



## Template

2018-12-20

### About arc42

arc42, the Template for documentation of software and system architecture.

By Dr. Gernot Starke, Dr. Peter Hruschka and contributors.

Template Revision: 7.0 EN (based on asciidoc), January 2017

© We acknowledge that this document uses material from the arc 42 architecture template, <http://www.arc42.de>. Created by Dr. Peter Hruschka & Dr. Gernot Starke.

### Note

This version of the template contains some help and explanations. It is used for familiarization with arc42 and the understanding of the concepts. For documentation of your own system you use better the *plain* version.

### Integrantes:

- Natalia Arias Lozano
- Ronald Cardona Martínez

## Introducción y Metas

Los recientes desarrollos tecnológicos en manejo y procesamiento de datos, vinculados con el avance de nuevos canales por donde se ofrecen productos o servicios a usuarios, abren una posibilidad no solo de gestionar la información y procesarla sino llevarla y conectarla con los usuarios de forma óptima, seleccionado el mejor canal para cada fuente o conjunto de información, en función de las necesidades, requerimientos y particularidades de cada usuario, lográndose personalización y mejor experiencia de uso.

Aunque la mayoría de empresas que han adoptado modelos de negocio que involucren esta nueva interacción por parte de los usuarios, las organizaciones estudiantiles no se quedan atrás por lo que hacen un intento por reconfigurar la forma en que interactúan y realizan transacciones con sus mercados para obtener una ventaja competitiva, han integrado ampliamente nuevos canales en sus operaciones cotidianas (Molina, 2017) y los clientes han obtenido acceso a una multitud de formas de interactuar con los proveedores de productos y servicios. Los usuarios están experimentando la educación y el aprendizaje en espacios físicos, en los espacios digitales que posee (como el sitio web) y en los espacios digitales que no posee (incluidos los sitios web de redes sociales), y posiblemente no les guste. Las organizaciones educativas son conscientes del entorno multicanal y responden al intentar crear contenido una vez y reutilizarlo en todos los canales. Este es un gran primer paso, pero los usuarios están demandando más (Wisniewski, 2017).

En el presente documento se indica a grandes rasgos como se implementará una plataforma de tecnología m-learning y aprendizaje ubicuo que permita que la educación sea cada vez mas accesible y brinde planes de estudios personalizados, proveyendo aprendizaje experiencial en cualquier momento y basado en el contexto del usuario.

## Vista de Requerimientos

### *Drivers del Negocio*

- Maximizar el proceso de aprendizaje al utilizar el canal más adecuado para el perfil de cada estudiante.
- Lograr una hiper personalización de la malla curricular de cada estudiante de manera que este pueda tener un contenido adaptativo que se ajuste a sus necesidades concretas.

### *Historias de Usuario*

Identificador de la Historia	Rol	Característica / Funcionalidad	Razón / Resultado	Número de Escenario
HU-001	Yo como Estudiante / Profesor	Necesito autenticarme en el sistema utilizando mi correo y contraseña de la universidad.	Con la finalidad de acceder al contenido de mis cursos.	1
				2
				3
				4
HU-002	Yo como estudiante	Necesito ver los cursos en los que estoy matriculado en mi pagina de inicio	Para mantenerme actualizado acerca del avance de mis objetivos de aprendizaje	1
				2
				3
				4
HU-003	Yo como Estudiante	Necesito matricularme en un curso	Con la finalidad de acceder a sus contenidos.	1
				2
				3
				4
HU-004	Yo como Estudiante	Necesito ver la información general del curso como su currículo y tiempo estimado para completar	Para conocer mejor sus temas y tomar la decisión de tomarlo o no	1
				2
				3
				4
HU-005	Yo como Estudiante	Puedo cancelar un curso en el que estoy matriculado	Para dejar de tener acceso a sus contenidos y no recibir más actualizaciones de este	1
				2
				3

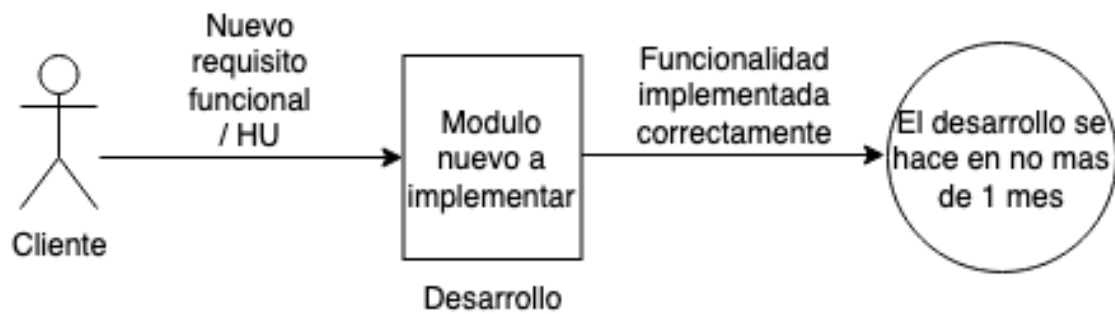
				4
HU-006	Yo como Estudiante	Necesito buscar cursos relacionados con temas de mi interés	Para conocer sus contenidos e información general acerca de ellos	1
				2
				3
				4

