registro de oferta
Id de Requisito	RF003
Versión	1
Autor	Jhorkman Dariel Bernal Herrera, Jesús Santiago Tovar Cuenca
Fecha	26-02-2018
Descripción	Registro de nuevo oferta
Actores	Usuario registrado, página web
Precondición	El usuario debe haber ingresado utilizando su usuario y contraseña
Flujo Normal	1. Ingresar en un formulario de registro los datos personales. 2. El sistema valida los datos.
Flujo alternativo	1. El usuario no se puede registrar la oferta 2. El usuario podrá pagar una cuota Premium de publicación
Postcondición	La oferta queda registrado

Tabla 7.
Caso de uso para publicación de ofertas

Nombre	Publicación de ofertas
Id de Requisito	RF004
Versión	1
Autor	Jhorkman Dariel Bernal Herrera, Jesús Santiago Tovar Cuenca
Fecha	26-02-2017
Descripción	Publicación de la oferta registrada
Actores	Usuario registrado, página web
Precondición	Ingreso registrado en la plataforma, registro previo de la oferta
Flujo Normal	El usuario dará de alta su publicación
Flujo alternativo	1. El usuario podrá modificar sus publicaciones existentes. 2. El usuario podrá cancelar la orden de publicación 3. El usuario tendrá la opción de crear un nuevo registro de oferta 4. El usuario podrá pagar una cuota Premium de publicación
Postcondición	La oferta publicada se mostrará junto a todos los demás ofertas publicadas por otros usuarios

Tabla 8.
Caso de uso para modificación de información personal.

Nombre	Modificación de información personal o de la oferta
Id de Requisito	RF005
Versión	1
Autor	Jhorkman Dariel Bernal Herrera, Jesús Santiago Tovar Cuenca
Fecha	26-02-2017
Descripción	Modificación de información personal o de la oferta

Actores	Usuario registrado, página web
Precondición	<ul style="list-style-type: none"> • Ingreso registrado en la plataforma • Registro previo del inmueble • Solo puede hacer cambios de los datos personales tres veces en seis meses para evitar
Flujo Normal	<ol style="list-style-type: none"> 1. Modificación de la oferta y datos personales 2. Dar de alta la publicación 3. Dar de baja la publicación
Flujo alternativo	El usuario no podrá cambiar sus datos personales si ha usado sus oportunidades durante un periodo de seis meses
Postcondición	La información actualizada será publicada de manera inmediata para evitar que las personas que interesadas en una oferta tengan confusiones

Tabla 10.
Caso de uso: procedimiento para la realización del test vocacional

Nombre	Procedimiento para realizar el test vocacional
Id de Requisito	RF006
Versión	1
Autor	Jhorkman Dariel Bernal Herrera, Jesús Santiago Tovar Cuenca
Fecha	26-02-2017
Descripción	Proceso por el cual el usuario realiza el test vocacional
Actores	Usuario, página web
Precondición	<ul style="list-style-type: none"> • Ingreso a la plataforma
Flujo Normal	<p>El usuario se dirige a la página de test vocacional, seguidamente aparecerá una ventana con grilla de 4*2 casillas en las cuales tendrá una imagen correspondiente a una inteligencia. Figura 46. Además, esta página contará con 8 fases, en cada una el usuario por lo menos deberá escoger dos imágenes de su agrado y máximo cuatro (4).</p> <p>Al completar las ocho (8) fases, se le dará un aviso de que ha terminado el test vocacional, figura 48, después (si el usuario esta logeado, de lo contrario, deberá iniciar sesión), se mostraba una lista de las oferta que se acomoda a su perfil cognitivo, dicha lista será generada por el sistema de recomendaciones.</p>
Flujo alternativo	El usuario no esta tiene cuenta, por lo tanto, deberá crear una cuenta, una vez creada la cuenta e iniciado sesión, se le dará el resultado del test.
Postcondición	Se mostrara el listado de ofertas educativas que se acomodan al perfil cognitivo del usuario.

Gestión de tareas

Asana es una aplicación colaborativa para la gestión de tareas. El producto tiene muchas funcionalidades, como espacios de trabajo, proyectos, proyectos personales, tareas, etiquetas, notas, comentarios y un buzón que organiza y actualiza la información en tiempo real. El producto está diseñado para facilitar a las personas y a los equipos la planificación y la gestión de sus proyectos y tareas. Cada equipo tiene un espacio de trabajo. Los espacios de trabajo contienen proyectos y los proyectos contienen tareas. (Parra, 2016)

Al utilizar un asistente de proyectos se puede llevar un mejor control del personal y del tiempo invertido en la realización del software. De esta manera consigue una visión más clara del estado de desarrollo.

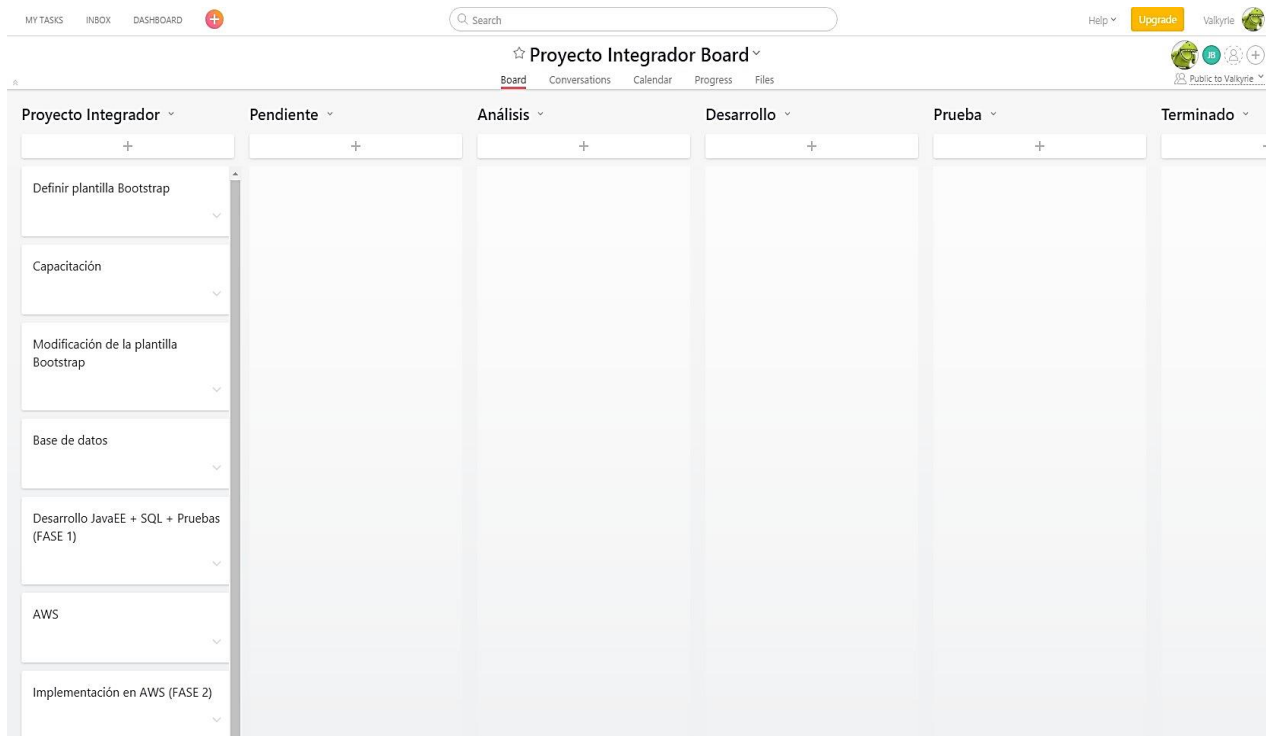


Figura 9. Dashboard de proyectos de Asana. (Asana.com, 2017)

Las tareas estarán divididas en diferentes tableros:

- Para hacer
- En progreso
- Finalizado
- Testing
- Producción

Diseño

A continuación, se hace anexa los diagramas UML de casos de uso y clases, los cuales ilustran de manera general el funcionamiento de la aplicación.

En la figura 20, se muestra de resumida los principales componentes de la aplicación, y como los usuarios interactúan con cada una de ellas, además, se muestra como está relacionado el sistema de recomendación con las búsquedas y el test de inteligencias.

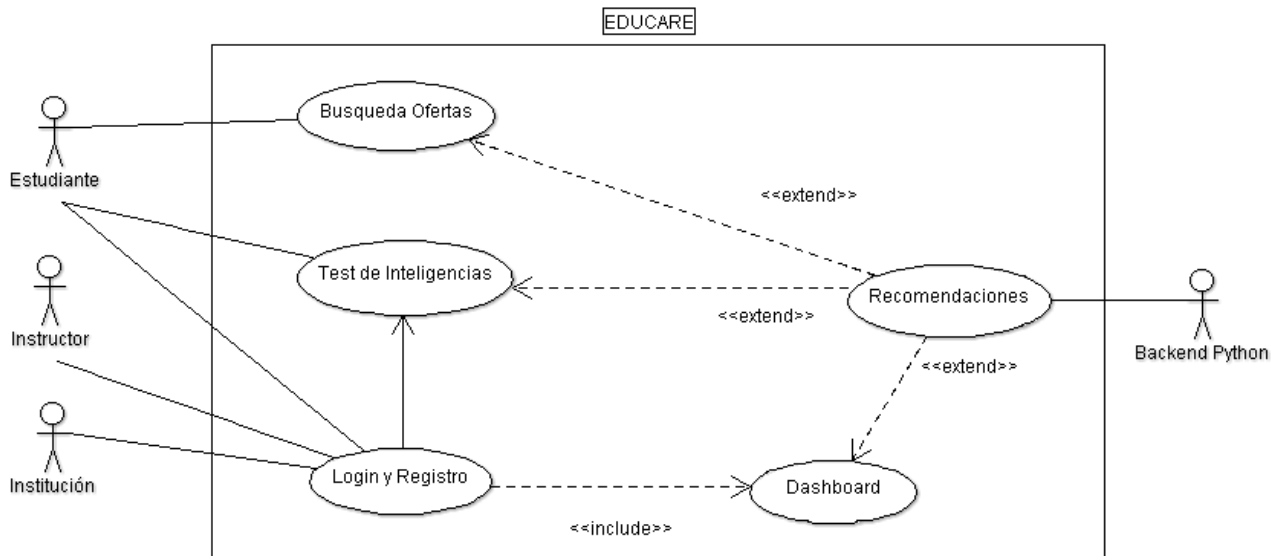


Figure 20 Módulos generales del proyecto

Casos de Uso

Login y registro de usuario

Como la mayoría de sistemas web, está permitido el registro de usuarios y su respectiva autenticación, por lo tanto, en la figura 21 se ilustra la interacción del usuario al registrarse o iniciar sesión en el sistema.

