

Architecture Definition Document

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

October 2018

1 Introducción

1. Nombre del sistema de software: SAPIENSBOK.

(a) 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. De igual forma, el usuario podrá unirse a un grupo de investigación.

En cada grupo se tendrán planes, a manera de tareas que se tienen por cada proyecto en un grupo de investigación.

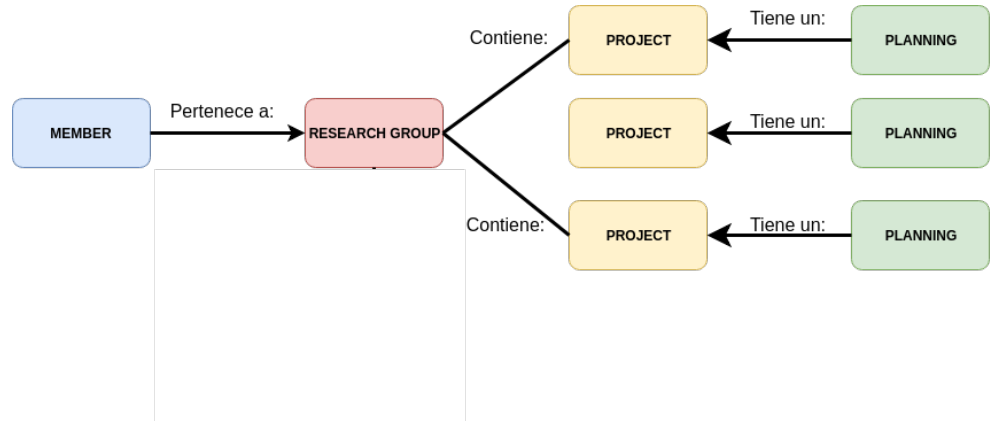
También se considera la idea de que empresas y entidades del estado, así como los departamentos encargados de la investigación en las universidades puedan ver las dinámicas de los grupos de investigación, y apoyen los procesos, ya sea con reclutamiento del grupo, apoyos económicos o de oportunidades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

(b) 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 el grupo de investigación.

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

(c) **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 los grupos de investigación.

(d) **Identificación y descripción de los intereses (concerns) y sus respectivos interesados (stakeholders)**

i. **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.

ii. **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.

iii. **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.

iv. **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.

v. **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.

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

Concern: Son departamentos del estado, que tiene a cargo al-

guna 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.

vii. **Stakeholder: Jeisson Andrés Vergara Vargas**

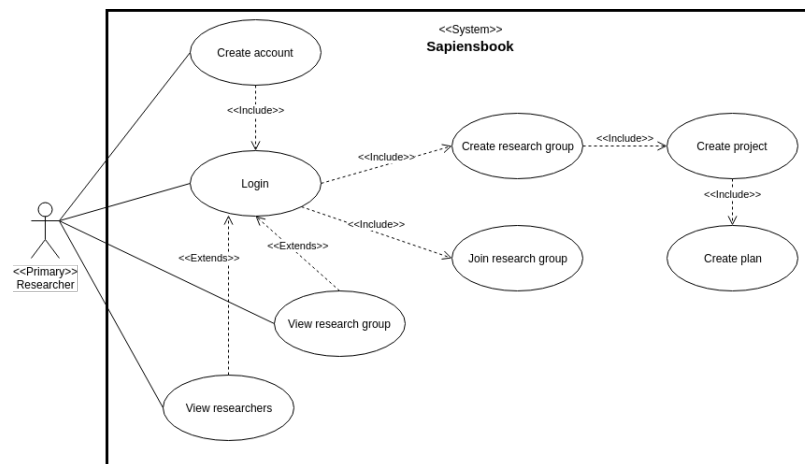
Concern: Profesor de la materia de arquitectura de software, quien determina la viabilidad de los componentes de proyecto, así como gestor principal en el desarrollo del mismo. Adicionalmente, es el evaluador del proyecto.

viii. **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

- Diagrama de caso de uso Sapiensbook



- A continuación se presentan los casos de uso con su respectiva descripción:

(a) Crear una cuenta (create account)

Nombre	Crear cuenta (create account)
Actores que participan	Researcher
Objetivos	Permitir el registro de nuevos usuarios en el sistema
Disparadores	Click en create account
Precondiciones	Estar en la vista inicial del sistema (home)
Post condiciones	Llevar al formulario de registro
Flujo básico	<ol style="list-style-type: none"> 1. Ingreso a la web de Sapiensbook 2. Click en create account 3. Redirigir a formulario de registro 4. Confirmación de registro
Flujo alternativo	Si ya se está registrado se dará aviso, o si se ingresan datos incorrectos.
Excepciones	<ul style="list-style-type: none"> - Problemas de conexión en la red - Datos incorrectos en formulario - Contraseña no válida - Sin confirmación de registrado
Cualidades de calidad	Login inmediato al sistema, conformación de cuenta.

