

# Architecture Definition Document

Daniel Esteban Rodriguez Meza      Nicolas Ricardo Enciso  
Freddy Alejandro Cuellar      Brayan Fabian Prieto  
Santiago Peña Sanabria

October 2018

## 1 Introducción

### 1. Nombre del sistema de software:

SAPIENSBOK.

### 2. Descripción del sistema de software:

La idea del proyecto, es la de la creación de una red social de grupos de investigación y sus miembros. El usuario podrá crear un grupo de investigación, en donde pondrá las áreas en las que se trabaja, los perfiles de personas que se pueden unir al grupo, así como miembros del grupo, intereses, metodologías, resultados y eventos que se lleven a cabo por el grupo. Así también, el usuario podrá unirse a un grupo de investigación, por lo que el administrador del grupo al que aplicó será la persona encargada de la admisión de la persona.

De igual forma, un grupo de investigación podrá invitar a algún usuario a ser parte de su grupo de investigación, para poder así ir conformando el equipo de trabajo. En los grupos de investigación se podrá tener un chat grupal por medio del cual los miembros se podrá comunicar. También en el apartado del grupo, se podrá ir viendo la actividad de los miembros, publicaciones, resultados, documentación, calendario y demás documentación que consideren es necesario para poder trabajar.

También se considera la idea de que empresas y entidades del estado, así como los departamentos encargados de la investigación en las universidades puedan ver las dinámicas de los grupos de investigación, y apoyen los procesos, ya sea con reclutamiento del grupo, apoyos económicos o de oportunidades académicas.

### Descripción general idea

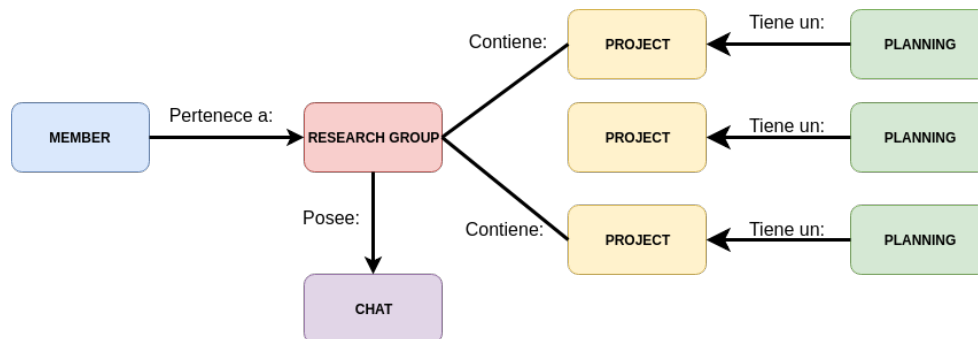
La idea general en la que se componen los microservicios, se basa en un orden de contención y pertenencia. La contención, hace referencia a

aprovechar la arquitectura de microservicios, para dividir por componentes los elementos que hacen parte del sistema, siendo así, que un componente esté compuesto a su vez de otros componentes. Por ejemplo, un grupo de investigación, adicional a su información general, está compuesto de proyectos, los cuales a su vez, tienen un componente de planeación que es un microservicio independiente.

Cada microservicio al ser independiente, hace más fácil el mantenimiento del sistema, y la forma en la cual se componen los elementos. En cuanto a pertenencia, varios microservicios tienen a su vez componentes de usuarios, en éste caso de investigadores (researchers), los cuales hacen parte de cada grupo de investigación, así como de los proyectos que se realizan en un grupo de investigación.

A continuación se muestra un modelo de cómo algunos microservicios componen a otros.

A continuación se muestra un esquema que ilustra la idea detrás del sistema de software, en cuanto a su composición lógica.



Composición de los microservicios

### Oportunidades del sistema de software:

La relevancia de nuestro proyecto, está en la de brindar una plataforma que ayude a los miembros de la academia, entre estudiantes y profesores, a organizar sus grupos de investigación, y se haga de forma paralela una difusión de los grupos que se mantienen activos en las instituciones de educación. Se quiere crear una plataforma que ayude a las personas a unirse a grupos de investigación, crearlos y trabajar en ellos, para que se fomente la investigación, y se invite a los miembros de la comunidad académica a ser parte de alguno de éstos grupos, eliminando barreras de contacto, conocimiento, temática y demás.

Adicionalmente, se busca que en la plataforma, se incentive el trabajo colaborativo, y sobre todo multidisciplinario, para que se creen proyectos con gran alcance y envergadura, que logren dar resultados beneficiosos para

sus áreas de estudio y la sociedad, para lo cual, se considera un componente de promoción de grupos con empresas, entidades estatales y demás, que ayuden a desarrollar los proyectos propuestos en los grupos, adicional a ser una forma de promoción de la ciencia mostrando las publicaciones hechas por lo grupos de investigación.