## Metas de Calidad

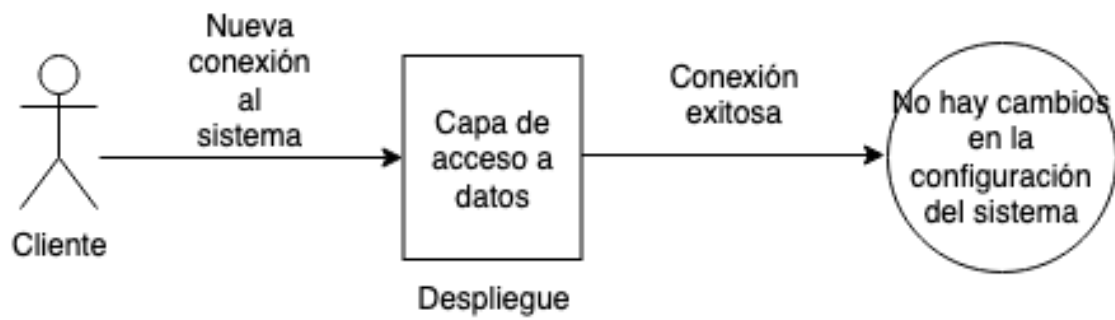
Nro	Meta de Calidad	Motivación
1	Portabilidad	El sistema debe ser agnóstico a los sistemas con los que se comunica en el contexto, por ejemplo, el sistema debe permitir una conexión con cualquier LMS (learning management system) de una institución educativa sin cambios en la configuración del sistema.
2	Adaptabilidad	La arquitectura debe estar diseñada de manera que sea posible cambiar el comportamiento del sistema a través del tiempo, teniendo en cuenta la retroalimentación que recibe del uso del estudiante.
3	Accesibilidad:	Los contenidos a los que se expone el usuario siempre deben ser los mas adecuados y precisos en función del contexto en el que se encuentra.
4	Conectividad	El sistema debe de soportar diferentes tipos de carga a nivel de red y requerimientos computacionales de los dispositivos del lado del cliente, conservando el 100% de las funcionalidades.

5	Extensibilidad	Es sistema debe tener la capacidad de agregar nuevos requerimientos, es decir, lo que se busca es aplicar el principio Open-Closed a nivel de arquitectura. Añadir una funcionalidad debe tomar menos de 30 días.
---	----------------	---