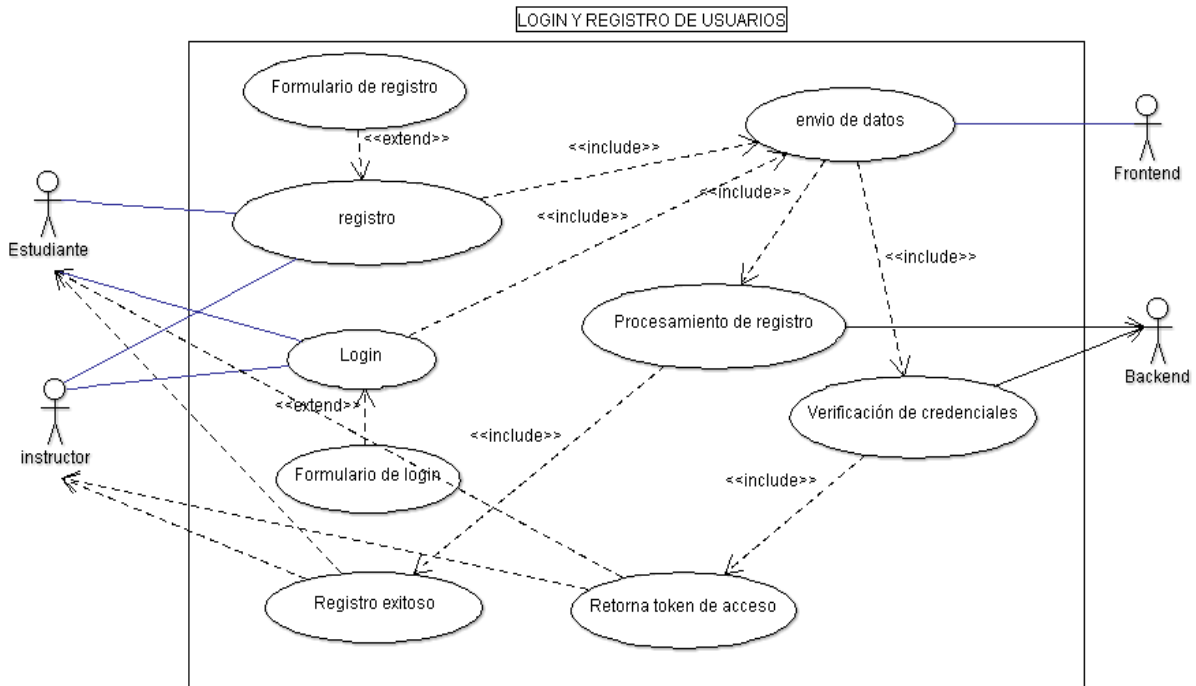


Figura 21. Registro de usuarios

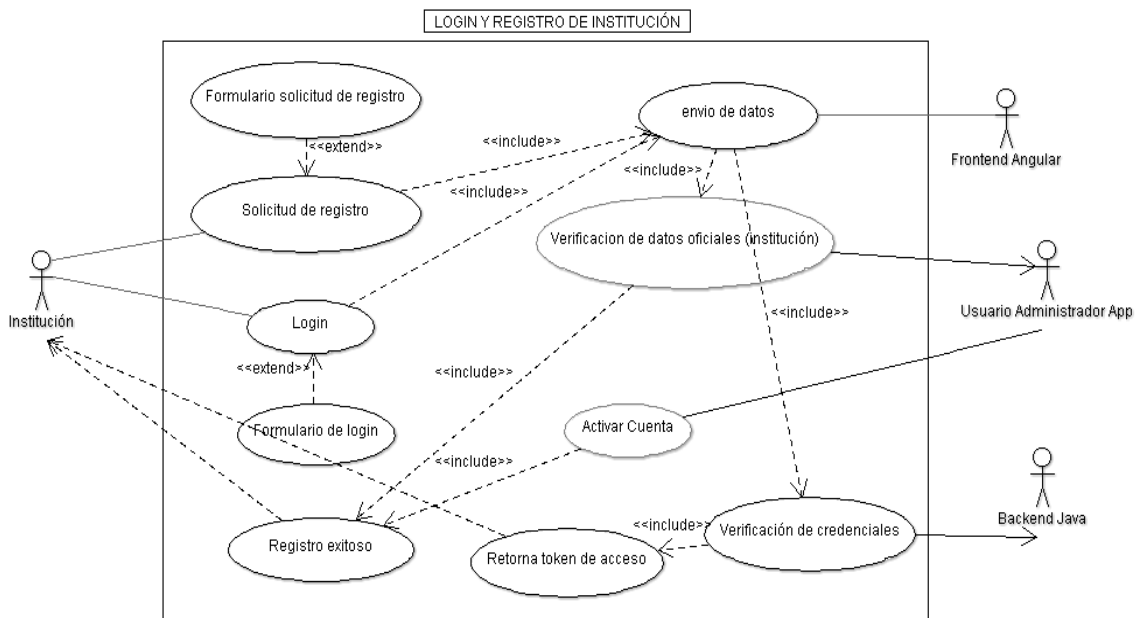
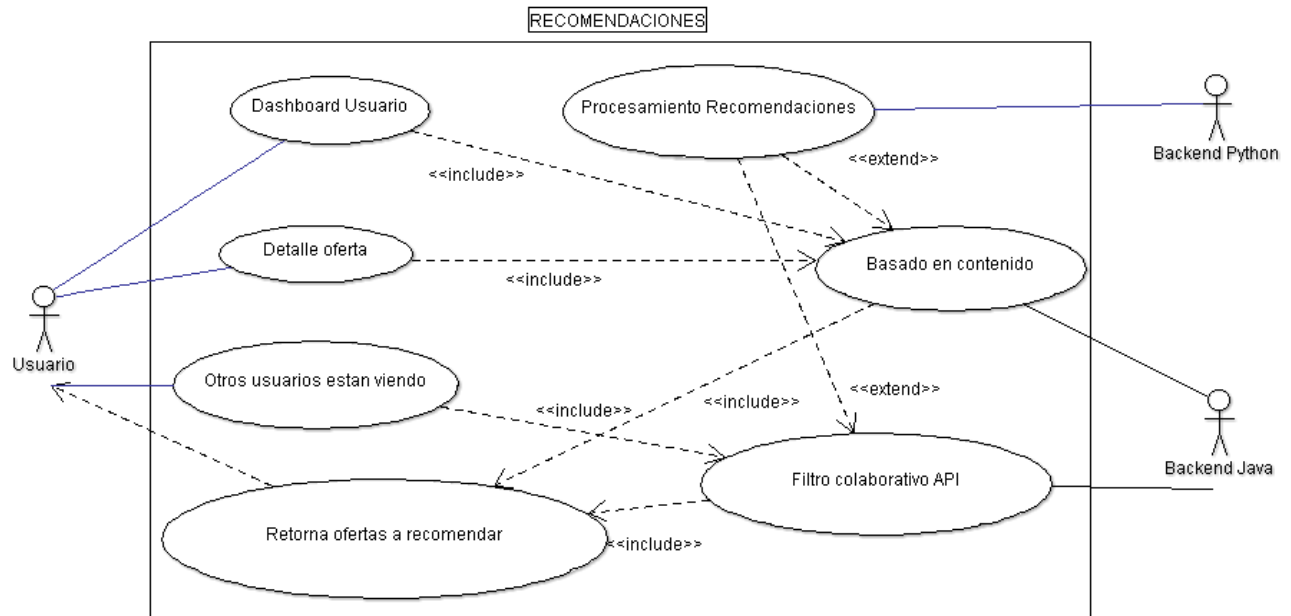