### 3. Identificación y descripción de los intereses (concerns) y sus respectivos interesados (stakeholders)

#### **Stakeholder: Estudiante pregrado**

**Concern:** Este personaje es aquella persona que está ingresando al mundo de la investigación, y que tiene interés en ser parte de algún grupo de investigación, así como ser el encargado de la administración de un grupo de investigación al que haya ingresado.

#### **Stakeholder: Estudiante maestría / doctorado**

**Concern:** Este personaje es aquella persona que tiene como principal foco de estudio la investigación en un determinado tema, por lo que conoce a profundidad sobre el tema, y es generador de conocimiento. Tiene una estrecha relación con estudiantes de pregrado y de maestría o doctorado, de forma en que ya sea ayuda a otros a empezar a investigar desde pregrado, o forma grupos de estudio enfocados en la investigación, puede ser un creador de grupos de investigación. Actúa bajo órdenes de su profesor tutor.

#### **Stakeholder: Profesor universitario**

**Concern:** Este personaje es la clave y la base de los grupos de investigación, es la persona encargada de crear los grupos, dirige las investigaciones, es tutor de los estudiantes de maestría y doctorado, así como también guía para estudiantes de pregrado que entran en la investigación, es quien toma las decisiones en el grupo, y el rumbo que va tomando éste.

#### **Stakeholder: Direcciones de investigación**

**Concern:** Es el departamento encargado de la gestión administrativa de los grupos de investigación en la universidad, el cual puede ver o tomar datos de los grupos que están funcionando en la plataforma, a modo de hacer tener una aproximación de las dinámicas de los grupos de investigación.

#### **Stakeholder: Empresas externas**

**Concern:** Son empresas, startups o compañías las cuales pueden ver los grupos de investigación que pueden ser apoyados o reclutados en su trabajo, viendo el potencial de sus proyectos y las personas que trabajan allí.

**Stakeholder: Entidades de gobierno para la investigación**

**Concern:** Son departamentos del estado, que tiene a cargo alguna función siendo instituciones de carrera, o que están relacionadas con el campo académico, de forma que puedan ver los grupos que están haciendo investigación, y se interesen en aquellos a los que les ven potencial, apoyándolos.

**Stakeholder: Jeisson Andrés Vergara Vargas**

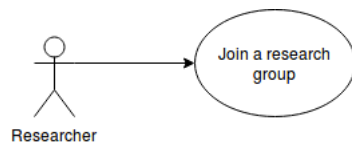
**Concern:** Profesor de la materia de arquitectura de software, quien determina la viabilidad de los componentes de proyecto, así como gestor principal en el desarrollo del mismo. Adicionalmente, es el evaluador del proyecto.

**Stakeholder: Henry Roberto Umaña Acosta**

**Concern:** Profesor de la materia arquitectura de software, quien apoya el proceso de desarrollo del proyecto, y es guía adicional en cuanto a conocimientos conceptuales aplicados al proyecto.