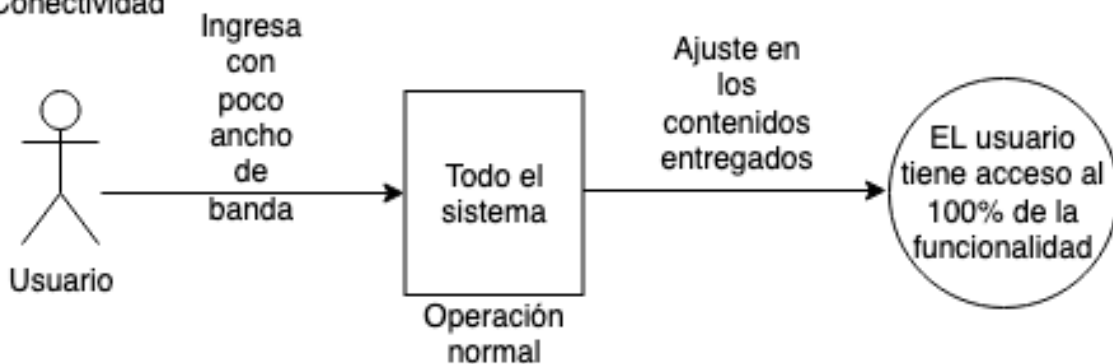
### Extensibilidad



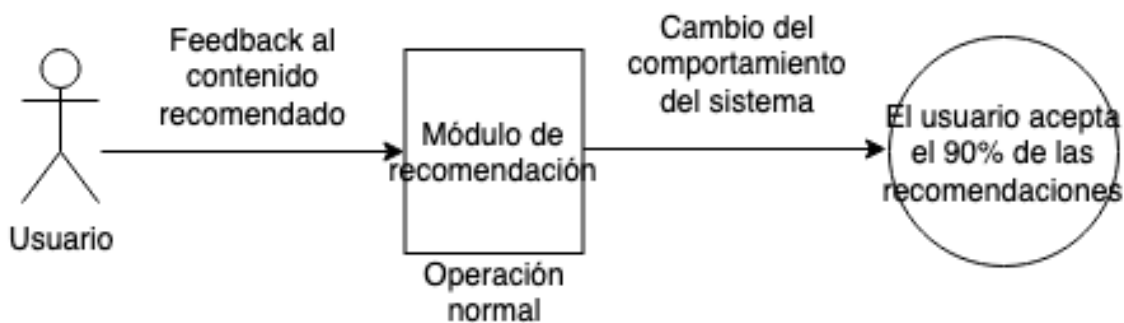
### Portabilidad



### Conectividad



### Reflectividad



## Stakeholders

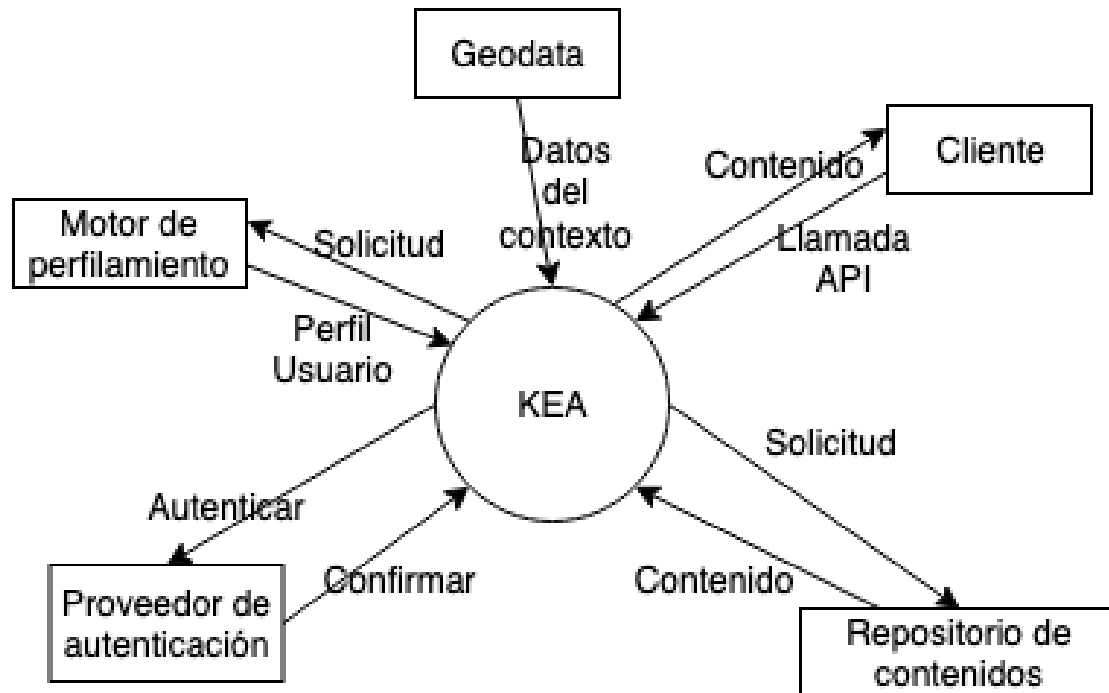
Role/Nombre	Contacto	Expectativas
<i>Product Owner / Marta Silvia Tabares</i>	<a href="mailto:mtabares@eafit.edu.co">mtabares@eafit.edu.co</a>	Desarrollar una plataforma de nivel TLR 6-7, sistema probado en entorno real.
<i>Co-investigadora / Paola Andrea Vallejo</i>	<a href="mailto:pvallej3@eafit.edu.co">pvallej3@eafit.edu.co</a>	<i>Lograr un enfoque educativo experiencial y ubicuo de fácil uso que aumente la calidad, eficacia y eficiencia de los programas de educación.</i>
<i>Desarrolladores / Natalia Arias, Ronald Cardona, José Sánchez.</i>	<a href="mailto:nariasl@eafit.edu.co">nariasl@eafit.edu.co</a> <a href="mailto:rcardo11@eafit.edu.co">rcardo11@eafit.edu.co</a> <a href="mailto:jsanch81@eafit.edu.co">jsanch81@eafit.edu.co</a>	<i>Esperan potenciar sus habilidades en el desarrollo de aplicaciones modernas.</i>
<i>Líder del proyecto / Paola Andrea Prada</i>	<a href="mailto:papradas@eafit.edu.co">papradas@eafit.edu.co</a>	<i>Implementar una metodología clara y estructurada que integre tanto elementos de desarrollos de ingeniería, como elementos de antropología digital, educación y usabilidad.</i>
<i>Líder del proyecto / José Antonio Solano</i>	<a href="mailto:jsolanoz@eafit.edu.co">jsolanoz@eafit.edu.co</a>	<i>Obtener resultados de producto y de proyecto que sean técnicamente óptimos, cumpla con las necesidades y requerimientos de uso y aplicación identificadas dentro de los tiempos propuestos.</i>
<i>Usuarios</i>		<i>Acceder a contenido educativo personalizado de acuerdo con mis gustos y mi contexto en cualquier parte, cualquier hora y en cualquier dispositivo.</i>

## Restricciones de la Arquitectura → en tabla restricción/motivación

- El proyecto de omnicanalidad en la educación hace parte de un portafolio de proyectos, este debe seguir un estilo arquitectónico orientado a microservicios para que la integración con el resto de los sistemas sea transparente.
- Seguir los estándares LOM (Learning Object Metadata) y CEDS (Common Education Data Standards) para el modelo de datos.
- Todas las herramientas que se utilicen en el proyecto deben ser open source.