Figura 10. Diagrama UML: Login y Registro de instituciones educativas



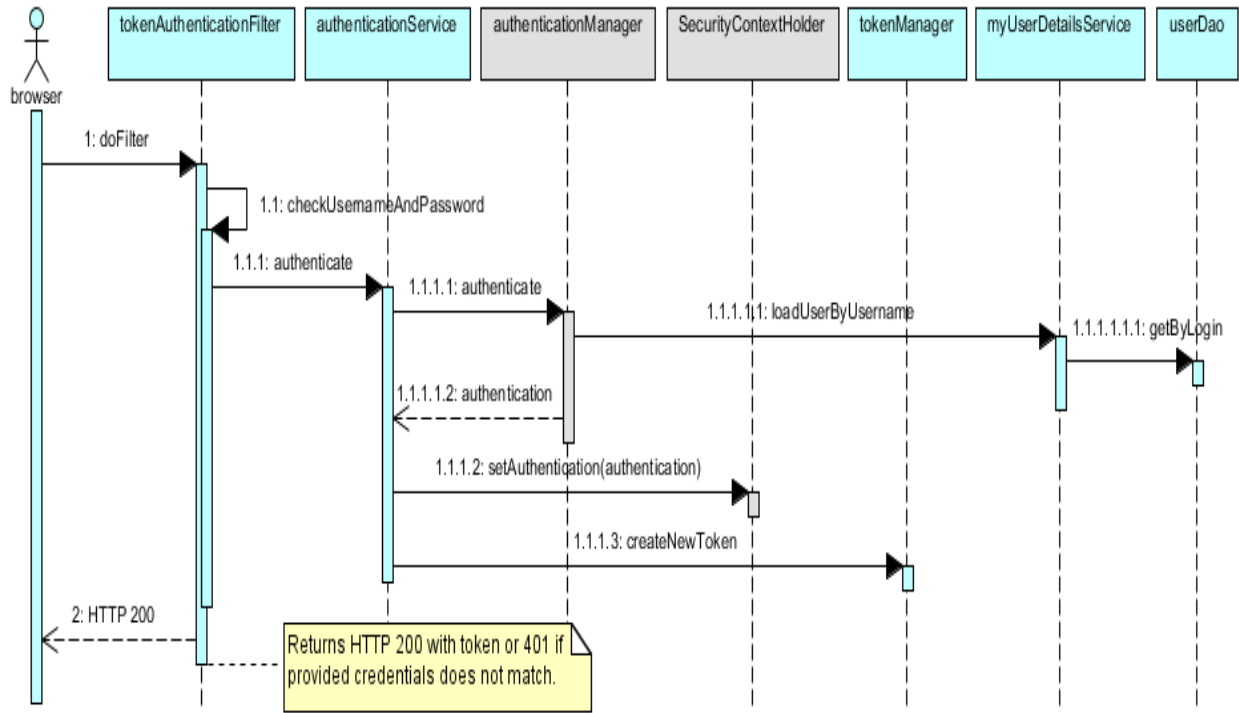


Figura 11. Diagrama de estado: login

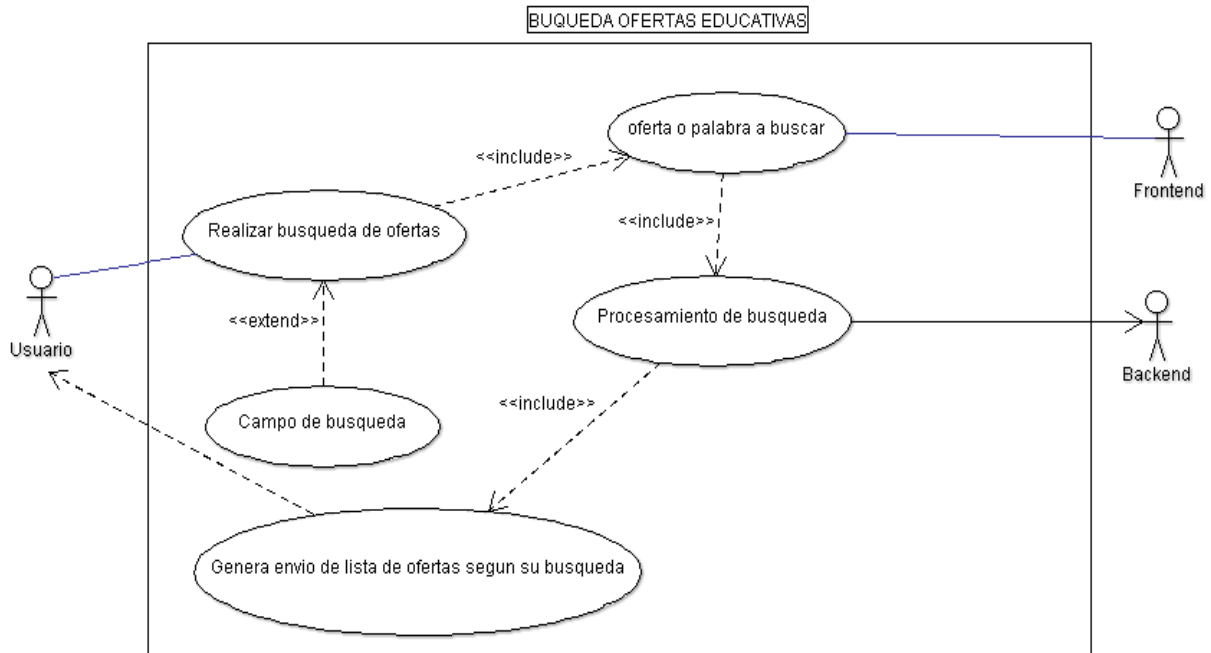


Figura 12. Diagrama UML: Búsqueda de ofertas educativas

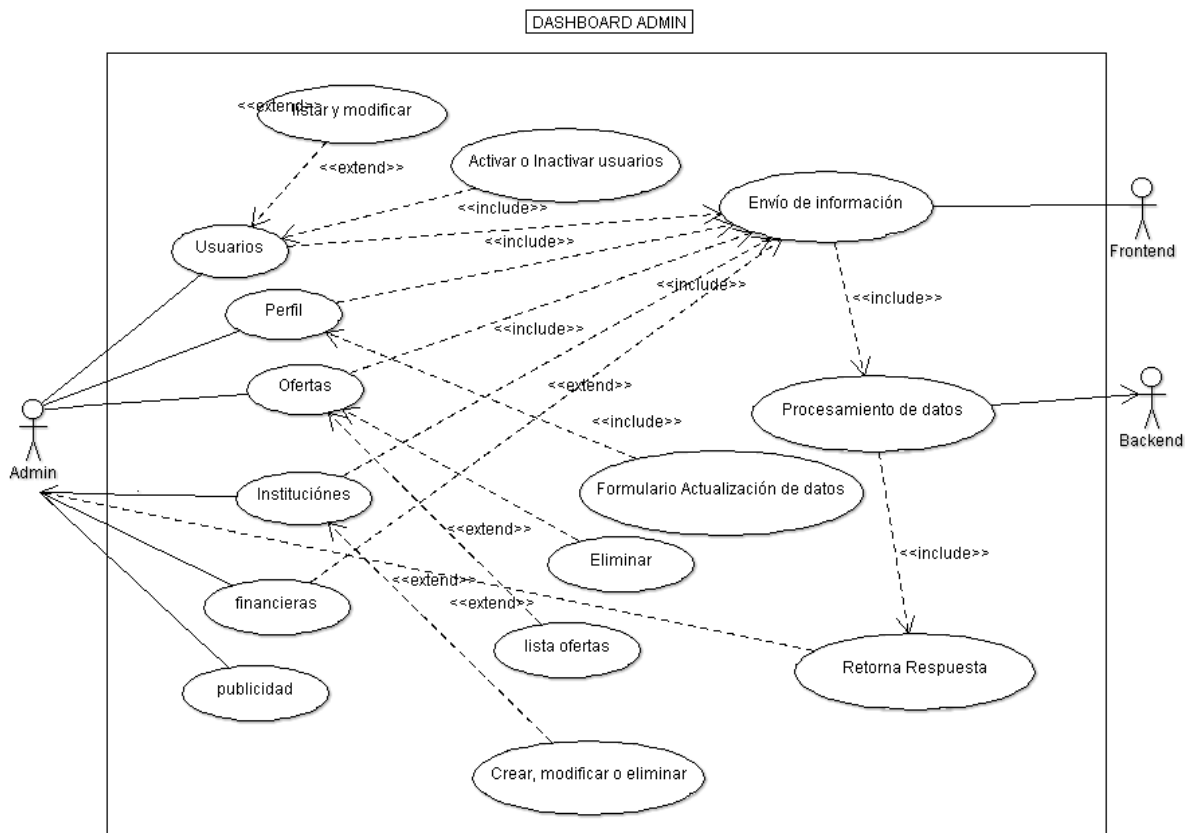


Figura 13. Diagrama UML: DasBoard Admin

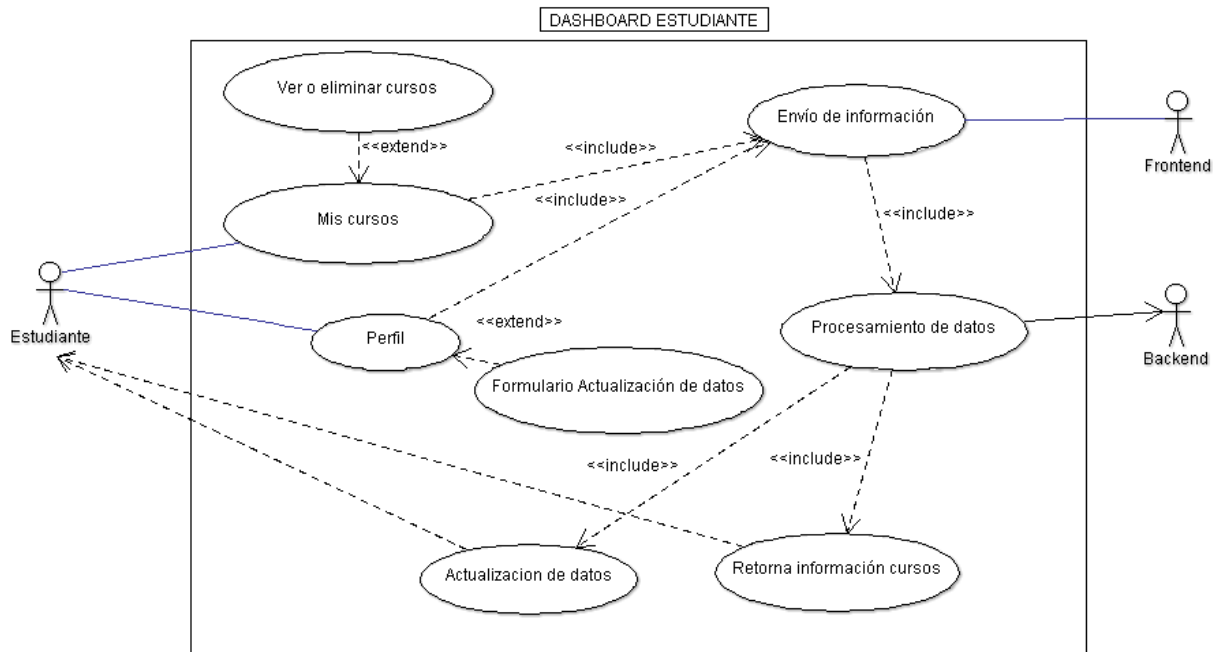


Figura 14. Diagrama UML: Dashboard estudiante

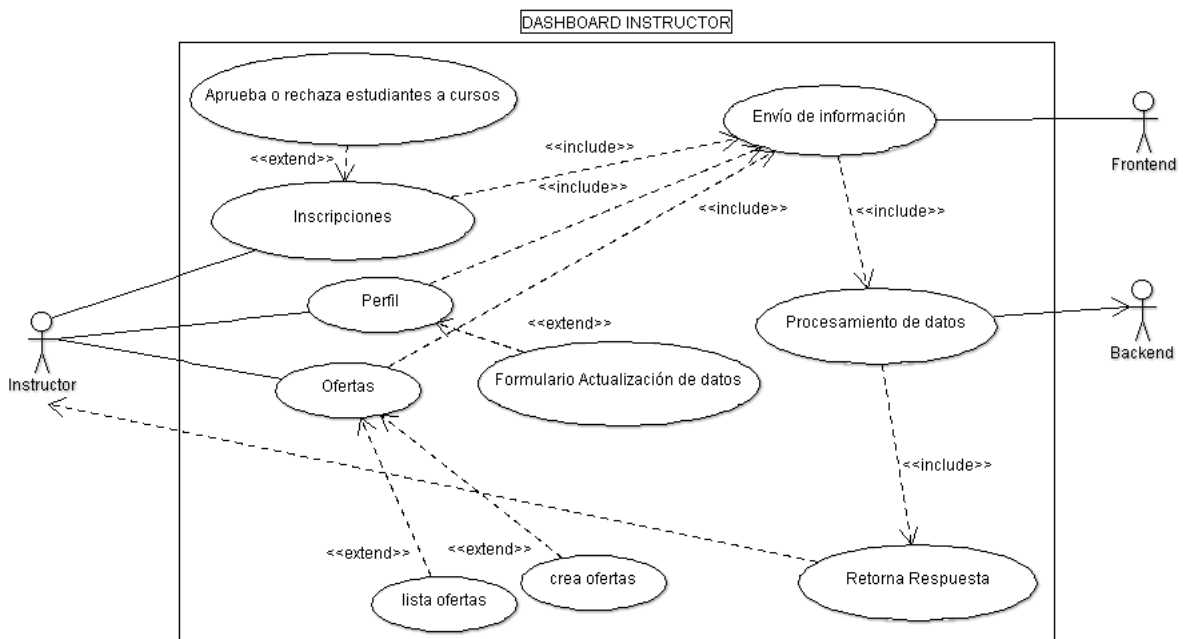


Figura 15. Diagrama UML: Dashboard instructor

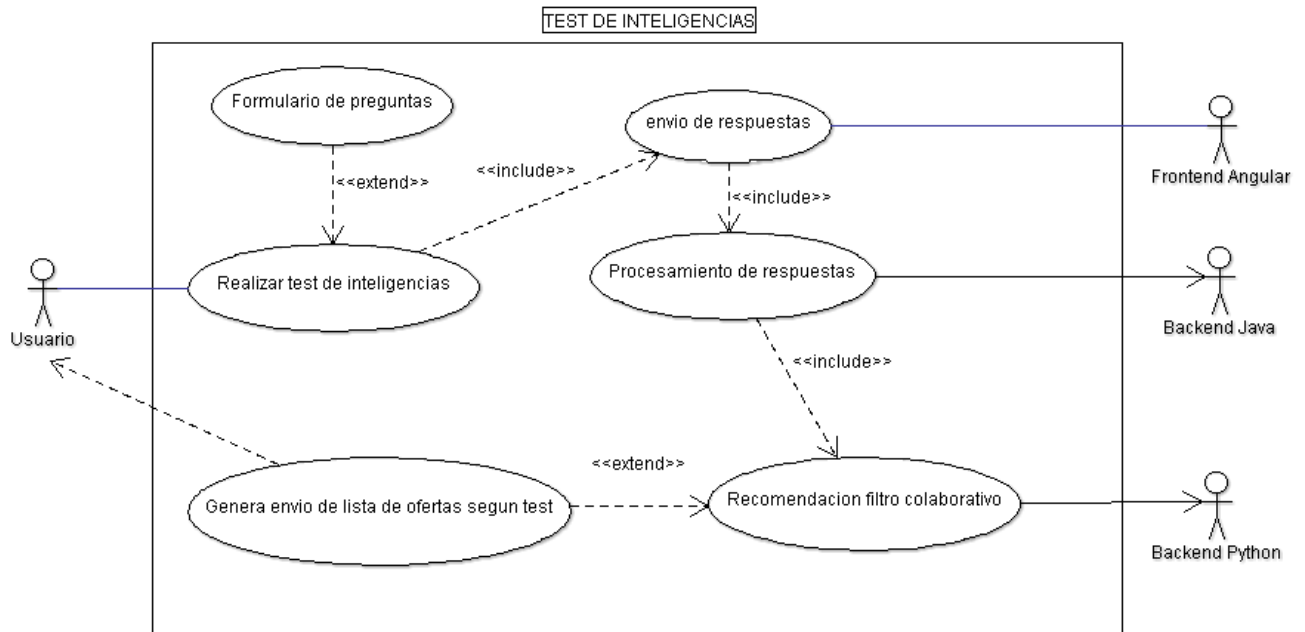


Figura 16. Diagrama UML: test de inteligencias

Modelo de base datos

Para el modelo Entidad-Relación (MER) de la base de datos, se mostrará por partes debido a que son bastantes tablas, por lo tanto, al imprimir una sola imagen con todas las entidades no se podría apreciar de manera adecuada el relacionamiento de las tablas.

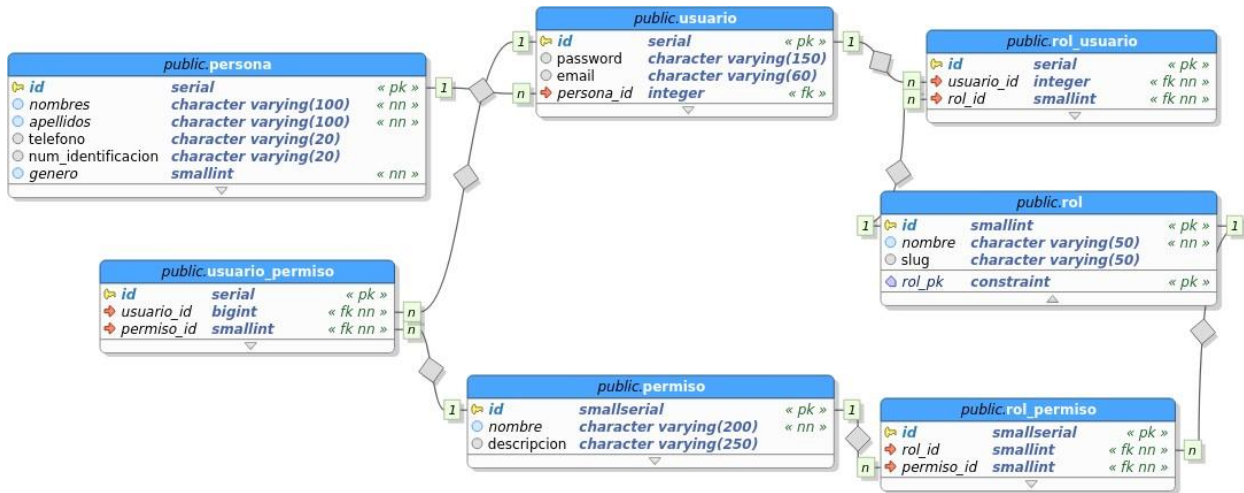


Figura 17. Modelo relacional para la seguridad

Roles y permisos:

Un rol es una colección de permisos definida para todo el sistema que se puede asignar a usuarios específicos en contextos específicos. La combinación de roles y contexto definen la habilidad de un usuario específico para hacer algo en alguna página. Los ejemplos más comunes son los roles de estudiante y maestro en el contexto de un curso.

En este diseño está contemplado varios escenarios, por ejemplo: una persona puede gozar de diversos perfiles con el mismo usuario y contraseña dentro de la aplicación, también asignar permisos especiales a determinados usuarios del sistema sin que afecte los permisos ya otorgado a los roles establecidos, entre otros.

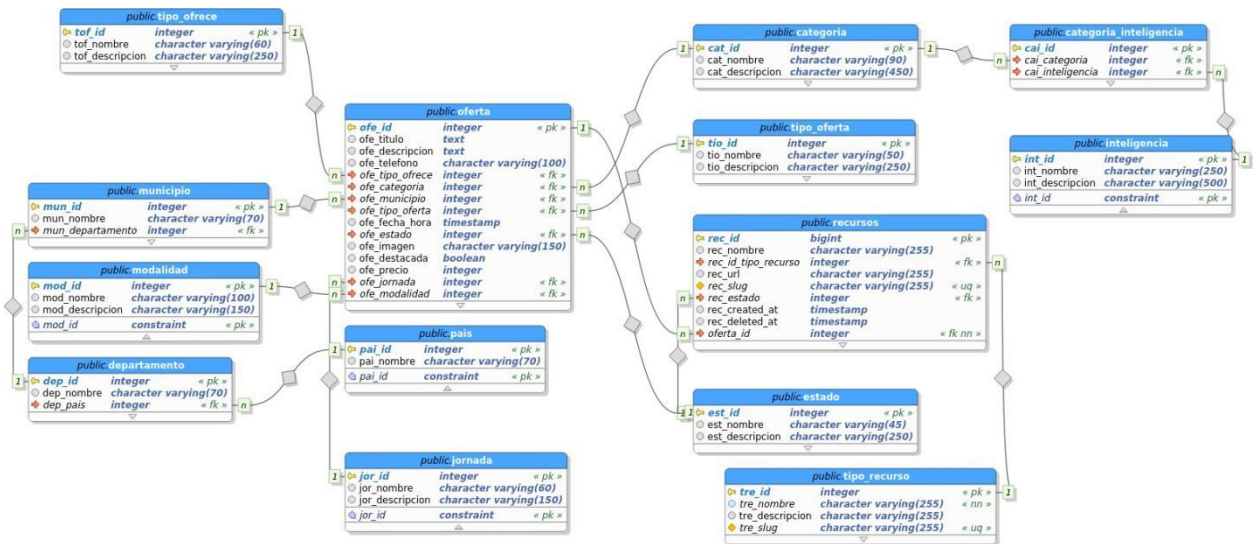


Figura 18. Modelo relacional para la administración de las ofertas

Para tener una información lo más detallada posible de cada oferta, se tomaron en cuenta varias características de las mismas, como lo son el tipo de jornada (mañana, tarde, fines de semana, etc.), la modalidad (distancia, presencial, etc.), categoría (salud, ciencias sociales, administración y finanzas, artes, etc.) considerando la posibilidad de una mejor organización de las áreas de aplicación en las ofertas. Igualmente, estas categorías se relacionan con inteligencias propuestas por Howard Gardner.

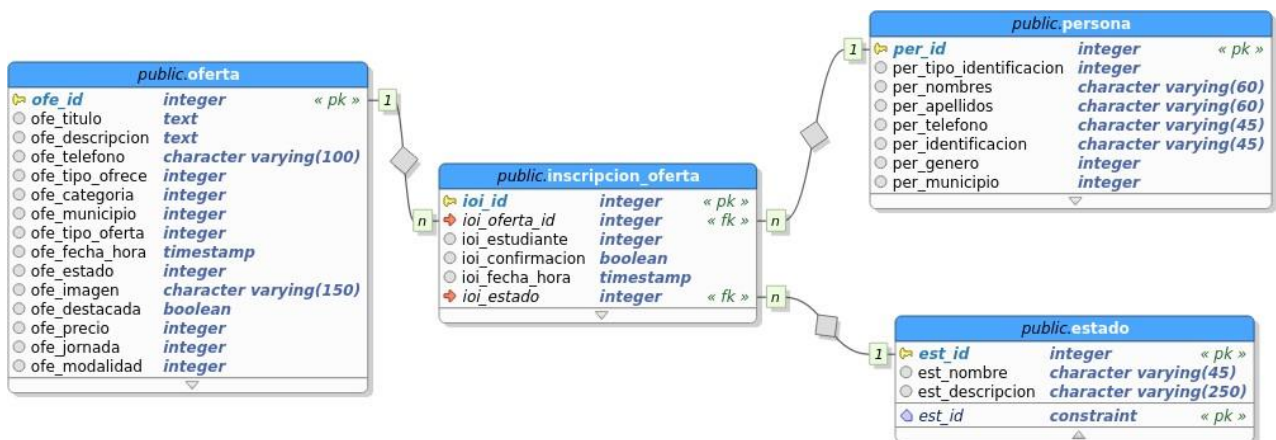


Figura 19. Modelo relacional para inscripción de ofertas

Una oferta puede contener muchas inscripciones por parte de los estudiantes, de igual manera, un estudiante puede estar inscrito en una o muchas ofertas educativas, por lo tanto, se forma una relación de muchos a muchos, entonces se debe romper esta relación, ya que habría redundancia de datos, para solucionar este problema se hace una tabla intermedia, en este caso la entidad inscripcion_oferta, figura 30.

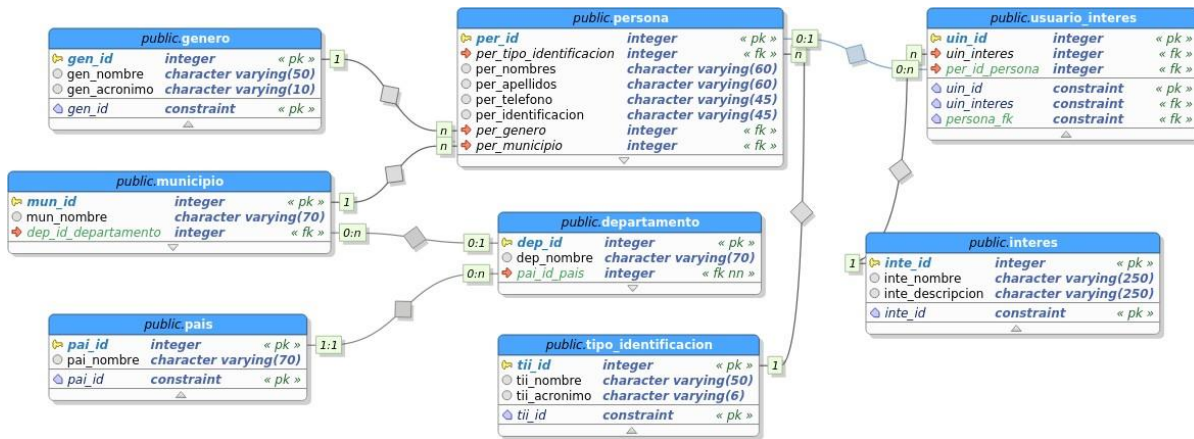


Figura 20. ER Información usuarios e intereses

Para ofrecer una mejor recomendación a los usuarios de manera individual, se capturan los intereses y/o áreas afines, con ello se crea un perfil de preferencias, el cual será enviado al modelo de recomendaciones, con ello se le podría decir: “Tal vez le pueda interesar” u “Otras personas están viendo.”

Descripción de requisitos funcionales

Diseño web

A continuación, se listarán las imágenes del diseño de la página web.



Figura 21. Página de inicio

La página de inicio será lo más sencilla posible, con solo una descripción del servicio que ofrece y muy importante, un buscador de ofertas.