(b) Login en sistema

Nombre	Login
Actores que participan	Researcher
Objetivos	Permitir ingresar a la cuenta del researcher
Disparadores	Click en Login
Precondiciones	Estar en la vista inicial del sistema (home), en view research groups o view researchers.
Post condiciones	Ingreso con cuenta al home del sistema, crear grupo de investigación, unirse a un grupo de investigación
Flujo básico	<ol style="list-style-type: none"> 1. Ingreso a la web de Sapiensbook 2. Click en login 3. Ingreso del username (email) y contraseña 4. Acceso o denegación de cuenta
Flujo alternativo	Si ya se está registrado se dará aviso, o si es incorrecto el usuario o la contraseña se dará aviso. Posibilidad de cambiar contraseña opción olvidé mi contraseña.
Excepciones	<ul style="list-style-type: none"> - Problemas de conexión en la red - Username o contraseña incorrecta - No está registrado el usuario (researcher)
Cualidades de calidad	Login inmediato al sistema, corto tiempo de espera.

(c) Ver grupo de investigación (view research group)

Nombre	View Research Group
Actores que participan	Researcher con o sin login
Objetivos	Permitir visualizar la información de un grupo de investigación
Disparadores	Click en alguno de los grupos de investigación que se muestran en el sistema
Precondiciones	Estar en la vista inicial del sistema (home), luego de haber hecho login, o al hacer una búsqueda
Post condiciones	Visualización de la información del grupo de investigación (miembros, temática, Universidad etc)
Flujo básico	<ol style="list-style-type: none"> 1. Ingreso a la web de Sapiensbook 2. Click en alguno de los grupos de investigación mostrados 3. Vista de la información completa del grupo de investigación
Flujo alterno	Grupo de investigación eliminado mostrando nula información
Excepciones	<ul style="list-style-type: none"> - Sin información del grupo de investigación - Grupo de investigación eliminado
Cualidades de calidad	Manejo de excepciones a causa de inconsistencia en registro de grupos de investigación, visualización sencilla de la información del grupo de investigación.

(d) Ver investigadores (view researchers)

Nombre	View Researchers
Actores que participan	Researcher con o sin login
Objetivos	Permitir visualizar de los researchers que están en el sistema.
Disparadores	Click en alguno de los researchers
Precondiciones	Estar en la vista inicial del sistema (home), luego de haber hecho login, o al hacer una búsqueda
Post condiciones	Visualización de alguna información del researcher (Nombre, temas, área, universidad)
Flujo básico	<ol style="list-style-type: none"> 1. Ingreso a la web de Sapiensbook 2. Click en alguno de los researchers mostrados 3. Visualización de información básica del researcher
Flujo alterno	Researcher eliminado con nula información, o no disponible.
Excepciones	<ul style="list-style-type: none"> - Sin información del researcher - No existe el registro del researcher mostrado
Cualidades de calidad	Manejo de excepciones a causa de inconsistencia en registro de researchers, visualización de sólo la información básica por motivos de seguridad.

(e) Crear grupo de investigación (create research group)

Nombre	Create research group
Actores que participan	Researcher con login
Objetivos	Crear un grupo de investigación (research group) en el sistema
Disparadores	Click en create research group
Precondiciones	Haber hecho login en el sistema
Post condiciones	Visualización de alguna información del researcher (Nombre, temas, área, universidad)
Flujo básico	<ol style="list-style-type: none"> 1. Hacer login en el sistema 2. Click en create research group 3. Llenar la información del grupo a crear 4. Confirmar creación
Flujo alterno	Nombre igual al de otro grupo ya registrado, datos mal ingresados, no haber hecho login antes.
Excepciones	<ul style="list-style-type: none"> - Nombre del grupo duplicado - No poder crear el grupo por no estar registrado el researcher
Cualidades de calidad	Manejo de excepciones a causa de inconsistencia en registro del grupo de investigación, verificación de nombres para evitar duplicados.