## Alcance y Contexto del Sistema

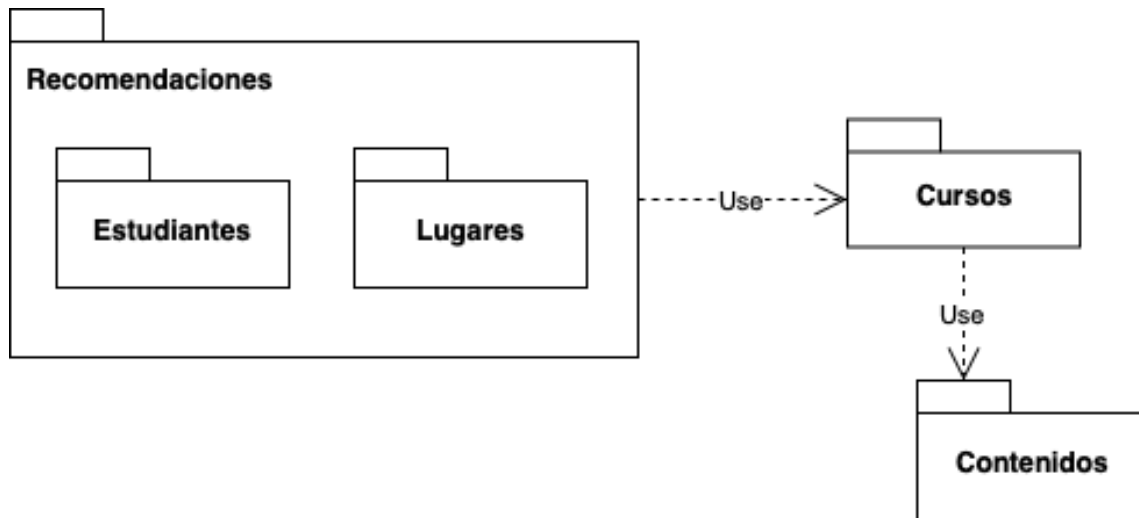
### Diagrama de contexto



El sistema de recomendación se alimenta de los datos de el contexto del usuario (Geodata), de los datos de perfilamiento del usuario (Motor de perfilamiento) y de el repositorio de contenidos para formar las recomendaciones de contenidos que mejor se adaptan a las preferencias de aprendizaje del estudiante y que le permiten cumplir con sus objetivos de aprendizaje. Asimismo, se muestra un cliente que se va a encargar de presentar los objetos de aprendizaje al usuario y de brindarle una interfaz para la gestión de estos, con su respectivo proveedor de identidades.

