

UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO
FACULTAD DE INGENIERÍA EN CIENCIAS DE LA
COMPUTACIÓN Y TELECOMUNICACIONES
UNIDAD DE POSTGRADO ESCUELA DE INGENIERÍA



DESARROLLO DE UNA APLICACIÓN ORIENTADO A
COMPONENTE, UTILIZANDO EL MODELO C4 PARA UNA
GESTION DE LIBROS

MODULO:

Análisis y diseño orientado a componentes

INTEGRANTES:

Luis Fernando Vidal Salazar

Jesus Escobar Ovando

Alcides Ticlla Choque

Lorena Churqui Apaza

Santa Cruz de la Sierra – Bolivia

Noviembre – 2019

Contenido

<u>1. INTRODUCCIÓN.....</u>	<u>1</u>
<u>2. OBJETIVO</u>	<u>2</u>
<u>2.1 OBJETIVO GENERAL</u>	<u>2</u>
<u>2.2 OBJETIVOS ESPECÍFICOS.....</u>	<u>2</u>
<u>3. DESCRIPCIÓN DE LA ARQUITECTURA</u>	<u>2</u>
<u>3.1 EXPRESS.....</u>	<u>2</u>
<u>3.2 MONGOOSE.....</u>	<u>3</u>
<u>3.3 MONGODB</u>	<u>3</u>
<u>3.4 MODELO C4 PARA ARQUITECTURA DE SOFTWARE.....</u>	<u>3</u>
<u>3.4.1 NIVEL 1 – CONTEXTO</u>	<u>4</u>
<u>3.4.2 NIVEL 2 – DIAGRAMA DE CONTENEDOR</u>	<u>5</u>
<u>3.4.3 NIVEL 3 – DIAGRAMA DE COMPONENTES.....</u>	<u>5</u>
<u>3.4.4 NIVEL 4 – EL CÓDIGO</u>	<u>5</u>
<u>3.5 VUE</u>	<u>6</u>
<u>4. DESCRIPCIÓN DE LOS ELEMENTOS DEL SISTEMA</u>	<u>6</u>
<u>4.1 DIAGRAMA DE CONTEXTO</u>	<u>6</u>

<u>4.2 DIAGRAMA DE CONTENEDOR.....</u>	<u>8</u>
<u>4.3 DIAGRAMA DE COMPONENTE</u>	<u>10</u>
<u>4.4 DIAGRAMA DE CÓDIGO</u>	<u>12</u>
<u>BIBLIOGRAFÍA</u>	<u>13</u>

Lista de figuras

Ilustración 1 Diagrama de Contexto	7
Ilustración 2 Diagrama de Contenedor	9
Ilustración 3 Diagrama de Componente.....	11
Ilustración 5 Componente Autor	12
Ilustración 4 Componente Perfil.....	12
Ilustración 6 Componente Libro	12
Ilustración 7 Componente Session	12

1. Introducción

Todo sistema de gestión necesita de componentes para su total funcionalidad, dichos componentes pueden ser desarrollados para su uso interno o externo, es decir el componente desarrollado puede ser utilizado dentro o fuera de la compañía, como caso de estudio en este proyecto proponemos desarrollar un par de componentes para administrar el inicio de sesión, gestión de libros , autor y la administración de perfil.

Las personas que navegan por internet para poder utilizar redes sociales, herramientas de trabajo (Aplicaciones web), Blogs, etc. es necesario que tenga que crear una cuenta de usuario por cada una de estas, lo que significa que el número de cuentas de usuarios crece exponencialmente en función a los números de estos servicios disponibles en la nube, por lo que se dificulta el poder acordarse de todas.

Para este caso experimental de programación orientada a componentes se pretende crear un gestor de cuentas de usuario, la cual consiste en que el usuario pueda crear una cuenta de usuario por una sola vez, y con esta poder iniciar sesión en todas las web app que consuman el servicio del componente.

2. Objetivo

2.1 Objetivo General

Realizar el desarrollo de una aplicación orientado a componente, utilizando el modelo C4 para una gestión de libros.