(f) Unirse a un grupo de investigación (Join research group)

Nombre	Join research group
Actores que participan	Researcher con login
Objetivos	Unirse a un grupo de investigación
Disparadores	Click en create research group
Precondiciones	Haber hecho login en el sistema
Post condiciones	Ser aceptado en el grupo de investigación
Flujo básico	<ol style="list-style-type: none"> 1. Hacer login en el sistema 2. Click en un grupo de investigación 3. Click en Join to the group 4. Recibir admisión por el director del grupo 5. Entrar al grupo
Flujo alterno	Luego de hacer click en buscar grupos de investigación, dar a unirse para aplicar, links hacia grupos.
Excepciones	<ul style="list-style-type: none"> - No ser admitido al grupo por el director - No poder acceder al grupo de investigación por eliminación del grupo - Salir del grupo a causa de la eliminación del mismo
Cualidades de calidad	Admisión por parte del grupo como libertad de elección, facilidad de aplicar al grupo, respuesta tan pronto sea dado el veredicto de admisión.

(g) Crear proyecto de investigación (create project)

Nombre	Create project
Actores que participan	Researcher con login perteneciente a un grupo de investigación (research group)
Objetivos	Crear un proyecto de investigación dentro de un grupo de investigación
Disparadores	Click en create project en el grupo de investigación
Precondiciones	Haber hecho login en el sistema, pertenecer al grupo de investigación
Post condiciones	Tener mínimo un encargado del proyecto
Flujo básico	<ol style="list-style-type: none"> 1. Hacer login en el sistema 2. Pertenecer al grupo de investigación donde se hará el proyecto. 3. Click en create project 4. Ingresar los datos solicitados 5. Proyecto confirmado como creado
Flujo alternativo	Duplicado de proyecto
Excepciones	<ul style="list-style-type: none"> - No ser parte del grupo de investigación - Datos mal ingresados - Imposibilidad de crear proyecto por eliminación del miembro o grupo
Cualidades de calidad	Administración de los proyectos del grupo, posibilidad de añadir personas al proyecto.

(h) Crear plan de proyecto (create plan)

Nombre	Create plan
Actores que participan	Researcher con login perteneciente a un grupo de investigación (research group), y que pertenezca al proyecto en donde se crea el plan
Objetivos	Crear un plan en un proyecto de investigación
Disparadores	Click en create plan, dentro de un proyecto de investigación
Precondiciones	Haber hecho login en el sistema, pertenecer al grupo de investigación, y ser miembro del proyecto de investigación donde se creará el plan
Post condiciones	Tener mínimo un miembro del plan
Flujo básico	<ol style="list-style-type: none"> 1. Hacer login en el sistema 2. Ingresar al grupo de investigación al que pertenece 3. Ingresar al proyecto de investigación del que hace parte 4. Click en crear plan 5. Ingresar datos 6. Confirmar creación de plan
Flujo alternativo	Duplicado del plan, imposibilidad de crear plan por no pertenecer al grupo, eliminación de un plan.
Excepciones	<ul style="list-style-type: none"> - No ser parte del grupo de investigación - Datos mal ingresados - No ser miembro del proyecto de investigación - Eliminación del proyecto o grupo
Cualidades de calidad	Administración de los planes del proyecto, posibilidad de añadir más miembros al plan de proyecto.

3 Requisitos No Funcionales

- El sistema debe responder al usuario en un tiempo menor a 5 segundos en cada proceso seleccionado.
- Los datos modificados en las bases de datos deben ser actualizados en menos de 5 segundos.
- Las contraseñas de los usuarios(researchers), deben ser cifradas usando RSA.
- La aplicación web debe tener un diseño de tipo responsive de manera que sea fácilmente usable en diferentes dispositivos con diferentes tamaños de pantalla.
- La aplicación web debe tener una disponibilidad de al menos 85% de las veces que el usuario (researcher) intente ingresar.

- (f) La aplicación web debe estar alojado en un ambiente cloud independiente de máquinas locales.
- (g) La aplicación web debe ser compatible con los navegadores web Google Chrome, Microsoft Edge, Opera y Safari para su funcionamiento.
- (h) El idioma usado en la aplicación web debe ser el inglés.
- (i) El software usado para el desarrollo de la aplicación web debe ser de procedencia libre, bajo las licencias GNU, MIT.
- (j) La aplicación web debe ser funcional en los sistemas operativos basados en UNIX como Linux y MacOS, Windows y sistemas operativos móviles como webOS, Android, iOS, y Windows Phone.

4 Diseño Arquitectónico

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.