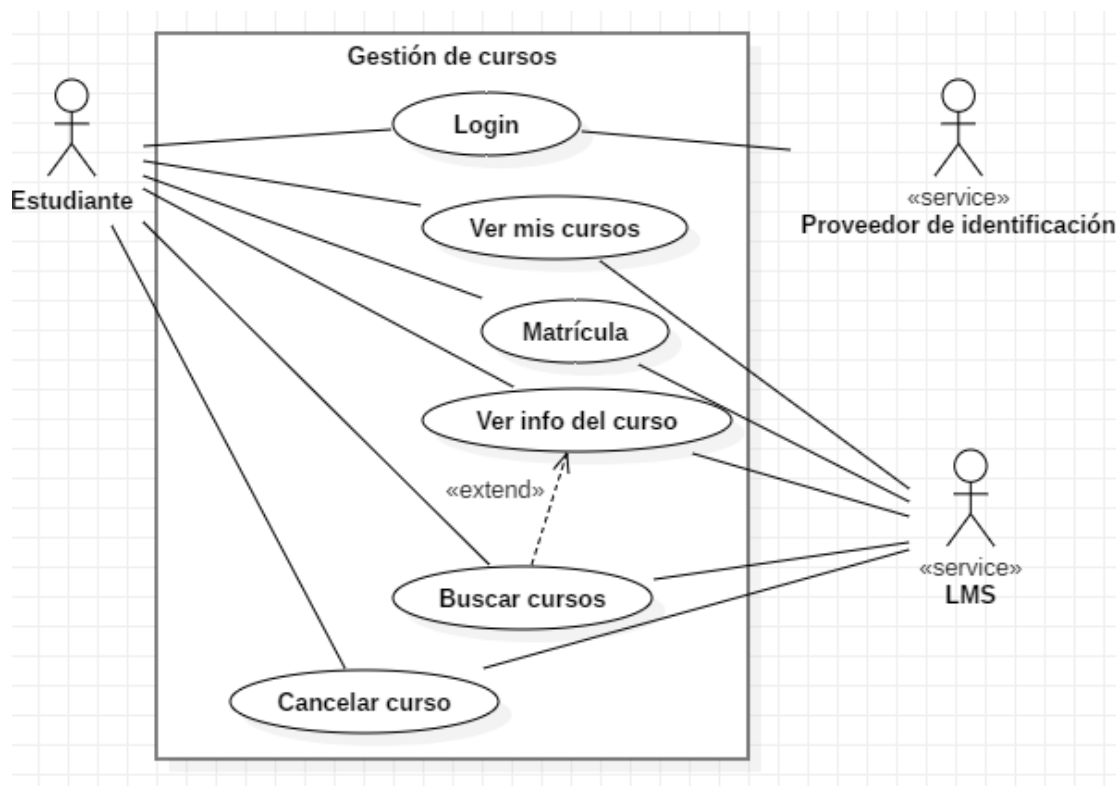


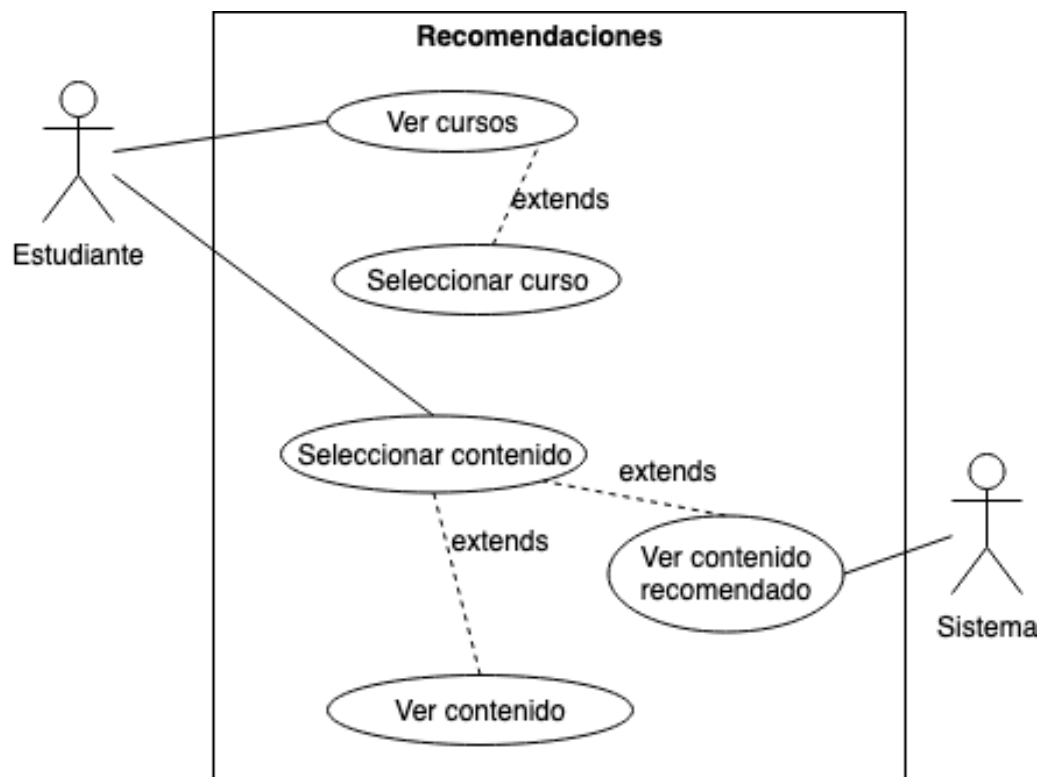
## Vista Conceptual



El diagrama anterior representa la visión que el usuario tiene sobre el negocio. Como se puede observar, un estudiante puede acceder a un curso y a un contenido recomendado por el sistema, el cual se basa en su perfil y en el lugar donde se encuentra, o en su defecto puede buscar otro.