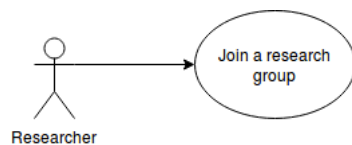
## 2 Requisitos Funcionales

- Unirse a un grupo de investigación

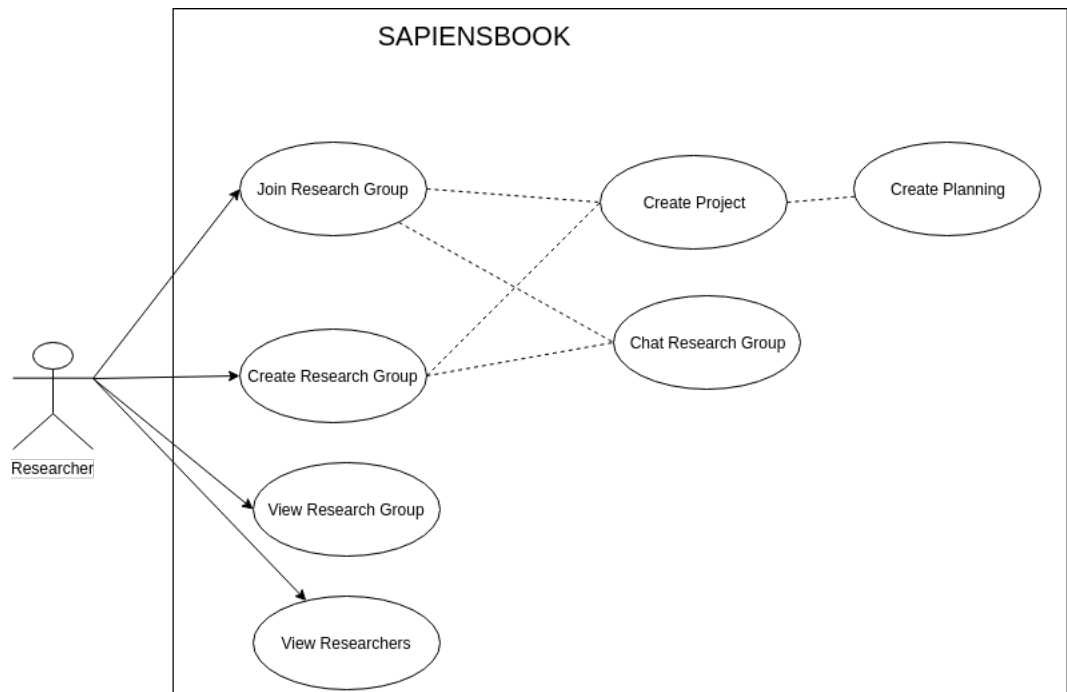


En este caso cualquier usuario (investigador) podrá aplicar a un grupo de investigación de acuerdo con sus habilidades y experiencia

- Crear un grupo de investigación



Un investigador podrá también crear un grupo, siendo este el director del mismo



En el anterior diagrama se muestrta que un researcher puede entrar o crear un grupo, ver grupos u otros investigadores.

### 3 Requisitos No Funcionales

Sin Contenido

### 4 Diseño Arquitectónico

1. Descripción de los estilos y/o patrones arquitectónicos usados en el diseño e implementación de los primeros componentes construidos.

**Patrones arquitectónicos usados:**

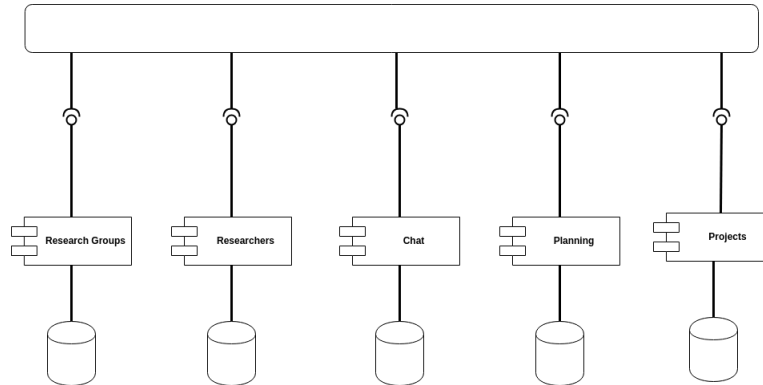
**Patrón microservicios:**

En la planeación de nuestro proyecto, la idea general que se siguió, se basó fundamentalmente en los lineamientos dados por el Delivery 1, es decir, la idea fundamental, fue seguir el patrón de arquitectura de microservicios.

Cada miembro del grupo creaba uno de los microservicios de todo el sistema de software, para posteriormente, reunirlos, usando el concepto

REST, en donde cada microservicio se comunica por peticiones HTTP, con un formato estándar, como lo es JSON.

A continuación, se muestra un esquema de la arquitectura de microservicios, con los respectivos nombres, adicional a la base de datos que cada microservicio tiene individualmente.



Modelo general de la arquitectura Sapiensbook

A continuación, se muestra el microservicio que cada miembro del equipo desarrolló con el respectivo lenguaje usado:

NOMBRE	MICROSERVICIO	LENGUAJE
Daniel Rodriguez	Researchers	Javascript
Santiago Peña	Research Groups	Ruby
Nicolas Enciso	Planning	C#
Alejandro Cuellar	Projects	Go

Table 1: Asignación microservicios y lenguaje

### Patrón monolítico:

Teniendo en cuenta que se sigue una arquitectura de microservicios, donde cada miembro hace uno de éstos, se usa entonces adicional, el patrón monolítico, en la medida en que, siguiendo la idea de microservicio, se busca tener componentes que sean sencillos de testear, desplegar y escalar.

En ese orden de ideas, cada microservicio sigue un patrón monolítico, de una aplicación que se conecta a una base de datos, siendo altamente acoplado, pero que tiene como ventaja la sencillez, y el aprovechamiento de seguir bajo la arquitectura de microservicios, un estándar de comunicación entre éstos, usando JSON, por lo cual, a la hora de escalar, es simplemente crear pequeñas partes (microservicios).