(a) Vistas de modelo de datos

i. Vista modelo de datos

A. Modelo Researcher

Researcher_db	
PK	Id
	Name(String) Lastname(String) Email(String) Password(String) Study_areas (Array) Organization (String) Nationality (String) Gender(String) Languages (Array) Skills (Array)

Entidades:

- name (String)
Nombre del investigador.
- Lastname (String)
Apellido del investigador.
- Email (String)
Email público del investigador.
- Password (String)
Contraseña de la cuenta del investigador.
- Study Areas (Array)
Áreas en las que el investigador trabaja o se enfoca.
- Organization (String)
Grupo al que pertenece como asociaciones o empresas.
- Nationality (String)
País de origen del investigador.
- Gender (String)
Género del investigador
- Languages(Array)
Idiomas en los que se puede comunicar el investigador.

- **Skills (Array)**
Lista de habilidades que posee el investigador adicionales a las de su área.
- **Relaciones:**
Al ser una única tabla no se tienen relaciones.

B. Modelo Research Group

ResearchGroup_db	
PK	Id
	Topics(Array)
	Website(string)
	Fields(array)
	Logo(string)
	University(String)
	Status(String)
	Skill(String)
	Member(Array)
	Director (String)
	Leader (String)
	Description (String)
	Name(String)

Entidades:

- **Topics (Array)**
Temas en los que se trabaja en el grupo de investigación.
- **Website (String)**
Sitio web del grupo de investigación.
- **Fields (Array)**
Áreas de la ciencia a la que pertenece el grupo de investigación.
- **Logo (Image)**
Logotipo o imagen representativa del grupo de investigación.
- **University (String)**
Nombre de la universidad o institución educativa a la que está adscrito el grupo de investigación.

- Status (string)
Estado del grupo respecto a su actividad y la posibilidad de admitir nuevos miembros.
- Skills (string)
Habilidades las cuales busca el grupo de investigación en los investigadores que quieren aplicar a ser adminitos.
- Members (Array)
Lista de miembros (researchers) que pertenecen al grupo de investigación.
- Director (string)
Nombre del researcher el cual dirige el grupo de investigación.
- Leader (string)
Nombre del researcher que está a cargo del grupo en cuanto a comunicación, admisiones y demás.
- Description (string)
Descripción sobre el grupo de investigación, sus objetivos y demás.
- Name (string)
Nombre titular del grupo de investigación
- **Relaciones:**
Al ser una única tabla no se tienen relaciones.

C. Modelo Project

Project_db	
PK	<u>Id</u>
	Planeacion_id (integer []) Status(String) Miembros(String []) Lider_de_proyecto(String) Titulo(String) Areas_de_estudio(String []) Descripcion(String)

Entidades:

- Planeacion-id (Integer [])
Lista de identificadores de los planes que se tienen en el proyecto.
- Status (string)
Estado en el que se encuentra el proyecto
- Miembros (string [])
Investigadores que están trabajando en el proyecto de investigación.
- Lider-del-proyecto (string)
Nombre de la persona que está a cargo del proyecto.
- Título (string)
Título otorgado al proyecto de investigación.
- Áreas-de-estudio (string [])
Campos en los cuales el proyecto se enfoca en su investigación.
- Descripcion (string)
Descripción de los objetivos del proyecto, ideas en desarrollo y metodologías a seguir.
- **Relaciones:**
Al ser una única tabla no se tienen relaciones.

D. Modelo Planning

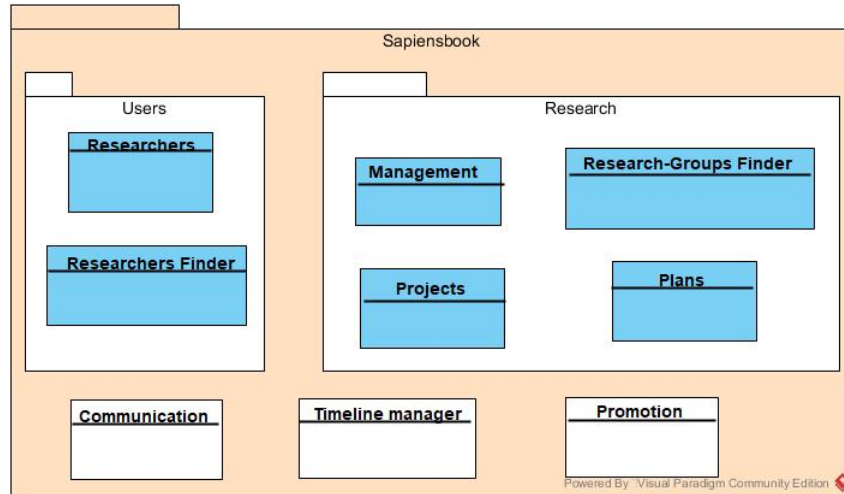
planning_db	
PK	<u>Id</u>
	DayStartPlan(byte) MonthStartPlan(byte) DayEndPlan(byte) idHeadMember(integer) MonthEndPlan(byte) Publish(String) Resources(float) Description(String)