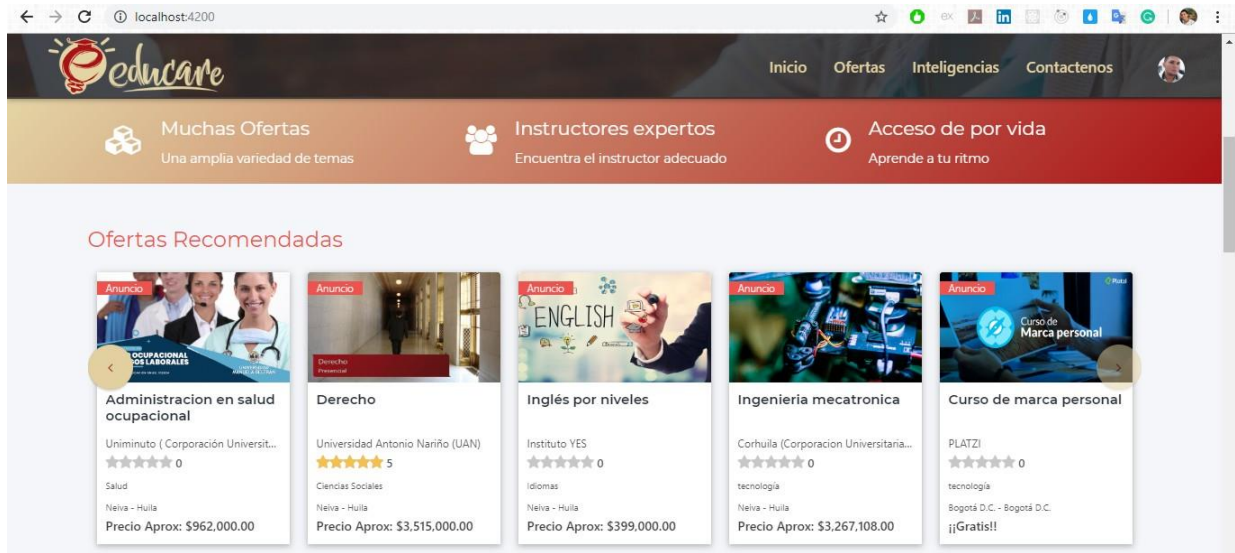


Figura 22. Ofertas destacadas(pago)

En esta sección(imagen 33) se muestran las ofertas destacadas, es decir, que los usuarios y/o instituciones han pagado para que aparezcan de primero en la página de inicio.

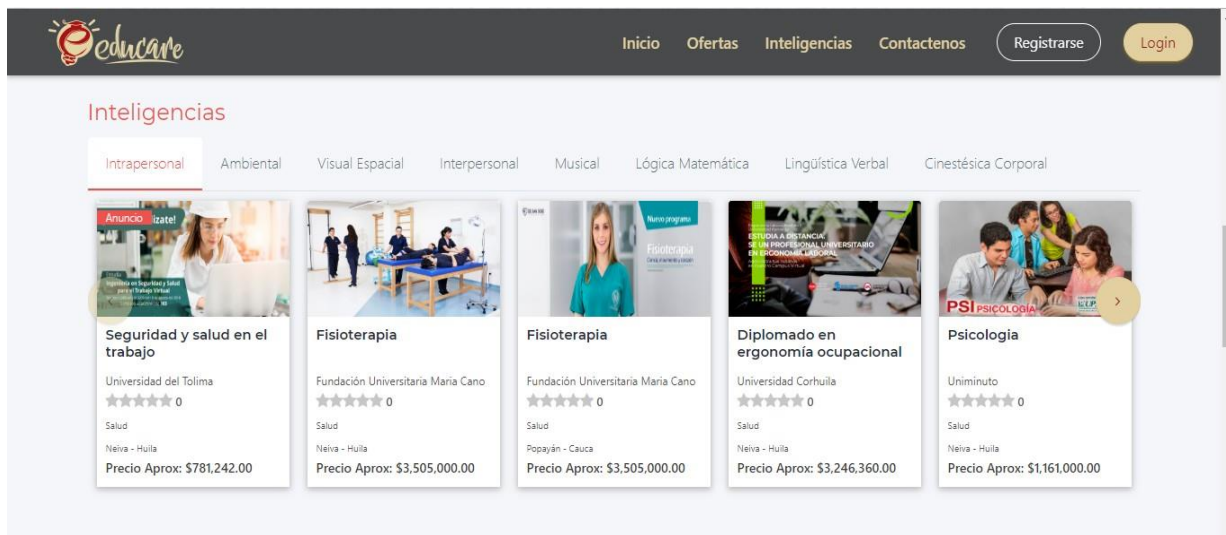


Figura 23. Ofertas clasificadas según su la inteligencia predominante

Como se aprecia en la imagen 34, se listarán algunas ofertas según su inteligencia dominante, dicho de otro modo, se dará un breve listado de las inteligencias con algunas ofertas vinculadas a ella.

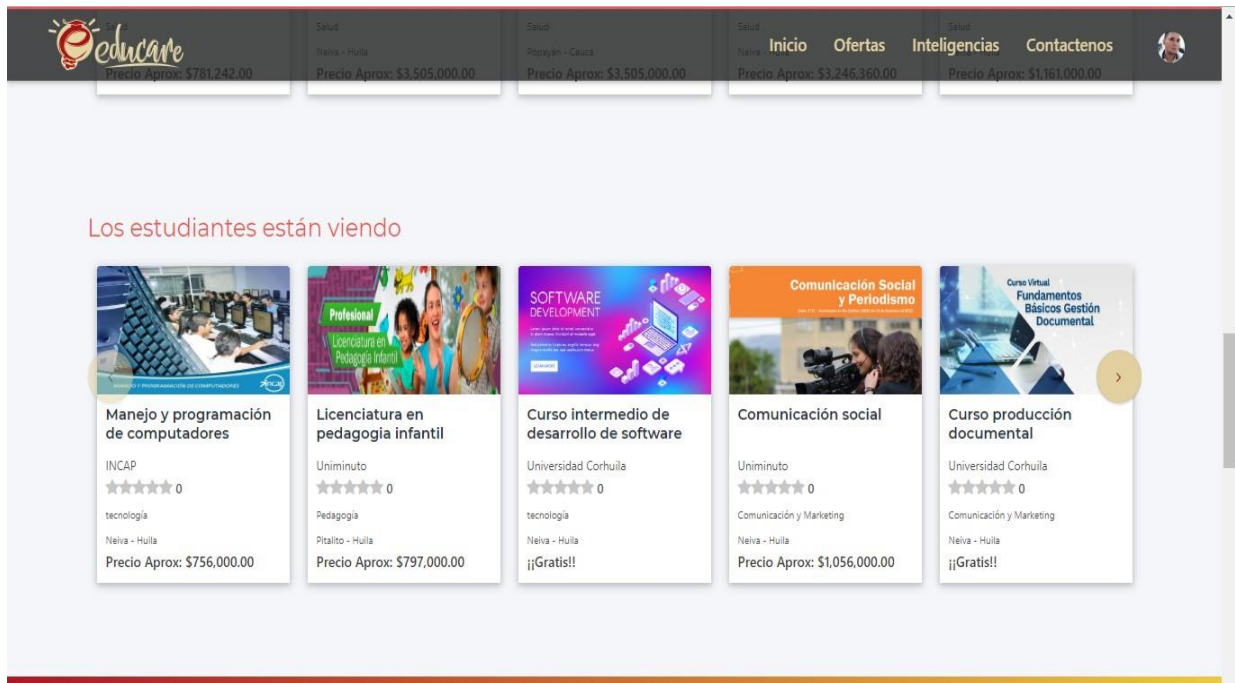


Figura 24. Ofertas más vistas por los usuarios del sistema.

Para una mejor interacción con los usuarios, se mostrará las ofertas más buscadas o vistas por los estudiantes registrado en la página esto se logra con el sistema de Inteligencia Artificial (IA) en recomendación (micro-servicio Python).

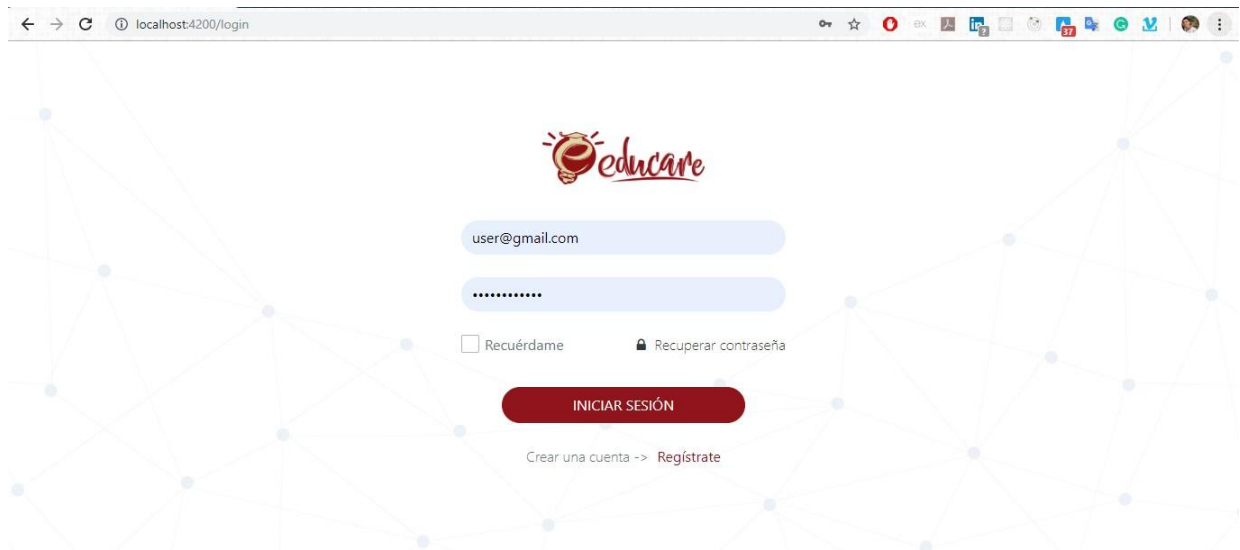


Figura 25. Formulario de login del sistema.

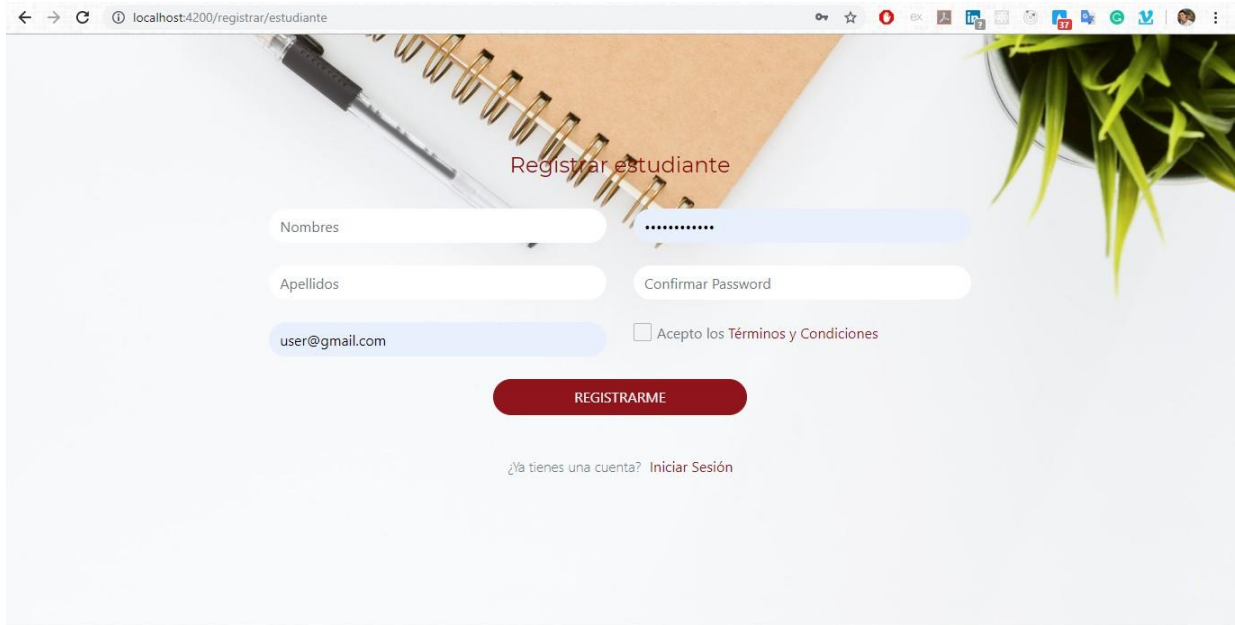


Figura 26. Formulario para el registro de usuarios.

Inicialmente, no se pedirán muchos datos, para de cierta manera, no aburrir al usuario con tantas preguntas, solo se pedirá la información más básica, más adelante, en el perfil de cada uno se le solicitará una información más detallada de cada quien.



Figura 27. Buscador de ofertas.

Como se pudo apreciar en la imagen 32, se da a disposición un pequeño formulario para buscar oferta, cuando se haya ingresado el nombre del curso, carrera y/o otro interés del usuario, se le conducirá a la página ilustrada en la imagen 38, donde se revelarán las ofertas que tienen relación con el criterio de búsqueda

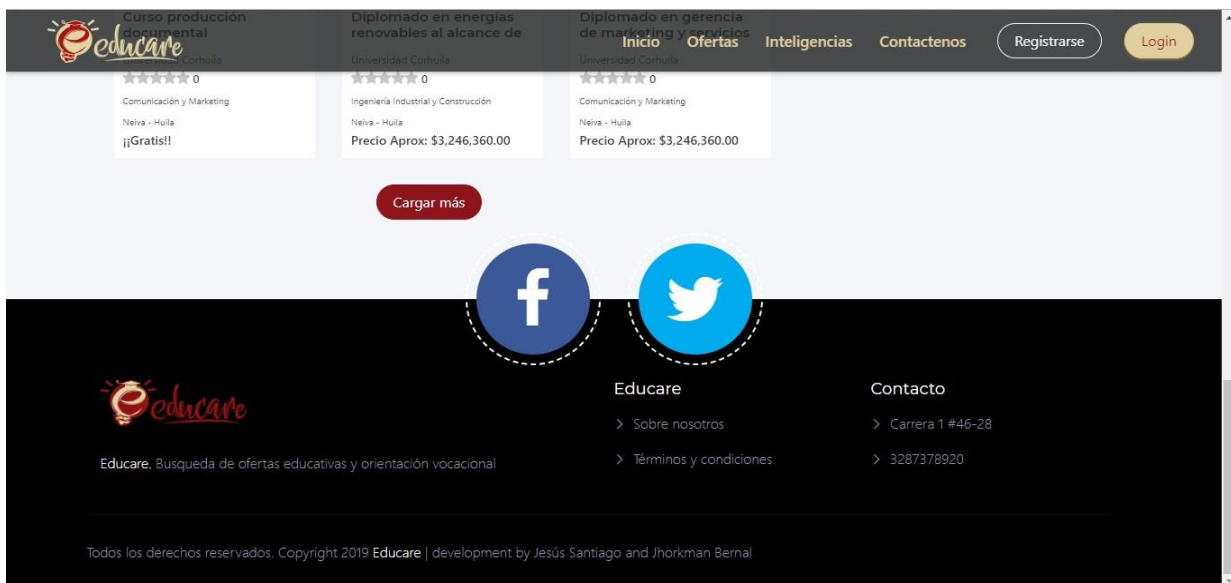


Figura 28. Pie de pagina

Se expone información básica de contacto, términos y condiciones, redes sociales y demás información de importancia para el usuario

Paneles de administración.

Pantallazos del diseño de administración de la página.