**Patrón modelo vista controlador MVC:**

De cierta forma, la arquitectura de microservicios, tiene mucha similaridad con MVC, en cuanto a que, cada microservicio emplea de manera parcial MVC. El microservicio cuenta con la capa de Modelo para hacer uso de persistencia de datos (Base de datos), y con un controlador, que se asocia básicamente al microservicio en cuanto al lenguaje de programación y el framework usado, quien a su vez, gestiona las peticiones que se llevan al modelo.

**Patrón n-Tier**

En cuanto a los microservicios, los cuales reciben peticiones HTTP y devuelven JSON con información luego de hacer uso de una base de datos, se contiene una idea de que en el nivel de microservicios, existe la lógica de backend, que contiene el patrón n-Tier. El microservicio se encarga de resolver peticiones y gestionar la persistencia. Si bien, en la etapa en la que se encuentra el proyecto, no se hace uso de ninguna UI, se considera en el desarrollo posterior, por lo que, el patrón de tener un frontend que envía y recibe peticiones hacia un backend, quien a su vez devuelve información, para que frontend los muestre al usuario, se hace, en efecto, uso del patrón n-tier, en nuestro sistema de software.

**Estilos arquitectónicos usados:****Estilo capas(layered):**

Nuestro sistema hace uso de capas, donde los microservicios exponen una interfaz a modo de API de tipo REST que es usado por las capas superiores, quienes a su vez, van usando éstos, modificarlo y subirlos en la cadena de capas, donde la final, es la de presentación, donde la UI le muestra al usuario la información requerida.

**Estilo cliente servidor:**

El cliente servidor es evidente, en cuanto a la forma en la cual se hacen las peticiones a los microservicios, conservando el principio cliente-servidor de que, los servidores, en éste caso los microservicios, no conocen a los clientes, pero los clientes, es decir el navegador por el cual se consume en microservicio, si conoce el servidor.

**2. Modelos de datos (un diagrama E-R por cada base de datos).****• Planeación o Planning:****Descripción breve:**

Éste microservicio contiene la información referente a fechas de inicio y finalización de cada plan en un proyecto, el encargado del plan, e

información referente a la actividad del plan en el proyecto, como publicaciones y descripciones.

Planning	
PK	Id
	DayStartPlan(byte)
	MonthStartPlan(byte)
	DayEndPlan(byte)
	MonthEndPlan(byte)
	Publish(String)
	Resources(double)
	Description(String)

Modelo ER Planning

- **Lenguaje:** C#
- **Framework:** ASP.NET
- **Base de datos:** SQL Server

- (a) **Research groups:** Éste microservicio es el encargado de alojar la información referente a los grupos de investigación que se van creando en la plataforma, y que constituyen la base del sistema.

Research group	
	Id
	Topics(Array)
	Website(string)
	Fields(array)
	Logo(string)
	University(String)
	Status(String)
	Skill(String)
	Member(Array)
	Director (String)
	Leader (String)
	Description (String)

Modelo ER Research

- **Lenguaje:** Ruby
  - **Framework:** Rails
  - **Base de datos:** Mongo DB
- (b) **Researchers o usuarios:** Éste microservicio es el que aloja a los investigadores, la información personal básica, y el perfil



académico, así como de experiencia y demás skills. Las personas pueden ser profesores o estudiantes.

User	
Id	
	Name(String) Lastname(String) Email(String) Password(String) Study_areas (Array) Organization (String) Profile(Object) Nationality (String) Gender(String) Languages (Array) Skills (Object)

Modelo ER User

- **Puerto:** 3001
  - **Lenguaje:** Javascript
  - **Framework:** NodeJS
  - **Base de datos:** Mongo DB
- (c) **Project:** Éste microservicio conforma los proyectos que viven en un grupo de investigación, con un tema específico y personas dedicadas a ejecutar el proyecto, haciendo que un grupo de investigación se conforme a partir de proyectos, y cada proyecto contiene un esquema de planeación.
- **Puerto:** 3004
  - **Campos:**
    - \* ID Proyecto (integer)
    - \* ID planeación (array, integer, ID planeación)
    - \* Status (string, lista de opciones predeterminada) S
    - \* Miembros asignados (integer, ID researchers)
    - \* Líder proyecto (integer, ID researcher)
    - \* Titulo (string)
    - \* Áreas de estudio (string, lista de opciones predeterminadas)
    - \* Descripcion (text)
  - **Lenguaje:** Go
  - **Framework:**