Entidades:

- DayStartDate (byte)
Día en número en el cual se inicia el plan del proyecto.
- MonthStartDate (byte)
Mes en número en el cual se inicia el plan del proyecto.
- DayEndDate (byte)
Día en número en el cual se planea terminar el plan del proyecto.
- MonthEndDate (byte)
Mes en el que se tiene pensado terminar el plan del proyecto de investigación.
- idHeadMember (Integer)
Identificador del investigador a cargo del plan de proyecto.
- Publish (string)
Publicaciones que se han hecho en el plan de proyecto en su desarrollo.
- Resources (float)
Dinero que se tiene disponible como recursos para el desarrollo del pplan , dentro del marco del proyecto.
- Description (string)
Descripción acerca de lo que se quiere lograr con el plan

de investigación, la forma en la que se desarrollará, y el uso de los recursos destinados a ello.

ii. Vista descomposición



- **Elementos:**

- A. **Módulos:**

- **Communication**

- Encargado de las funcionalidades de comunicación en los grupos de investigación entre los investigadores. Se tiene como módulo con único padre el sistema dado que no ha sido desarrollado, sino que se expresa una funcionalidad que se dispondría en futuro.

- **Timeline manager**

- Representa el manejo de lo que se muestra en el home del sistema, como funcionalidad aún no desarrollada.

- **Promotion**

- Representa la funcionalidad de la promoción de determinados grupos dadas algunas características, determinando la posición en la que se muestran en el timeline.

- **Users**

- Engloba los submódulos que tienen las funcionalidades de los usuarios en éste caso los investigadores.

- **Research**
Contiene las funcionalidades referentes a los grupos de investigación. Cada grupo se compone de planes y proyectos.

B. Submódulos:

- **Researchers**
Encargado del manejo de usuarios en cuanto a persistencia y comunicación de dichos registros de datos.
- **Researchers Finder**
Funcionalidad encargada de hacer búsquedas sobre los registros de investigadores.
- **Management**
Módulo de manejo de usuarios, eliminación, actualización y muestra.
- **Projects**
Módulo del manejo de proyectos guardados en un grupo de investigación.
- **Research-groups Finder**
Funcionalidad para buscar sobre los registros de grupos de investigación.
- **Plans**
Planes que hacen parte de un proyecto, un proyecto se compone de planes, siendo así entonces, un plan es parte de un grupo.

C. Sistema:

- **Sapiensbook**
Sistema completo, aplicación web en todo su conjunto.

• Relaciones:

Dado a que es una vista de descomposición, las relaciones son de "es parte de", por lo cual se describen a continuación dichas relaciones:

Relación módulo Users con submódulos researchers y researchers finder

Esta relación muestra que los submódulos researchers y researchers finder hacen parte del módulo Users, en la medida en que contienen funcionalidades relacionadas con el manejo de los usuarios en el sistema, en éste caso, researchers.

Relación módulo Research con submódulos Management, Projects, Plans y Research-groups finder