## Casos de uso

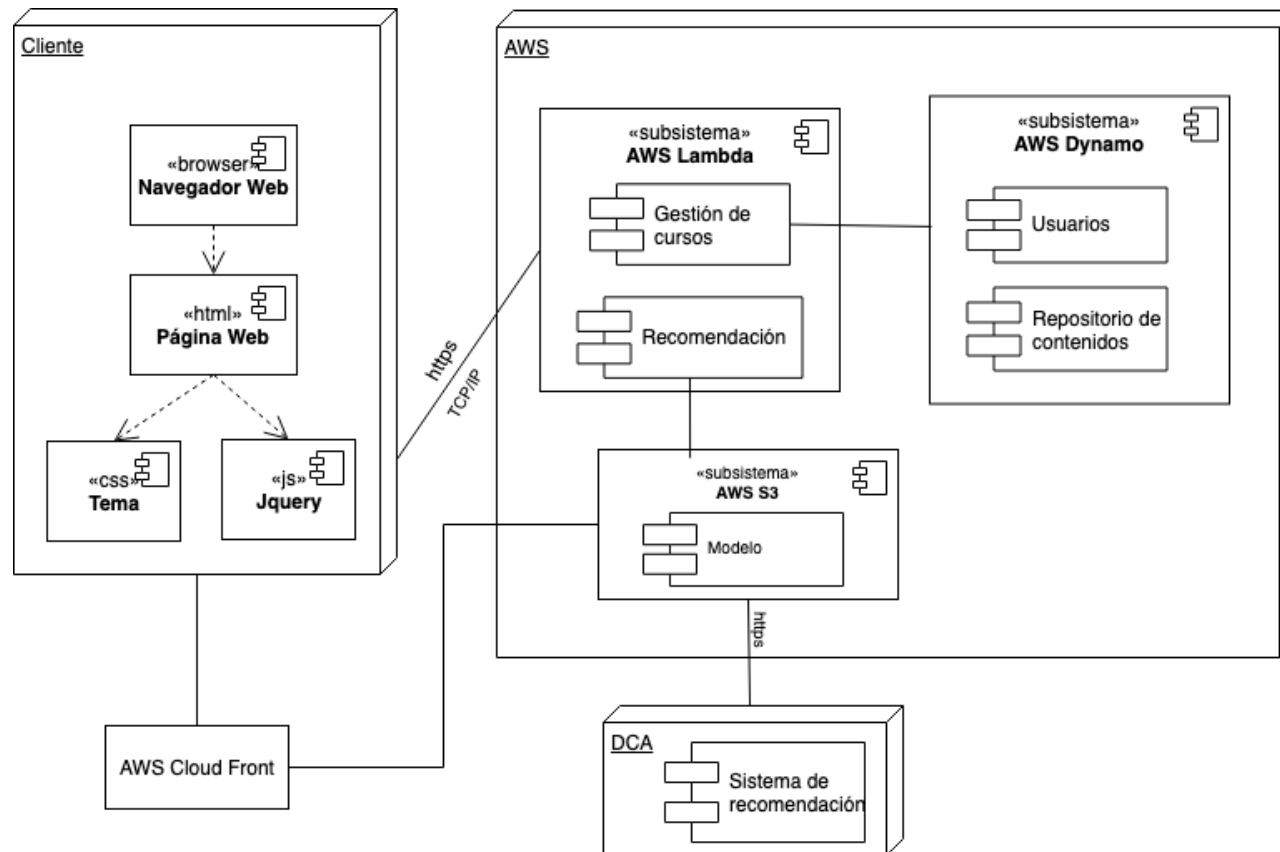




En los anteriores casos de uso se muestran las interacciones más importantes que se van a dar en el sistema. Estos, son los principales motivadores de cualquier decisión de arquitectura que se toma, ya que son los rasgos generales que debe cumplir el sistema.

Por ejemplo, el usuario es el principal actor externo que esta en contacto con el sistema, por medio de la plataforma web de aprendizaje, en este caso un estudiante puede utilizar el sistema para dos procesos generales: Gestionar sus cursos y desarrollar su proceso de aprendizaje. En cada una de estas etapas existen diferentes casos de uso. Algunos de ellos llevan al sistema a interactuar con otros de terceros o subsistemas dentro de este mismo programa. Un ejemplo claro de esto es, cuando un usuario ingresa a la plataforma necesita proveer unos datos de identificación, por motivos de seguridad, estos datos están almacenados en un servidor externo al sistema y por lo tanto debe establecerse una conexión a ese sistema para verificar la identidad.