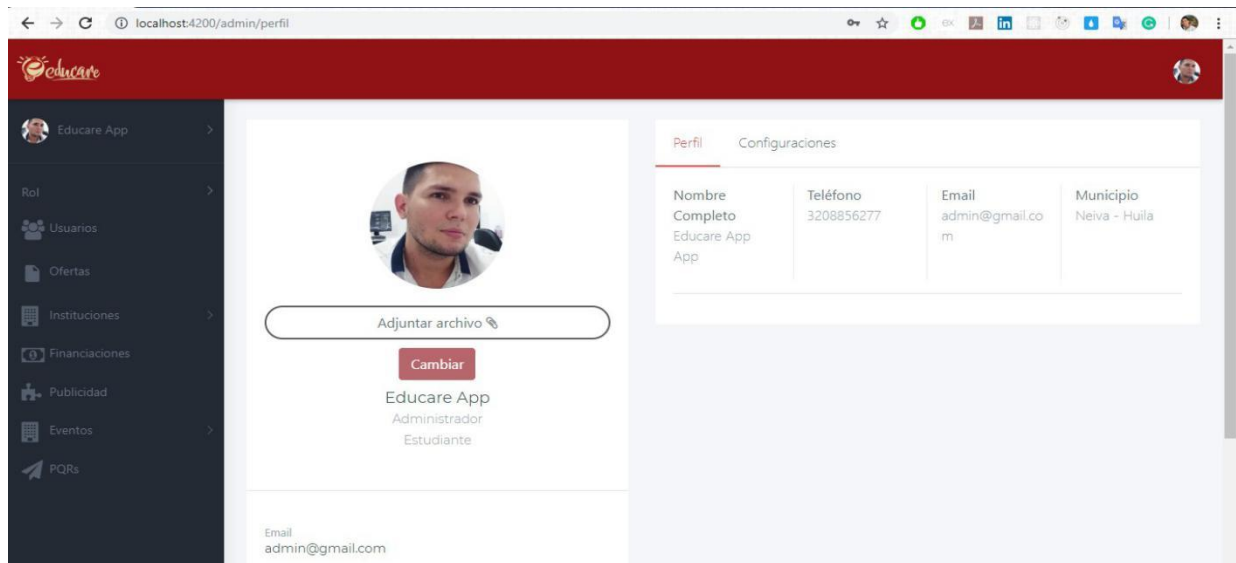


Figura 29. Página de perfil de usuario

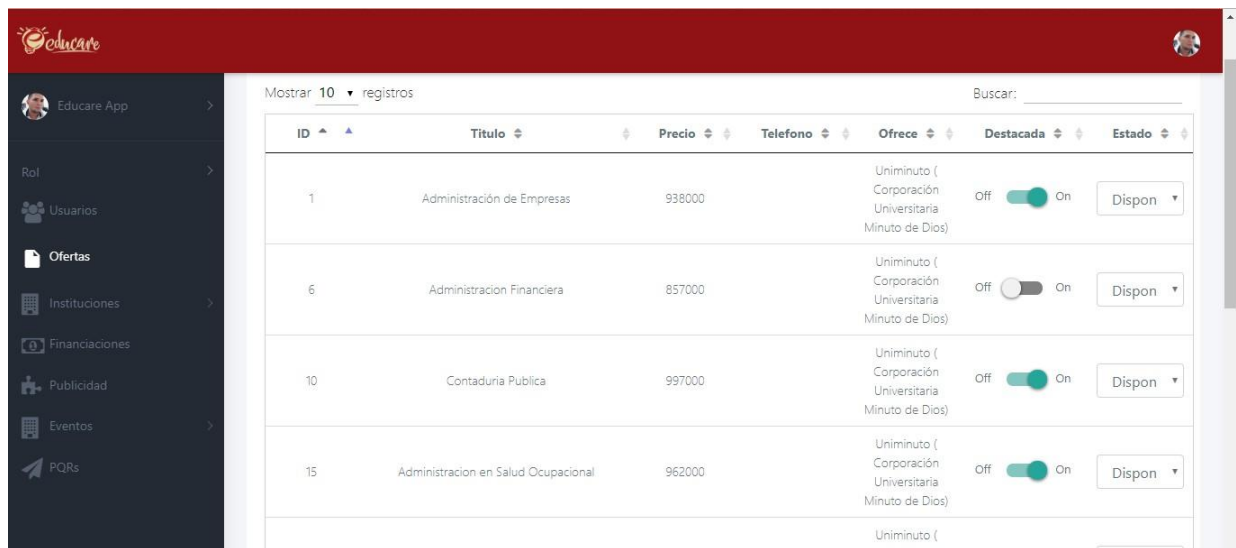


Figura 30. Gestión de ofertas

ID	Nombre	Email	Nit	Telefono	Tipo Inst.	Estado
17	UNINAVARRA	contacto@uninavarra.edu.co	900480042-2	8740089	Universidad	Activo
18	Uniminuto (Corporación Universitaria Minuto de Dios)	eliana.garcia@uniminuto.edu	800.116.217-2	8716260	Universidad	Activo
19	Universidad Antonio Nariño (UAN)	secretaria.academica.neiva@uan.edu.co	00000000-0	8770853	Universidad	Activo
21	Fundación Escuela Tecnológica de Neiva (FET)	contactenos@fet.edu.co	900440771-2	88703107 - 8603438	Universidad	Activo
24	Universidad del Tolima	atencionalciudadano@ut.edu.co	890700640-7	2772021	Universidad	Activo
25	CUN (Corporación Unificada Nacional de Educación Superior)	contacto@cun.edu.co	860401734-9	8759782-8759781	Universidad	Activo
34	Corhuila (Corporación Universitaria del Huila)	contacto@corhuila.edu.co	800107584-2	8754220	Universidad	Activo

Figura 31. Administración de instituciones

Una de las apuestas fuertes de la página web, es permitir a las instituciones de educación superior publicar las carreras universitarias, cursos, talleres, entre otros. Con ello, los usuarios tienen un gran abanico de posibilidades de las mismas ofertas en diferentes instituciones. Unas de las estrategias para monetizar el aplicativo, es la venta de publicidad, ads que se tendrán cabida en ciertas partes estratégicas de la página web.

Figura 32. Manejo de publicidad

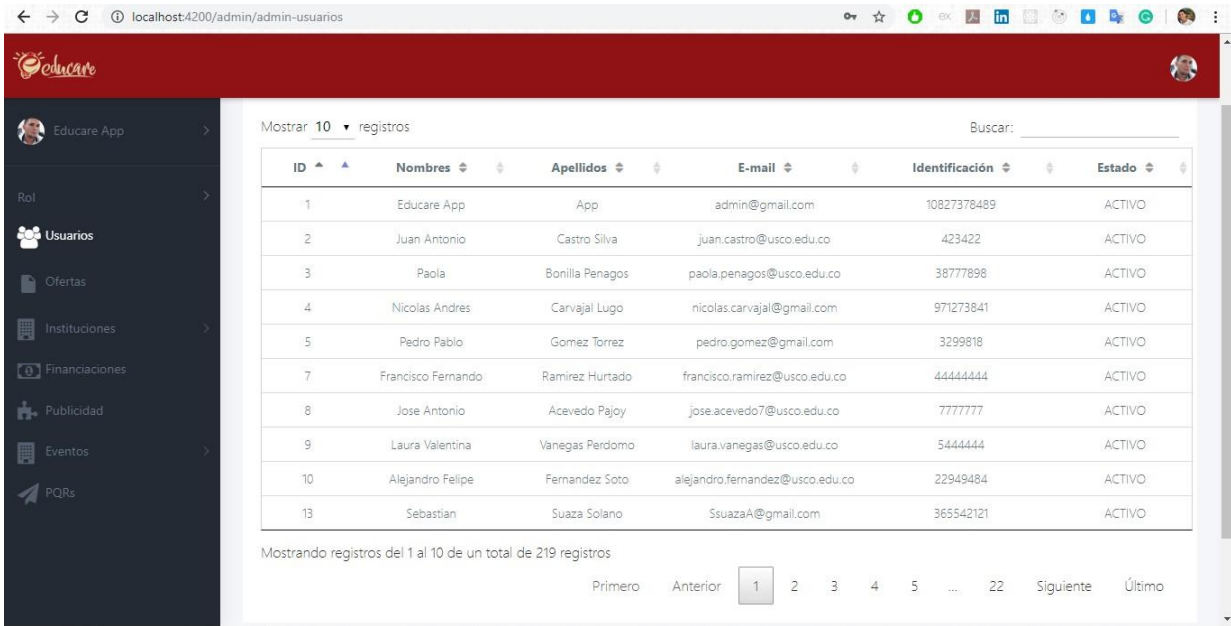


Figura 33. Administración de usuarios

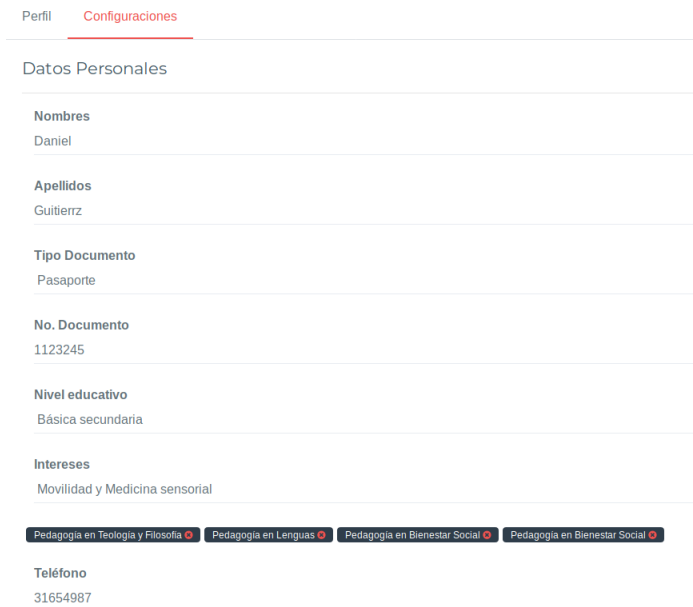


Figura 34. Formulario de datos personales.

Una vez el usuario se haya registrado, se le pedirá que llene el formulario de la imagen 45, el cual capturara toda la información necesaria, además, es esta parte se podrá agregar los intereses o preferencias en cuanto a ofertas educativas se refiere.



Figura 35. Detalle de la oferta.

Aquí se da una información más detallada de la oferta, como lo es la descripción, horarios disponibles, temario, precio y datos de contacto.



Figura 36. Diseño test de inteligencias.

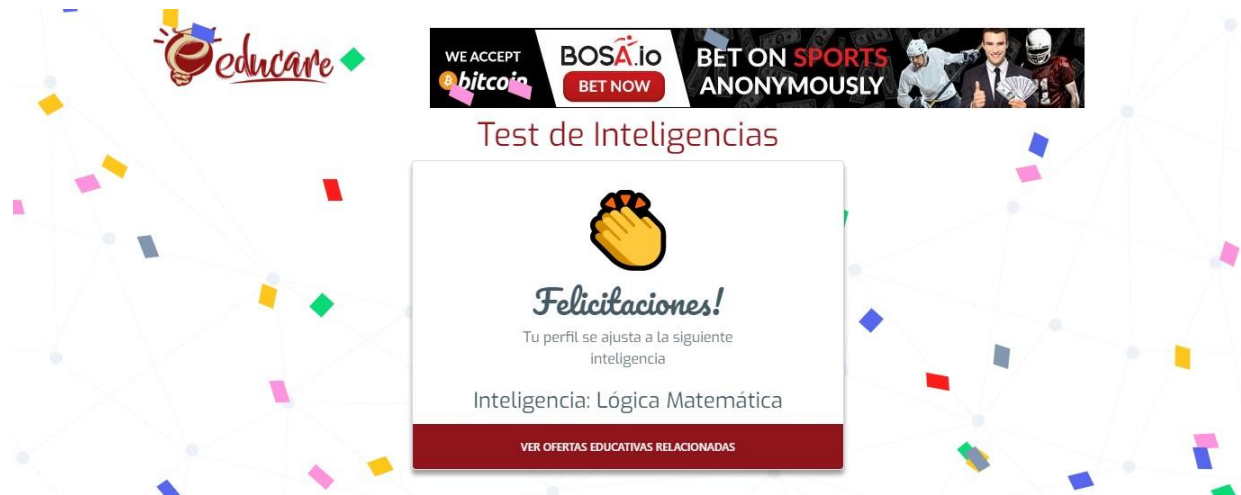


Figura 37. Fin test de inteligencias.

Una vez los usuarios hayan terminado todas las fases del test, se mostrará la inteligencia dominante, además, se muestra el botón para ver las ofertas relacionadas con dicha inteligencia.

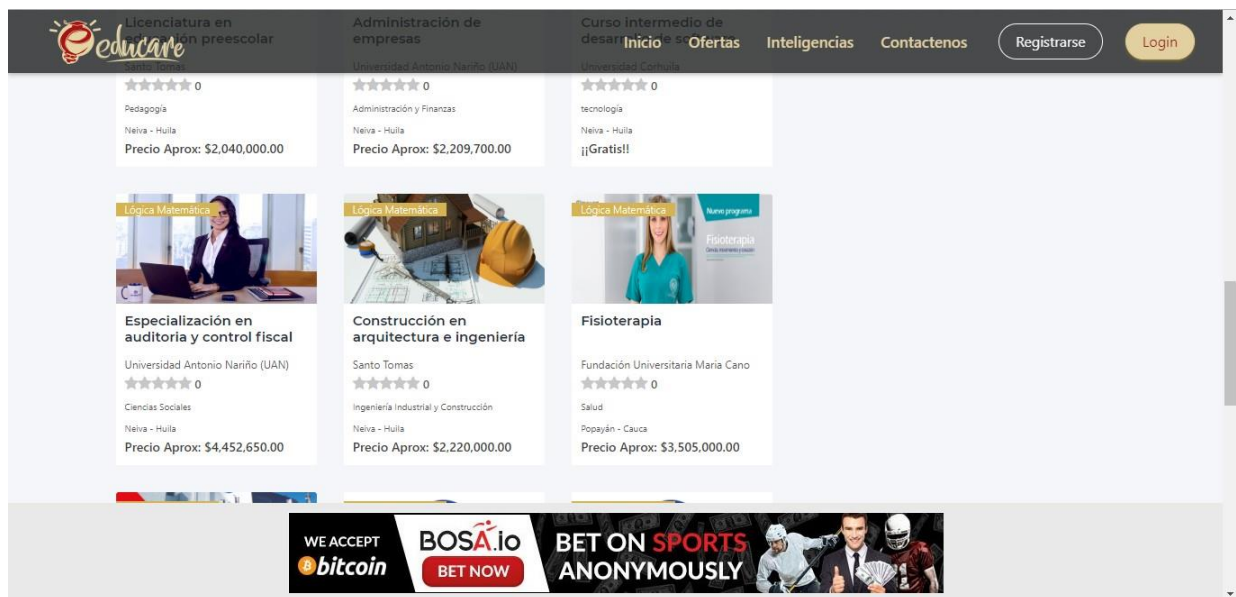


Figura 38. Resultados test de inteligencias.

Lista de ofertas relacionadas con la inteligencia que gana en el test hecho previamente por el usuario.

Implementación

Como ya se mencionó en la sección, la estructura de microservicios y su funcionamiento, además de sus ventajas, a continuación, se enseña la estructura de Educare.