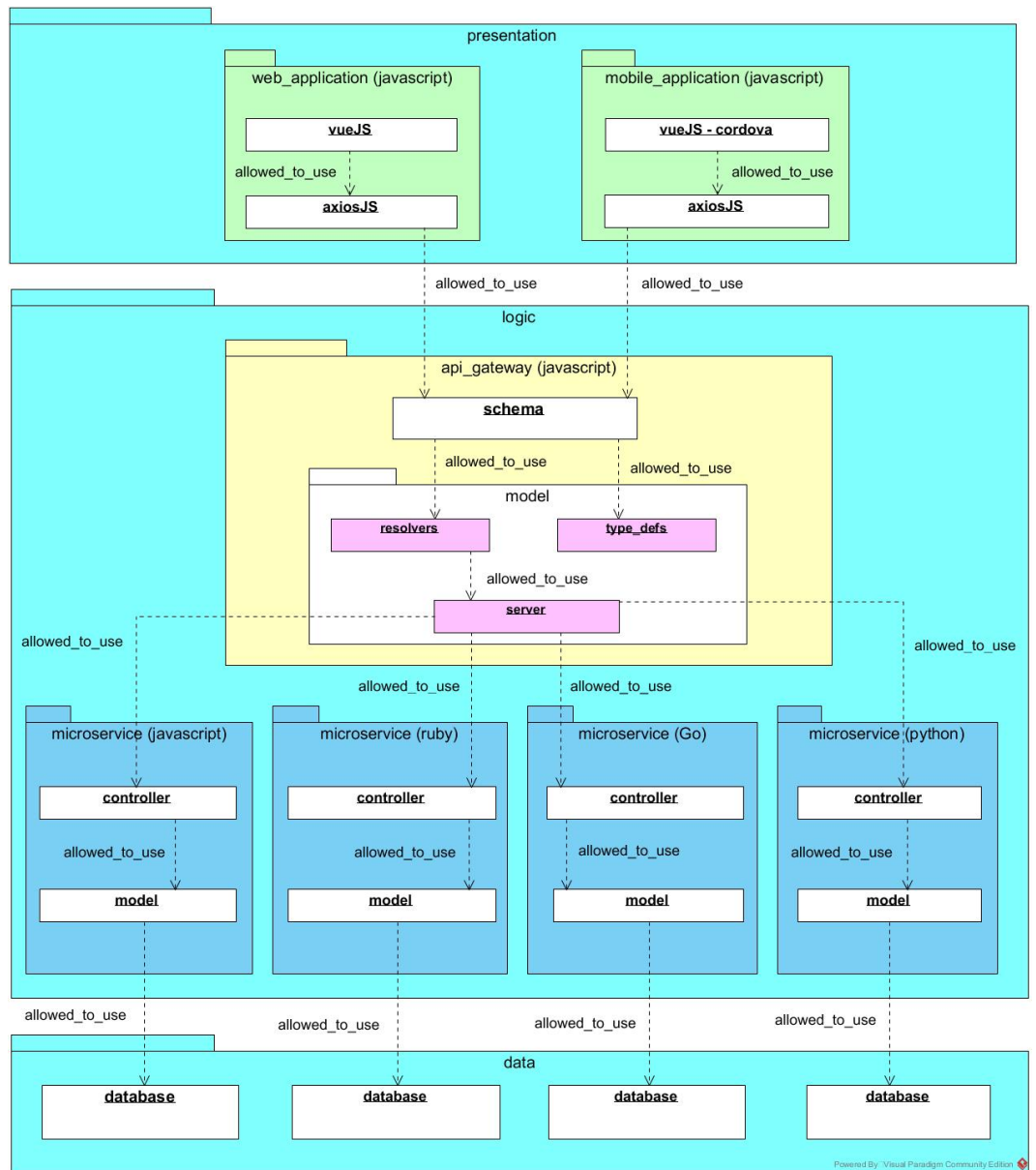
Esta relación muestra que los submódulos management, projects, plans y research-groups finder hacen parte del módulo funcional de research, en donde se contienen los elementos que constituyen la administración, el manejo, la creación y la forma de mostrar los grupos de investigación que se tienen en el sistema.

Relación módulos communication, timeline manager y promotions con sistema Sapiensbook

Esta relación muestra que los módulos mencionados hacen parte del sistema, sin más, esto se da debido a que son módulos con funcionalidades que no se tienen todavía desarrollados, sino que tienen como funcionalidad el proveer de información para la asignación de trabajos en futuro, adicional a mostrar partes del software a modo de estructura, ayudando a tener una mejor perspectiva del sistema en su planteamiento teórico.

Relación Módulos Users y Research con el sistema Sapiensbook

La relación muestra que ambos módulos hacen parte del sistema completo, además de contener a su vez submódulos, configurando así la estructura completa que se plantea para construir el sistema Sapiensbook.



A. Elementos:

- Capas:

- Presentation:

Capa encargada del frintend del sistema, la interacción con el usuario, tiene la relación de "permite usar" hacia abajo, conectándose con al api gateway.