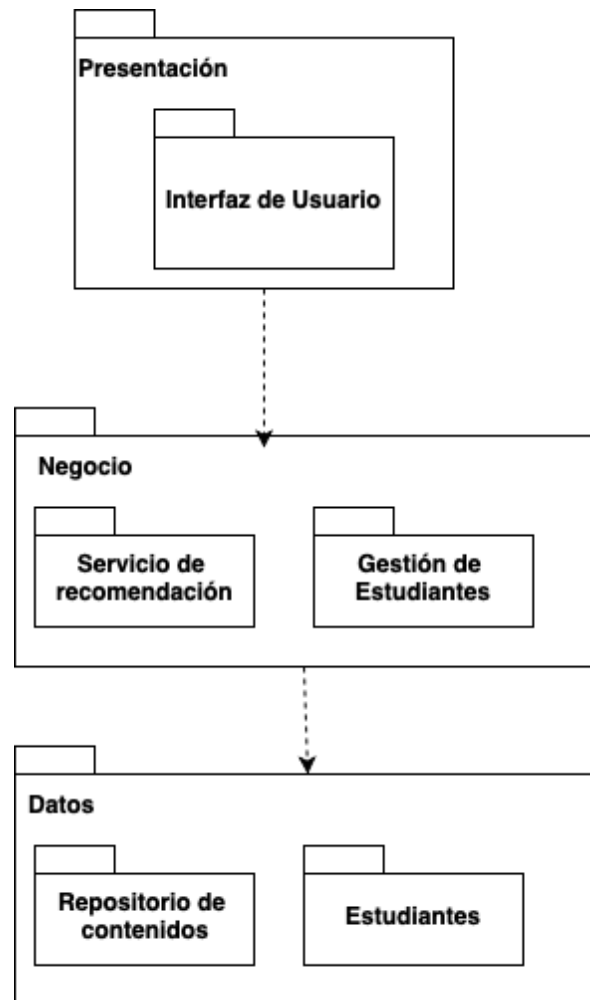
## Vista de despliegue



El despliegue de la aplicación se hará por medio de aws, para ello, utilizaremos s3, donde se guardan los elementos estáticos como lo es el modelo, tras ser entrenado en el DCA. También, se utilizará dynamo como base de datos para todo el manejo de la información de los usuarios y de los contenidos que se ofrecen.