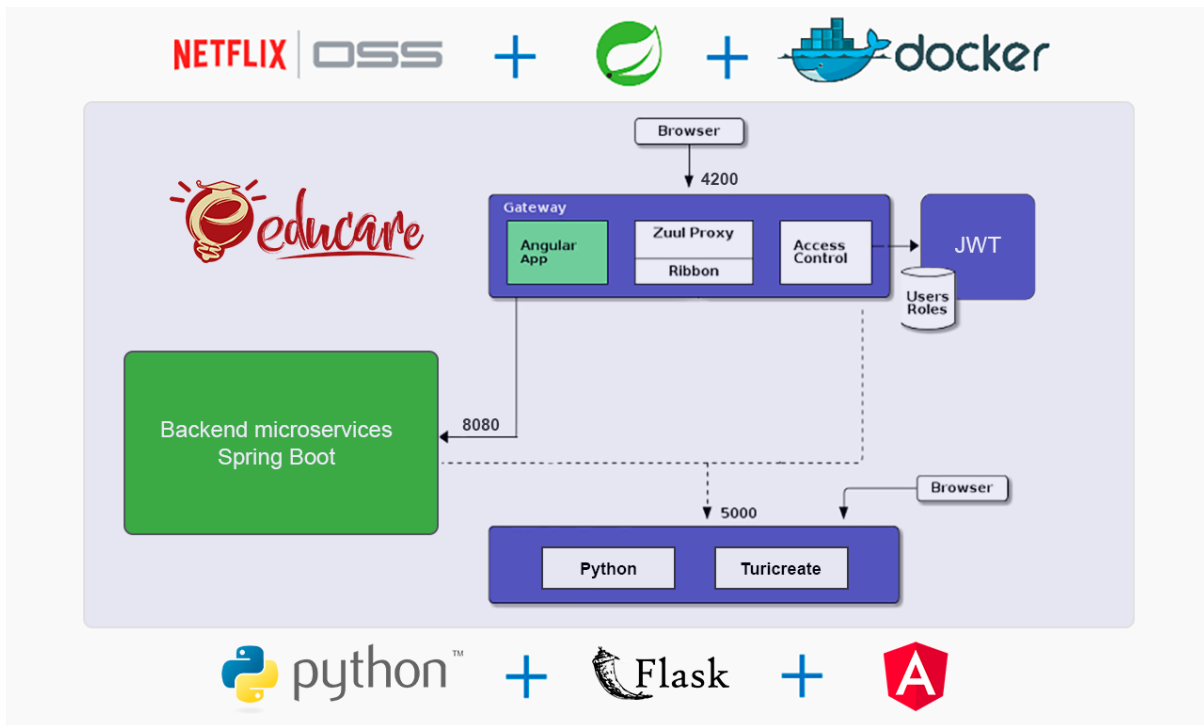


Figura 39. Estructura de despliegue de la aplicación

Desarrollo de sistema de recomendación

Construcción de modelo simple de popularidad y filtrado colaborativo usando Turicreate

Se empieza creando entrenando un modelo, el cual será invocado en el microservicio de Flask:

```
1 import turicreate as tc
2 import requests
3
4 api_url = 'http://localhost:8080/api/python-data'
5 data_request = requests.get(api_url)
6 data = data_request.json()
7
8 data = tc.SFrame(data)
9 m = tc.factorization_recommender.create(data, target='intelligence'
10 )
11 # saving model
12 m.save("recommendation.model")
13
14
```

Código 1. Archivo de entrenamiento del modelo de recomendación

En las líneas 1 y 2 se importan las librerías necesarias para la ejecución del script, en las líneas 4,5 y 6, se hace una petición al microservicio de Spring Boot, el cual dará un archivo Json con la calificación de las ofertas por los usuarios, el cual tendrá la siguiente estructura:

```
1 {
2   "user_id": [ 75, 108, 19, 34, 38, 175],
3   "item_id": [12, 45, 102, 58, 74, 61],
4   "intelligence": [4, 3, 5, 4, 2, 4]
5 }
6
```

Código 2. Objeto JSON que emite spring boot

Una vez Python consuma la información, se procede a entrenar el modelo, líneas 8 y 9, seguidamente se proceda exponer el modelo entrenado a un microservicio:

```
1 from flask import Flask, jsonify
2 import turicreate as tc
3
4 app = Flask(__name__)
5
6 m = tc.load_model("recommendation.model")
7
8
9 @app.route("/recommendations")
10 def recommendations():
11     r = m.recommend(k=8)
12     return jsonify(list(r))
13
14
15 @app.route("/user-recommendation/<id_user>")
16 def user_recommendation_by_id(id_user):
17     user_r = m.recommend(users=[id_user], k=1)
18     return jsonify(list(user_r))
19
20
21 @app.route("/similar-items")
22 def similar_items():
23     user_r = m.get_similar_items(k=10)
24     return jsonify(list(user_r))
25
26
27 @app.route("/")
28 def hello_world():
29     return "Hello World!"
30
31 if __name__ == "__main__":
32     app.run()
33
34
```

Código 3. Configuración Flask para el microservicio de recomendación

Los datos de salida en formato JSON de las APIs sería de la siguiente manera:

```
1  [
2  {
3    "item_id": 58,
4    "rank ": 1,
5    "score": 5.929876766204833,
6    "user_id": 130
7  },
8  {
9    "item_id": 76,
10   "rank ": 2,
11   "score": 5.703564367294311,
12   "user_id": 130
13  },
14  {
15   "item_id": 36,
16   "rank ": 3,
17   "score": 5.585953316688537,
18   "user_id": 130
19  },
20  {
21   "item_id": 1,
22   "rank ": 4,
23   "score": 5.361506900787353,
24   "user_id": 130
25  },
26  {
27   "item_id": 195,
28   "rank ": 5,
29   "score": 5.277116498947143,
30   "user_id": 130
31  }
32 ]
33
```

Código 4. Objeto JSON de ofertas recomendadas

Cuando el microservicio de Python haya hecho la recomendación, el servicio de spring boot tomará el objeto JSON y enviará al frontend las ofertas recién recomendadas. Un ejemplo sería:

```
1  {
2    "exito": true,
3    "codigo": 200,
4    "data ": [
```

```
5 {
6   "id": 58,
7   "titulo": "Tecnología en gestión de operaciones industriales",
8   "descripcion": null,
9   "precio": "2209700",
10  "telefono": null,
11  "tipoOfrece": {
12    "id": 1,
13    "nombre": "Institucion",
14    "descripcion": "Institucion"
15  },
16  "categoria": {
17    "id": 7,
18    "nombre": "Ingeniería Industrial y Construcción",
19    "descripcion": "Ingeniería Industrial y Construcción"
20  },
21  "municipio": {
22    "id": 1,
23    "nombre": "Neiva",
24    "departamento": {
25      "id": 1,
26      "nombre": "Huila",
27      "pais": {
28        "id": 1,
29        "nombre": "Colombia"
30      }
31    }
32  },
33  "tipoOferta": {
34    "id": 4,
35    "nombre": "Carrera Tecnológica",
36    "descripcion": "CARRERA TECNOLÓGICA"
37  },
38  ...
39 }
40
```

Para el despliegue del proyecto se optará por una solución cloud, cuyas ventajas dan gran avance en el despliegue de la plataforma.

Cloud

La computación Cloud, en español, en nube, es la entrega de servicios alojados a través de Internet, en lugar de en casa, (sus propias instalaciones). Las razones para elegir una solución en la nube son las siguientes.

Flexibilidad.

Los servicios basados en la nube son ideales para negocios con demandas de ancho de banda cada vez mayores o fluctuantes. De esta forma las capacidades se pueden aumentar o reducir en un instante según la necesidad.

Recuperación de desastres.

La computación en la nube ayuda a organizaciones pequeñas a enfrentar este problema al brindar soluciones rápidas y sencillas que no requieren gran conocimiento técnico.

Actualizaciones automáticas de software.

Los proveedores se ocupan de actualizaciones de software regulares, incluidas actualizaciones de seguridad, así no se pierde tiempo haciendo.

Gastos de capital.

La computación en la nube reduce el alto costo del hardware. Simplemente se paga a la medida que se requiera.

Mayor colaboración.

Los equipos de desarrollo pueden acceder, editar y compartir documentos en cualquier momento y desde cualquier lugar.

Flexibilidad laboral.

Las empresas pueden ofrecer ventajas de trabajo más flexibles a los empleados para que puedan disfrutar del equilibrio entre el trabajo y la vida que les conviene.

Control de documentos.

Con un servicio en la nube los empleados comparten las mismas versiones de archivo y no existe el problema de intercambio físico de documentos.

Seguridad.

Debido a que los datos se almacenan en la nube, se puede acceder a los documentos sin importar lo que pueda suceder a las máquinas físicas de trabajo.

Competitividad.

El traslado a la nube da acceso a tecnología de clase empresarial para todos, permite que las pequeñas empresas actúen igual o más rápido que los competidores grandes y establecidos.

Tecnologías utilizadas

Para el desarrollo del proyecto, se hace uso de las siguientes tecnologías.

*Tabla 9.
Tecnologías a utilizar*

Lenguajes de programación web	Framework web
- Java EE	- Bootstrap
- TypeScript	- Angular
Bases de datos	- JQuery
- SQL	Servicio web
- Python 3.6	- AJAX
Motor de bases de datos	- JSon

- PostgreSQL
 - REST
 - Mapeo Objeto-Relacional**
 - Google Maps API
 - Hibernate - JPA
 - Servidor de aplicaciones Java EE**
 - Plataforma de servicios Cloud**
 - Apache TomCat
 - AWS
 - OpenShift (opcional)
-

Cronograma

Este proyecto se desarrolló en un tiempo aproximado de 149 días, comprendidos entre el 2 de noviembre del 2018, y el 5 29 de mayo del 2019, con un total de 2468 horas de trabajo y un costo total de \$36'581.157,32.

A continuación, se presenta las fases, cronograma y detalle de actividades realizadas durante el desarrollo del proyecto, comprendidas en planificación, análisis, desarrollo, implementación, pruebas y cierre.

Tabla 10 Principales actividades realizadas

Semana Proceso	2018								2019																							
	Noviembre				Diciembre				Enero				Febrero				Marzo				Abril				Mayo							
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
Planificación	■	■																														
Análisis información		■	■	■																												
Definición de requisitos			■	■																												
Diseño ER base datos					■	■																										
Diseño arquitectura del sistema					■	■	■																									
Desarrollo									■	■	■	■	■	■	■	■	■	■	■	■												

10	Requisitos Funcionales	2.667 días	Documentador; Analista de sistemas; Gerente de proyecto	8	11/21/2018 15:00	11/26/2018 11:20
11	Requisitos no funcionales	2 días	Documentador; Analista de sistemas	8	11/21/2018 15:00	11/23/2018 15:00
12	Aprobación de requisitos	0 días	Gerente de proyecto; Analista de sistemas	10FF; 11	11/26/2018 11:20	11/26/2018 11:20
13	Ajuste requisitos	0.5 días	Documentador; Analista de sistemas	12	11/26/2018 11:20	11/26/2018 16:20
14	Aprobación requisitos finales	0 días	Gerente de proyecto	13FF	11/26/2018 16:20	11/26/2018 16:20
15	Diseño ER base datos	5 días			11/26/2018 16:20	12/3/2018 16:20
16	Diseño preliminar DB	4 días	Experto Base Datos	14	11/26/2018 16:20	11/30/2018 16:20
17	Aprobación ER DB	0 días	Gerente de proyecto; Analista de sistemas	16FF	11/30/2018 16:20	11/30/2018 16:20
18	Ajuste ER DB	1 día	Experto Base Datos	17	11/30/2018 16:20	12/3/2018 16:20
19	Aprobación final ER DB	0 días	Gerente de proyecto	18FF	12/3/2018 16:20	12/3/2018 16:20
20	Diseño arquitectura del sistema	18.917 días			11/2/2018 8:00	11/28/2018 16:20
21	Definición de tecnologías a utilizar	18.917 días			11/2/2018 8:00	11/28/2018 16:20
22	Diseño arquitectura Backend	1 día	Analista de sistemas	9	11/26/2018 16:20	11/27/2018 16:20
23	Diseño arquitectura Frontend	1 día	Analista de sistemas	9; 22	11/27/2018 16:20	11/28/2018 16:20

24	Diseño arquitectura Sistema de recomendación	1 día	Analista de sistemas	9	11/26/2018 16:20	11/27/2018 16:20
25	DBMS	1 día	Experto Base Datos	9; 22	11/27/2018 16:20	11/28/2018 16:20
26	Arquitectura, monolítica vs microservicios	1 día	Analista de sistemas	9	11/26/2018 16:20	11/27/2018 16:20
27	Aprobación tecnologías	0 días	Analista de sistemas		11/2/2018 8:00	11/2/2018 8:00
28	Desarrollo	103.167 días			11/23/2018 15:00	4/17/2019 16:20
29	Preparación entorno de desarrollo	7.167 días			11/23/2018 15:00	12/4/2018 16:20
30	Instalación de Node JS y NPM	0.5 días	DevOps	23	11/28/2018 16:20	11/29/2018 11:20
31	Instalación JDK y Maven	1 días	DevOps	22	11/27/2018 16:20	11/28/2018 16:20
32	Instalación PostgreSQL	1 días	DevOps; VPS Digital Oceans	19	12/3/2018 16:20	12/4/2018 16:20
33	Instalación Docker	1 días	DevOps; VPS Digital Oceans	11	11/23/2018 15:00	11/26/2018 15:00
34	Configuración de contenedores Docker	1 day	DevOps; VPS Digital Oceans	33	11/26/2018 15:00	11/27/2018 15:00
35	Desarrollo Backend	57 días			11/27/2018 16:20	2/14/2019 16:20
36	Creación de API's publicas	40 días	Programador Backend	22	11/27/2018 16:20	1/22/2019 16:20
37	Desarrollo API test de inteligencia	10 días	Programador Backend	22	11/27/2018 16:20	12/11/2018 16:20
38	Configuración seguridad	5 días	Programador Backend	22	11/29/2018 11:00	12/6/2018 11:00
39	Desarrollo API's estudiante	5 días	Programador Backend	22	12/3/2018 8:00	12/7/2018 17:00
40	Desarrollo API's instructor	5 días	Programador Backend	22	12/4/2018 8:00	12/10/2018 17:00
41	Desarrollo API's instituciones	5 días	Programador Backend	22	12/6/2018 8:00	12/12/2018 17:00

42	Desarrollo API's administrador	10 días	Programador Backend	36; 37; 38; 39; 40; 41	1/22/2019 16:20	2/5/2019 16:20
43	Pruebas Backend	7 días	Auditor; Programador Backend; Analista de sistemas	42	2/5/2019 16:20	2/14/2019 16:20
44	Diseño sistema de recomendación	28 días			2/14/2019 16:20	3/26/2019 16:20
45	Creación modelo de recomendación	10 días	Programador IA	35	2/14/2019 16:20	2/28/2019 16:20
46	Entrenamiento Modelo recomendación	10 días	Programador IA	45	2/28/2019 16:20	3/14/2019 16:20
47	Train Test Split	5 días	Programador IA	46	3/14/2019 16:20	3/21/2019 16:20
48	Creación API REST	3 días	Programador IA	47	3/21/2019 16:20	3/26/2019 16:20
49	Desarrollo Frontend	87 días			11/28/2018 16:20	3/29/2019 16:20
50	Diseño UI	20 días	Diseñador Gráfico; Programador FrontEnd	23	11/28/2018 16:20	12/26/2018 16:20
51	Desarrollo Landing Page	2 días	Programador FrontEnd	36	1/22/2019 16:20	1/24/2019 16:20
52	Desarrollo UI test inteligencias	3 días	Programador FrontEnd	37; 44	3/26/2019 16:20	3/29/2019 16:20
53	Panel administración estudiante	5 días	Programador FrontEnd	39	12/10/2018 8:00	12/14/2018 17:00
54	Panel administración instructor	3 días	Programador FrontEnd	40	12/11/2018 8:00	12/13/2018 17:00
55	Panel administración institución	3 días	Programador FrontEnd	41	12/13/2018 8:00	12/17/2018 17:00
56	Panel administración administrador	8 días	Programador FrontEnd	42	2/5/2019 16:20	2/15/2019 16:20

57	Pruebas FrontEnd	3 días	Programador FrontEnd; Auditor	56	2/15/2019 16:20	2/20/2019 16:20
58	Pruebas	3 días	Auditor; VPS Digital Oceans	35; 49	3/29/2019 16:20	4/3/2019 16:20
59	Ajustes	10 día	Programador Backend; Programador FrontEnd	58	4/3/2019 16:20	4/17/2019 16:20
60	Versión final	0 días	Gerente de proyecto	59FF	4/17/2019 16:20	4/17/2019 16:20
61	Implementación	30 días			4/17/2019 16:20	5/29/2019 16:20
62	Instalación	3 días	DevOps; VPS Digital Oceans	60	4/17/2019 16:20	4/22/2019 16:20
63	Documentación	30 días	Documentador	60	4/17/2019 16:20	5/29/2019 16:20
64	Fin proyecto	0 días		63FF	5/29/2019 16:20	5/29/2019 16:20

Para una mejor gestión del cronograma, recursos y presupuesto, se apoyó en un software especializado para la administración de proyectos, en este caso fue ProjectLibre

El cronograma del proyecto se elaboró con las 5 fases esenciales para la gestión de proyectos del PMBOK:

- Inicio
- Planificación
- Ejecución
- Supervisión y control
- Cierre

Presupuesto

Tabla 11.