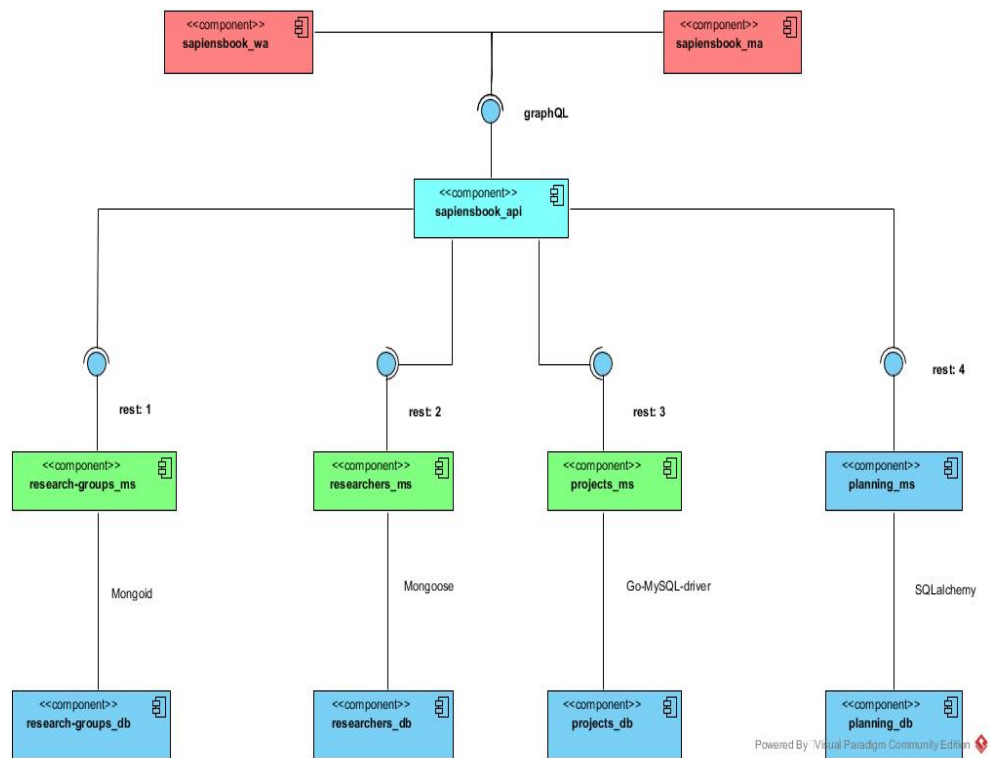
- **Logica:**
 Contiene los microservicios y el api gateway, el cual contiene la lógica de negocio, además de las funcionalidades del sistema. Tienen como relación el tener permitido el uso interno del api gateway hacia los microservicios, y los microservicios hacia las bases de datos.

Tiene como propiedades el contener las funcionalidades lógicas de todo el sistema, así como conexiones internas, como lo es el del api gateway con cada uno de los microservicios para poder armar estructuras de respuesta para la capa de presentación.

- **Data:**
 Contiene las bases de datos a las que se conectan los microservicios.

•

iv. **Vista de componentes y conectores (C&C)**



A. Elementos:

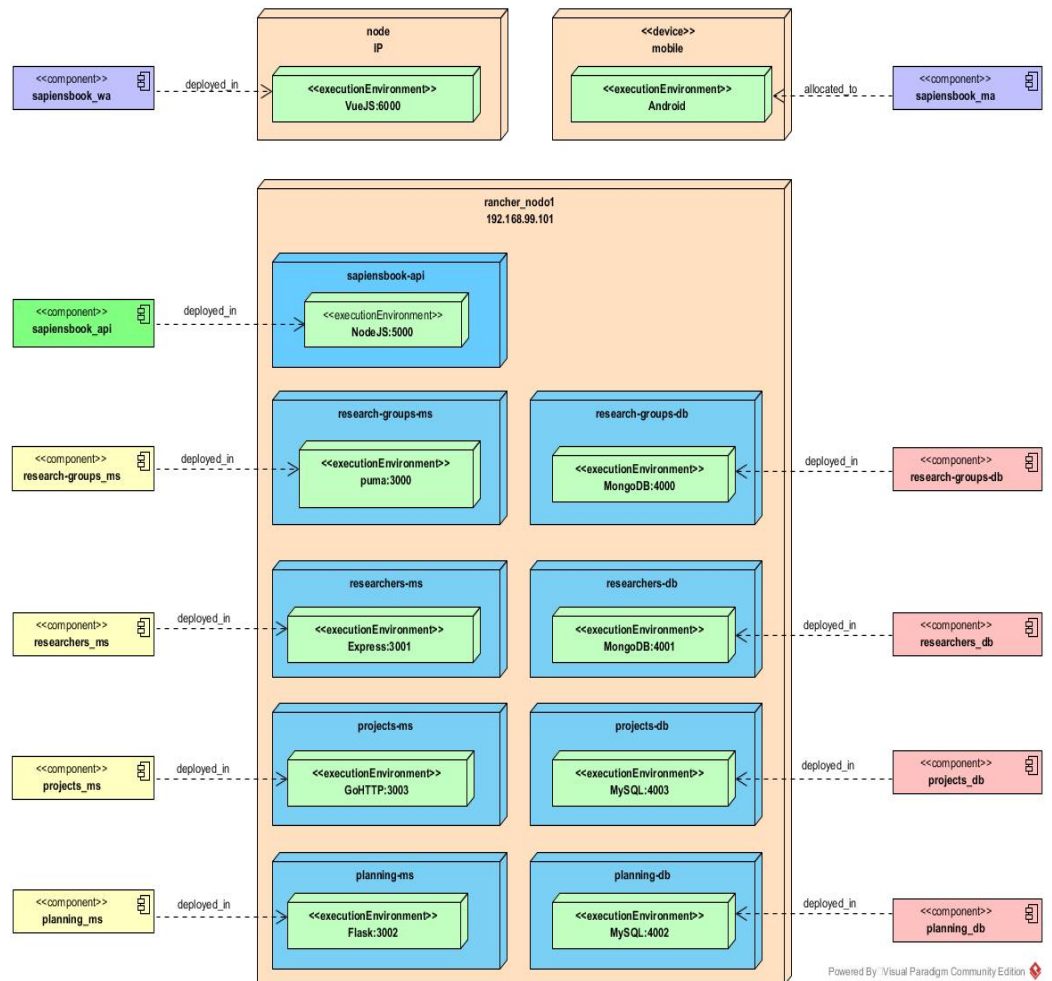
- Componentes:

- **research-groups-db** En cuanto a su interfaz de comunicación, usa puertos. Su conector es el Mongoid el cual mapea desde el microservicio a la base de datos. En su arquitectura, se tiene como patrón propio del componente el monolítico debido a que es altamente acoplado. Adicional tiene como estilo arquitectónico el de cliente servidor debido a que su interfaz actúa como medio de ser el componente el servidor dando datos a un cliente (el microservicio).
- **researcher-db** Su conector es el de mongoose, el cual mapea desde el microservicio la conexión a la base de datos NoSQL MongoDB. Tiene un estilo arquitectónico de cliente servidor debido a que provee datos a un cliente, el cual es el microservicio al que está conectado.

- **project-db** Tiene como conector Go-MySQL.
En su arquitectura tiene un patrón monolítico debido a que es un único componente son conexiones externas.
En estilos arquitectónicos, usa el de cliente servidor, dado a que provee datos traídos de registros en base de datos SQL.
- **plan-db** Tiene como conector SQLAlchemy el cual conecta el microservicio con la base de datos alojada en un modelo relacional SQL.
Usa como estilo arquitectónico el de cliente servidor, donde el cliente es el microservicio al que está conectado, y el servidor siendo el componente plan.db.
- **research-groups-ms** Usa como estilos arquitectónicos el de capas y cliente servidor, ya que está dividido en un modelo para la persistencia de datos, y controladores, los cuales toman las solicitudes del cliente al que está conectado (api gateway).
Usa como patrón MVC, debido a que separa el modelo de los controladores que dirigen acciones dadas determinadas rutas.
- **researcher-ms** De estilos usa el de cliente servidor porque provee servicios al api gateway, y el de capas n-tier debido a que separa por funcionalidades el modelo de los datos y las rutas que solicitan los servicios.
El componente a su vez usa patrón MVC, teniendo divididas las funciones de mapeo de datos en modelo, y el de manejo de solicitudes en el controlador.
- **plan-ms** Usa patrón MVC al separar funcionalidades de manejo de datos y de rutas.
Usa estilo cliente servidor al proveer datos al api gateway.
- **api gateway** tiene un estilo cliente servidor, debido a que actúa como cliente en la medida en que sus conectores, las rest, envían datos a su cliente el api, para luego ser dados como servidor, a los componentes web y móvil.
Usa como conector GraphQL además de herramientas javascript que le permiten organizar y estructurar los datos que toma como cliente de los microservicios.

- **sapiensbook-wa** Componente frontend del sistema en web de navegadores. Usa como patrón el de MVC, con el de vista muy definido dado a que muestra por GUI al usuario la información.
Como estilo usa el de cliente servidor debido a que pide datos del api gateway y los muestra a su cliente el usuario.
- **sapiensbook-ma** da las funciones a móviles, Usa patrón MVC y estilo cliente servidor.

v. Vista de despliegue



Como elemento tiene elementos de entorno y de software. De SW tiene los microservicios y sus respectivas bases de datos así como el api y los componentes de vista web y móvil.

Tiene como relaciones el alojado en.