## Vista Lógica

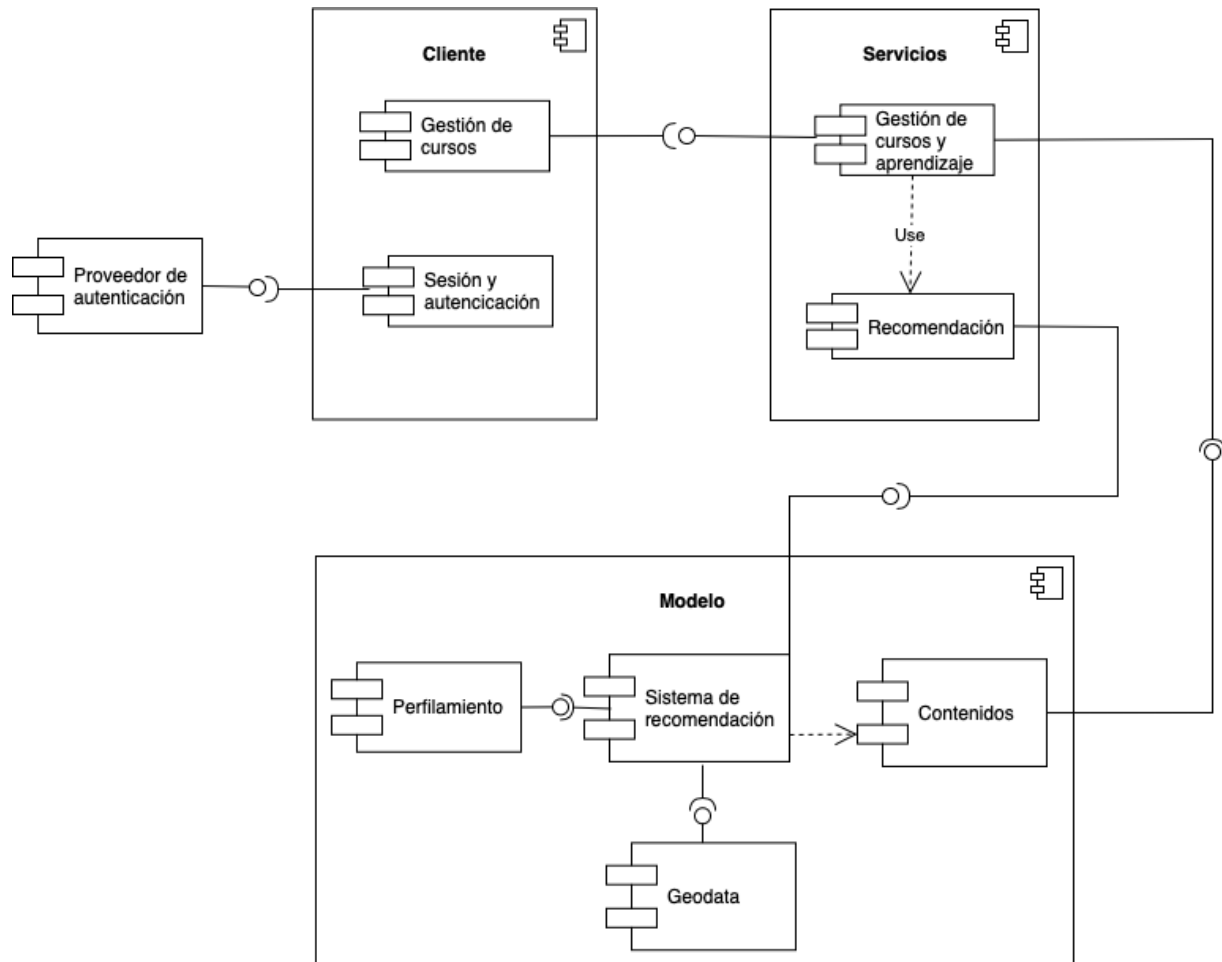
### Diagrama de paquetes de desarrollo



El modelo sigue el estilo arquitectónico 3 capas (presentación, negocio y datos). La capa de negocio contiene las funcionalidades *core* del sistema, las cuales se apoyan en la información de la capa de datos. El repositorio de contenidos está compuesto por objetos de aprendizaje que posteriormente serán recomendados por medio del servicio de recomendación, al usuario. Por otro lado, el paquete estudiante, esta conformado por los datos de todos los usuarios, con sus respectivos gustos y preferencias.

## Vista de Implementación.

### Diagrama de componentes.



En este diagrama también se evidencia el estilo arquitectónico 3 capas. Los componentes del cliente utilizan interfaces de otros, como el proveedor de autenticación que es externo al sistema y del modulo gestión de cursos y aprendizaje, el cual utiliza la recomendación devuelta por el componente restante de servicios. Esta se obtiene por medio de los diferentes elementos que conforman el componente modelo. Dicho componente utiliza tres fuentes de datos para generar una recomendación apropiada por el usuario.

## Estrategia de la Solución

### 1.1 Rendimiento y Escalabilidad

Requirement	How Met
1. El sistema debe soportar 1000 usuarios concurrentes.	Tener un balanceo de cargas, diseñar un backend <u>stateless</u> y tener una arquitectura elástica a nivel de aplicación y de base de datos que permitan una gran cantidad de usuarios al tiempo usando la aplicación.
2. El sistema debe aumentar la cantidad máxima de usuarios a medida que este sobrepasa el numero máximo de usuarios.	Se tendrá un sistema que permitirá el escalamiento horizontal y solo será necesario agregar más contenedores o mejorar las especificaciones de las bases de datos para atender a más usuarios.
3. Dependiendo del servicio, el sistema debe proveer un tiempo de respuesta menor a 3 segundos dadas las condiciones básicas de conectividad.	Se buscará implementar una arquitectura orientada a <u>microservicios</u> para que los recursos escalen individualmente según la necesidad del momento.

### 1.2 Seguridad

Requirement	How Met
1. Todos los usuarios deben estar autenticados para poder acceder a las funcionalidades de la aplicación.	Se validará la autenticación de los usuarios contra el Active <u>Directory</u> o LDAP de la Universidad EAFIT.
2. Encriptar la información personal del usuario.	Utilizaremos el protocolo HTTPS que nos permite enviar la <u>información</u> encriptada.

### 1.3 Disponibilidad y Resiliencia

Requirement	How Met
1. El sistema debe estar disponible un 99.9%	Tener servidores en balanceo de carga, tener replicados los datos y tener respaldo en caso de incumplimiento de <u>SLAs</u> por parte de los proveedores y arquitectura de <u>microservicios</u> . Esto se logra gracias al uso de los servicios de la capa <u>PaaS</u> de los proveedores de nube (AWS, <u>Azure</u> , GCP). Por definir.
1. Tolerancia a fallos.	En caso de fallo de alguno de los componentes ( <u>Aplicación</u> , balanceadores o base de datos), cada uno de estos niveles debe implementar un mecanismo de <u>fall-back</u> y <u>failover</u> para asegurar la consistencia de los datos de los usuarios.

## 1.4 Extensibilidad

Requirement	How Met
1. Debe ser posible agregar funcionalidades nuevas sin necesidad de hacer cambios grandes en el código base.	El código debe estar altamente desacoplado para que permita reutilizar y agregar código fácilmente. .

### Otros Atributos de Calidad:

#### • Accesibilidad

Este atributo es el objetivo principal del sistema pues al buscar la *omnicanalidad* lo que se busca es garantizar un acceso ubicuo en cualquier instante. Esto lo logramos a partir de la diversidad de canales (web y móvil) a partir de los cuales los usuarios pueden acceder a todos los servicios proporcionados independientemente del canal elegido.

#### • Regulación

En Colombia aplica la ley de habeas data (Ley Estatutaria 1266 de 2008), la cual satisfacemos al mantener privados los datos de las personas e inmutables por parte del equipo del proyecto u otras personas que no sean los dueños de la información.

#### • Usabilidad

El sistema debe ser de fácil uso y entendible por el tipo de público al que estamos apuntando, que son estudiantes de la Universidad EAFIT.