Presupuesto

Nombre	Trabajo
Diseñador Grafico	168 horas
Diseño UI	160 horas
Programador Backend	776 horas
Pruebas Backend	56 horas
Ajustes	80 horas
Creación de API's publicas	320 horas
Desarrollo API's administrador	80 horas
Desarrollo API's instituciones	40 horas
Desarrollo API test de inteligencia	80 horas
Desarrollo API's instructor	40 horas
Configuración seguridad	40 horas
Desarrollo API's estudiante	40 horas
Programador FrontEnd	456 horas
Diseño UI	160 horas
Panel administración instructor	24 horas
Ajustes	80 horas
Panel administración estudiante	40 horas
Panel administración administrador	64 horas
Pruebas FrontEnd	24 horas
Desarrollo Landing Page	16 horas
Desarrollo UI test inteligencias	24 horas
Panel administración intitución	24 horas
Experto Base Datos	48 horas
Ajuste ER DB	8 horas
Diseño preliminar DB	32 horas
DBMS	8 horas
Analista de sistemas	221.333 horas
Requisitos no funcionales	16 horas
Recopilación de información	60 horas
Arquitectura, monolitica vs microservicios	8 horas
Diseño arquitectura Frontend	8 horas
Análisis y diseño	28 horas
Diseño arquitectura Backend	8 horas
Alcance	4 horas
Diseño arquitectura Sistema de recomendación	8 horas

Aprovación tecnologías	0 horas
Aprovación ER DB	0 horas
Pruebas Backend	56 horas
Requisitos Funcionales	21.333 horas
Ajuste requisitos	4 horas
Aprovación de requisitos	0 horas
Documentador	369.333 horas
Requisitos Funcionales	21.333 horas
Recopilación de información	60 horas
Ajuste requisitos	4 horas
Documentación	240 horas
Requisitos no funcionales	16 horas
Análisis y diseño	28 horas
VPS Digital Oceans	0 horas
Pruebas	1
Configuración de contenedores Docker	1
Instalación Docker	1
Instalación	1
Instalación PostgreSQL	1
Gerente de proyecto	41.333 horas
Alcance	4 horas
Aprovación de requisitos	0 horas
Aprovación final ER DB	0 horas
Aprovación ER DB	0 horas
Requisitos Funcionales	21.333 horas
Aprovación requisitos finales	0 horas
Plan de negocio	8 horas
Acta de inicio	8 horas
Versión final	0 horas
Auditor	104 horas
Pruebas Backend	56 horas
Pruebas FrontEnd	24 horas
Pruebas	24 horas
DevOps	60 horas
Instalación Docker	8 horas
Instalación PostgreSQL	8 horas
Instalación de Node JS y NPM	4 horas
Configuración de contenedores Docker	8 horas

Instalación	24 horas
Instalación JDK y Maven	8 horas
Programador IA	224 horas
Train Test Split	40 horas
Entrenamiento Modelo recomendación	80 horas
Creación API REST	24 horas
Creación modelo de recomendación	80 horas

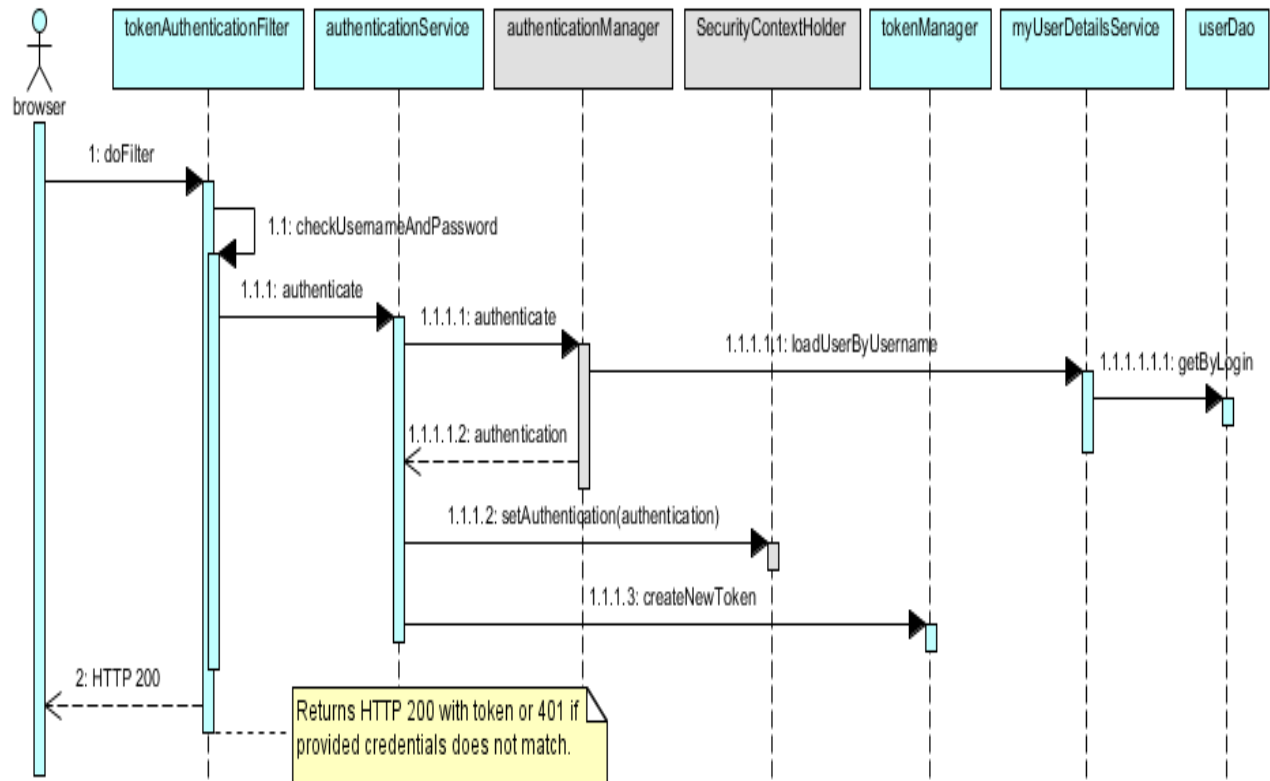
Conclusiones

Educare aporta una integración de diversas tecnologías para realizar recomendaciones a la comunidad educativa dependiendo de su perfil cognitivo, permitiéndole al estudiante escoger con precisión mejores recursos educativos. Además, cuenta con un banco de ofertas donde universidades e institutos de cualquier país pueden brindar cursos y conocimiento a través de esta plataforma de múltiples oferentes.

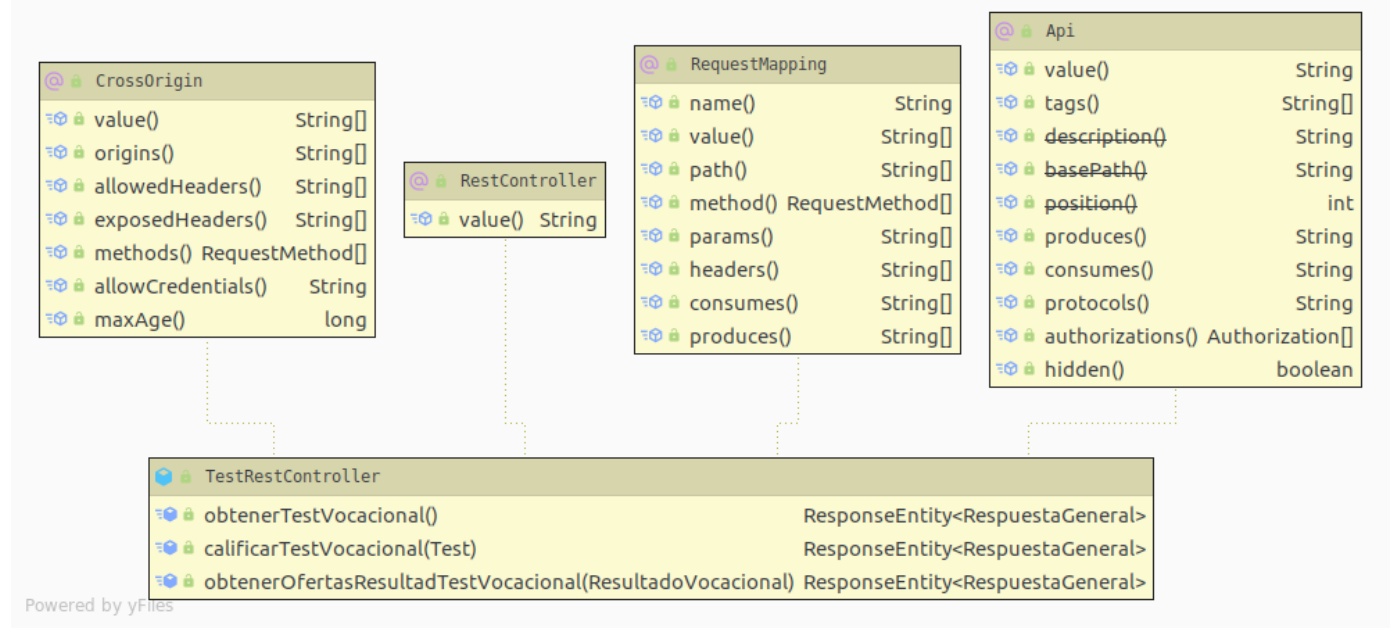
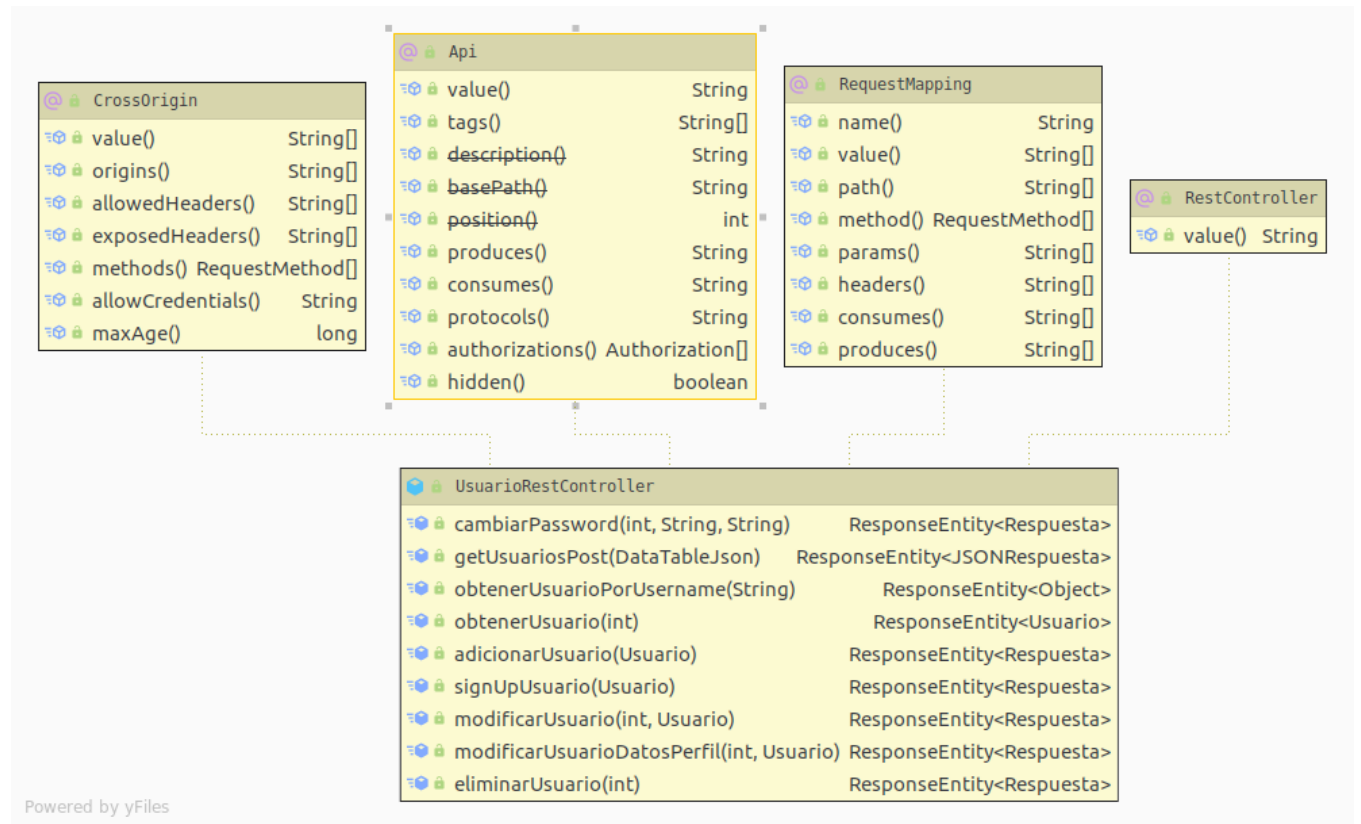
El uso de la inteligencia artificial ha dado resultados favorables en las diferentes áreas donde se ha implementado. Algoritmos de machine Learning y Deep Learning cada día son más capaces de realizar predicciones, recomendaciones basada en tendencias y/o preferencias de cada persona. Este tipo de tecnología beneficiara en gran manera al sector educativo, permitiendo evaluar con un test de inteligencias la orientación académica de cada usuario, así evitando en un futuro la pérdida de tiempo y dinero por una mala elección.

Los micro-servicios cuentan con numerosas ventajas que facilitan los procesos a la hora de utilización de un sistema. No obstante, al ser novedoso una gran mayoría de CIOs se verán frenados a implantar estas tecnologías en su empresa por la complejidad que presentan algunas funcionalidades. Sin embargo, la implantación de los micro-servicios podría repercutir con grandes resultados y de una manera muy positiva a grandes empresas tales como Amazon o Netflix que ya han implantado esta tecnología. Este proyecto cuenta con un factor diferenciador al aplicar las últimas tecnologías del mercado, así como las buenas prácticas de desarrollo de software.

Anexos



PROYECTO DE GRADO



PROYECTO DE GRADO

Api	
value()	String
tags()	String[]
description()	String
basePath()	String
position()	int
produces()	String
consumes()	String
protocols()	String
authorizations()	Authorization[]
hidden()	boolean

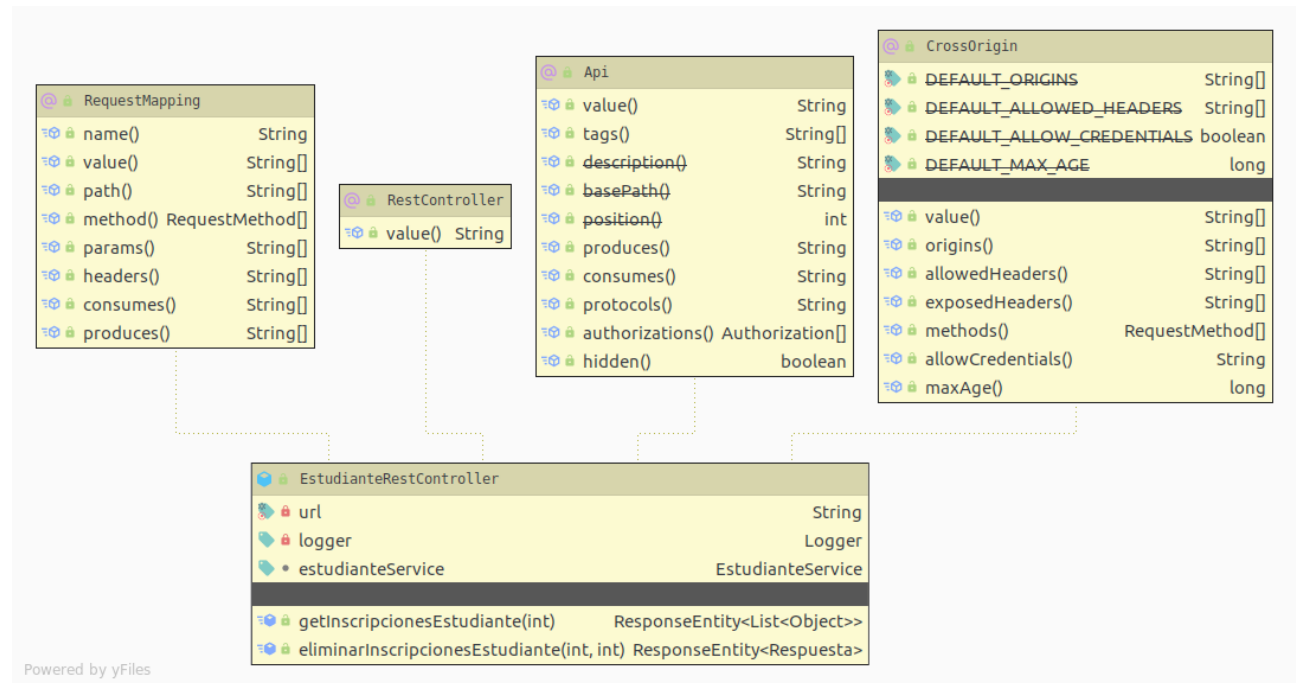
RequestMapping	
name()	String
value()	String[]
path()	String[]
method()	RequestMethod[]
params()	String[]
headers()	String[]
consumes()	String[]
produces()	String[]

RestController	
value()	String

CrossOrigin	
DEFAULT_ORIGINS	String[]
DEFAULT_ALLOWED_HEADERS	String[]
DEFAULT_ALLOW_CREDENTIALS	boolean
DEFAULT_MAX_AGE	long
<hr/>	
value()	String[]
origins()	String[]
allowedHeaders()	String[]
exposedHeaders()	String[]
methods()	RequestMethod[]
allowCredentials()	String
maxAge()	long

OfertaRestController	
url	String
logger	Logger
ofertaService	OfertaService
recommendationService	RecommendationService
<hr/>	
OfertaRestController(OfertaService, RecommendationService)	
<hr/>	
obtenerOferta(int)	ResponseEntity<RespuestaGeneral>
eliminarOfertaFreelancer(int, int)	ResponseEntity<Respuesta>
deleteInstitutionOffer(int, int)	ResponseEntity<Respuesta>
getOfertasBuscador(String, int, int, int, int, String)	ResponseEntity<JSONRespuesta>
getOfertasByTitulo(String, int, int)	ResponseEntity<JSONRespuesta>
getOfertasFiltros(int, int, int, int, String, int, int, int, int, int, int, int, String)	ResponseEntity<JSONRespuesta>
getOfertas(String, int, int, int, int, String)	ResponseEntity<JSONRespuesta>
getOfertasPost(DataTableJson)	ResponseEntity<JSONRespuesta>
adicionarOferta(Oferta)	ResponseEntity<Respuesta>
modificarOferta(int, Oferta)	ResponseEntity<Respuesta>
modificarOfertaGeneral(int, Oferta)	ResponseEntity<Respuesta>
eliminarOferta(int)	ResponseEntity<Respuesta>
verificarCalificacionOfertaInstitucion(int, int)	ResponseEntity<RespuestaGeneral>
verificarCalificacionOfertaFreelancer(int, int)	ResponseEntity<RespuestaGeneral>
calificarOfertaInstitucion(int, int, double, String)	ResponseEntity<Respuesta>
calificarOfertaFreelancer(int, int, double, String)	ResponseEntity<Respuesta>
getOfertasOfrecidasInstitucion(int)	ResponseEntity<RespuestaGeneral>
getOfertasOfrecidasFreelancer(int)	ResponseEntity<RespuestaGeneral>
getOfertasByInteligencia(int, int, int)	ResponseEntity<RespuestaGeneral>
getRecommendationByUser(int)	ResponseEntity<RespuestaGeneral>
<hr/>	
·ofertasDestacadas	ResponseEntity<RespuestaGeneral>
·allOfertas	ResponseEntity<JSONRespuesta>
·allOfertasByInteligencia	ResponseEntity<RespuestaGeneral>
·recommendations	ResponseEntity<RespuestaGeneral>

PROYECTO DE GRADO



Referencias

- Alarcón, J. M. (2017). *¿Qué es la máquina virtual de Java o Java Virtual Machine?*
Retrieved 2018-09-09, from <https://goo.gl/a67df7>
- Amazon. (2019). *¿qué es docker?* Retrieved 2019-04-12, from
<https://aws.amazon.com/es/docker/>
- Analyticsvidhya. (2018a). *Content-based filtering*. Retrieved 2019-05-20, from
<https://shorturl.me/GueT6xy9>
- Analyticsvidhya. (2018b). *User-user collaborative filtering*. Retrieved 2019-05-20, from
<https://shorturl.me/d453ZD>
- Angular.io. (2018). *What is Angular?* Retrieved 2018-09-23, from
<https://bit.ly/2H7XeUp>
- Apple. (2019). *User guide overview*. Retrieved 2019-03-09, from
<https://apple.github.io/turicreate/docs/userguide/>
- Art. 67. (2008). *Proyecto de acto legislativo no. 11/08 senado. por el cual se reforma el artículo 67 de la constitución política de colombia*. Retrieved 2018-04-15, from
<https://goo.gl/DakZq7>
- Asana.com. (2017). *Asana*. Retrieved 2017-06-1, from <https://asana.com/product>
- Beltran, P., G. (2015). *Sistema Recomendador Basado En Filtrado*. Retrieved 2018-09-09, from
<shorturl.at/GKW29>
- Benites, A. G. (2017). *¿Que és Java y por qué aprenderlo?* Retrieved 2018-09-09, from
<https://goo.gl/be7eSa>
- Benites, A. G. (2017). *¿Que és Java y por qué aprenderlo?* Retrieved 2018-09-09, from
<https://goo.gl/be7eSa>
- Betancur, D. M. (2009). *Modelo para la recomendación y recuperación de objetos de aprendizaje en entornos virtuales de enseñanza/aprendizaje Recommendation and*

PROYECTO DE GRADO

- retrieval model of learning object in virtual environments for teaching/learning*.
Revista Avances en Sistemas e Informática, 6(1). Retrieved 2019-11-05, from
<https://cutt.ly/VrpMx5D>
- Bustos L. M., Vázquez R. R., Alor H. G. (2015). *An architecture for developing Educational Recommender Systems. 8th Workshop on Intelligent Learning Environments. Research in Computing Science*. Instituto Politécnico Nacional. Vol. 106, pp 17-26. Retrieved 2020-01-15, from shorturl.at/fGLT8
- Colprensa. (2017, September 13). *Colombia es el país latino que menos invierte en educación, según la ocde*. *El País*. Retrieved 2018-04-15, from
<https://goo.gl/YXpahi>
- DNIA. (2018). *La institución*. Retrieved 2018-04-10, from <https://goo.gl/p6m4MB>
- EcuRed. (2020). *Metodología de cascada*. Retrieved 2020-01-04, from
<https://goo.gl/gnJMXJ>
- edX. (2018). *About edX*. Retrieved 2018-04-15, from <https://goo.gl/gnJMXJ>
- Educación 3.0 (2019). *Plataformas educativas que ofrecen formación online Virtual*
Retrieved 2020-01-10, from shorturl.at/hAY38
- ehu.eus. (2015). *La Máquina Virtual Java*. Retrieved 2018-09-09, from
shorturl.at/ciOSU
- Esaú, A. (2016). *Qué es la metodología Agile*. Retrieved 2020-01-10, from
shorturl.at/kJKR7
- Ferrándiz, C., Bermejo, R., Sainz, M., Ferrando, M., & Prieto, M. D. (1). *A study of logical-mathematical thinking from multiple intelligences framework*. *Anales De Psicología / Annals of Psychology*, 24(2), 213-222. Retrieved from
<https://revistas.um.es/analesps/article/view/42731>
- Fowler, M. (2014). *Microservices, a definition of this new architectural term*. Retrieved 2018-04-15, from <https://goo.gl/3r6KmM>
- Funiber. (2016). *¿Qué es FUNIBER?* Retrieved 2018-04-10, from <https://goo.gl/5dPnuL>

PROYECTO DE GRADO

- Galan, N. S. M. (2019). *Filtrado Colaborativo y Sistemas de Recomendación*. Retrieved 2019-03-09, from shorturl.at/nyGKM
- Google. (2019). *Compare trends*. Retrieved 2019-03-09, from <https://trends.google.com/trends/explore?geo=CO&q=estudialo,sena 20sofia> %
- Google Actívate. (2018). *¿Qué es Google Actívate?* Retrieved 2018-04-10, from <https://goo.gl/AXzYn3>
- JHipster. (2018). *Companies using jhipster*. Retrieved 2018-04-15, from <https://goo.gl/ZXvyDV>
- Lotz, M. (2013). *Waterfall vs. agile: Which is the right development methodology for your project?* Retrieved 2018-09-30, from <http://www.seguetech.com/waterfall-vs-agile-methodology/>
- Mimacom. (2017). *Arquitectura de microservicios con spring & netflix oss*. Retrieved 2019-04-20, from <https://www.mimacom.com/es/noticias/arquitectura-de-microservicios-con-spring-y-netflix-oss/>
- Moya, R. (2013). *¿Que son los Sistemas de Recomendación?* Retrieved 2018-09-09, from shorturl.at/gopwR
- OCDE. (2016). La educación en colombia. In *Revisión de políticas nacionales de educación* (p. 33). Retrieved 2018-04-15, from <https://goo.gl/epQHEP>
- Oracle. (2018). *¿qué es la tecnología java y para qué la necesito?* Retrieved 2018-09-09, from <https://goo.gl/8pWDgy>
- Parra, M. (2016). *¿qué es y cómo funciona trello?* Retrieved 2016-11-24, from <https://www.postedin.com/blog/que-es-y-como-funciona-trello/>
- Perales, I. (2015). *¿qué es el jdk y el jre? , java*. Retrieved 2018-09-09, from <https://goo.gl/zzWQTr>
- Politécnico de Colombia. (2018). *La institución*. Retrieved 2018-04-10, from <https://goo.gl/2FPYx1>
- Prieto, M. V. (2014). *Inteligencias múltiples*. Retrieved 2018-11-10, from <https://goo.gl/wbTTbz>

PROYECTO DE GRADO

Raffai, Z. (2018, July 30). *What is spring boot?* Retrieved 2018-09-09, from <https://goo.gl/6MwTv3>

Robles, V. (2017). *¿Que es un framework? ¿Para que sirven los frameworks de desarrollo web?* Retrieved 2018-09-09, from <https://goo.gl/Pr1DM2>

Rouse, M. (2016, December). *What is single-page application (SPA)?* Retrieved 2018-09-23, from <https://bit.ly/2I8di9t>

Selwyn, N., (2013). *Internet y educación.* Retrieved 2019-11-23, from shorturl.at/lsu28

Semana (2017). *La mayoría de universitarios no terminan la carrera que empiezan.* Retrieved 2019-11-23, from shorturl.at/nxEF6

SENA. (2017). *¿Quiénes somos?* Retrieved 2018-04-10, from <https://goo.gl/qfoUeB>

Sharma, P. (2018). *Comprehensive guide to build a recommendation engine from scratch (in python).* Retrieved 2019-05-20, from <https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-recommendation-engine-python/>

SimilarWeb. (2017). *Worldwide overview.* Retrieved 2018-10-08, from <https://bit.ly/2PQw8oD>

SimilarWeb. (2019). *Website performance.* Retrieved 2019-03-09, from <https://bit.ly/2uYiQh4>

Tena, M. (2018). *¿Qué es la metodología 'agile'?* Retrieved 2020-01-12, from shorturl.at/BIOS4

Thompson, J. (2015, April 23). *Introduction to Spring Expression Language (SpEL).* Retrieved 2018-09-09, from <https://goo.gl/aS32A9>

TIOBE. (2018). *TIOBE Index for April 2018.* Retrieved 2018-04-15, from <https://goo.gl/JW5NGB>

Tutellus. (2018). *Sobre Tutellus.* Retrieved 2018-04-15, from <https://goo.gl/vf9Mb6> Udemy.

(2018). *Enriching lives.* Retrieved 2018-04-10, from <https://goo.gl/3Qvujd> Unir. (2016). *Titulación universitaria.* Retrieved 2018-04-10, from

<https://goo.gl/rk12Bw>

PROYECTO DE GRADO

Universia. (2016). *Qué es universia*. Retrieved 2020-01-10, from shorturl.at/bjp28

Universidad San Mateo. (2017). *La importancia de la educación virtual*. Retrieved 2018-04-15, from <https://goo.gl/anj1hF>

Vesin B., Klašnja-Milićević A., Ivanović M. (2016). *Applying Collaborative Tagging for Providing Recommendation in Programming Tutoring System*. In: Chiu D., Marenzi I., Nanni U., Spaniol M., Temperini M. (eds) *Advances in Web-Based Learning – ICWL 2016*. ICWL 2016. Lecture Notes in Computer Science, vol 10013. Springer, Cham

Álvarez, C. (2014, August 12). *¿Qué es Spring Framework?* Retrieved 2018-09-09, from <https://goo.gl/FS7b8q>