2.2 Objetivos Específicos

- Desarrollar componente API SessionDS.
- Desarrollar componente API ProfileDS.
- Desarrollar componente API AutorDS.
- Desarrollar componente API LibrosDS.
- Desarrollar una aplicación web para la gestión de libros.
- Desarrollar una aplicación web para la gestión de usuarios.

3. Descripción de la Arquitectura

3.1 Express

Express es un framework web transigente, escrito en JavaScript y alojado dentro del entorno de ejecución NodeJS. El módulo explica algunos de los beneficios clave de este framework, como configurar tu entorno de desarrollo y realizar tareas comunes en desarrollo y publicación web.

https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/

3.2 Mongoose

Mongoose es una librería de Modelado de Datos de Objetos (ODM) para MongoDB y Node.js. Gestiona las relaciones entre los datos, proporciona validación de esquemas y se utiliza para traducir entre los objetos en código y la representación de esos objetos en MongoDB.

3.3 MongoDB

MongoDB es una base de datos de documentos NoSQL sin esquemas. Esto significa que puede almacenar documentos JSON en él, y la estructura de estos documentos puede variar ya que no se aplica como en las bases de datos SQL. Esta es una de las ventajas de usar NoSQL ya que acelera el desarrollo de aplicaciones y reduce la complejidad de las implementaciones.

<https://www.freecodecamp.org/news/introduction-to-mongoose-for-mongodb-d2a7aa593c57/>

3.4 Modelo C4 para Arquitectura de Software

Según la naturaleza del proyecto y de la aplicación se utilizará Modelo C4 para la descripción del diseño.

C4 es una alternativa para enfoques de arriba hacia abajo donde se empieza por definir un Contexto, Contenedor, Componentes y Código.

El modelo C4 consiste en un conjunto jerárquico de diagramas de arquitectura de software para contexto, contenedores, componentes y código.

La jerarquía de los diagramas C4 proporciona diferentes niveles de abstracción, cada uno de los cuales es relevante para una audiencia diferente.

Evite la ambigüedad en sus diagramas incluyendo una cantidad suficiente de texto, así como una clave o leyenda para la notación utilizada.

Para crear estos mapas de su código, primero necesita un conjunto común de abstracciones para crear un lenguaje ubicuo que describa la estructura estática de un sistema de software. El modelo C4 considera las estructuras estáticas de un sistema de software en términos de contenedores (aplicaciones, almacenes de datos, microservicios, etc.), componentes y código.

También considera a las personas que utilizan los sistemas de software que construimos.

3.4.1 Nivel 1 – Contexto

El Nivel 1, un diagrama de contexto del sistema, que muestra el sistema de software que está construyendo y cómo encaja en el mundo en términos de las personas que lo utilizan y los otros sistemas de software con los que interactúa.

El código de color en el diagrama indica los sistemas de software que ya existen (las cajas grises) y los que se van a construir (azules).

3.4.2 Nivel 2 – Diagrama de Contenedor

El nivel 2, un diagrama de contenedor, amplía el sistema de software y muestra los contenedores (aplicaciones, almacenamiento de datos, microservicios, etc.) que componen este sistema de software. Las decisiones tecnológicas son también una parte fundamental de este diagrama.

3.4.3 Nivel 3 – Diagrama de Componentes

El nivel 3, un diagrama de componentes, expande un contenedor individual para mostrar los componentes que contiene. Estos componentes deben asignarse a abstracciones reales (por ejemplo, una agrupación de códigos) en función de su código.

3.4.4 Nivel 4 – El Código

El Nivel 4, Este es un diagrama de clases UML, para ampliar un componente individual para mostrar cómo se implementa este componente.

3.5 Vue

Vue (pronounced /vju:/, like view) es un marco progresivo para construir interfaces de usuario. A diferencia de otros marcos monolíticos, Vue está diseñado desde cero para ser gradualmente adoptable. La biblioteca principal se centra solo en la capa de vista y es fácil de recoger e integrar con otras bibliotecas o proyectos existentes. Por otro lado, Vue también es perfectamente capaz de impulsar aplicaciones sofisticadas de una sola página cuando se usa en combinación con herramientas modernas y bibliotecas de soporte.

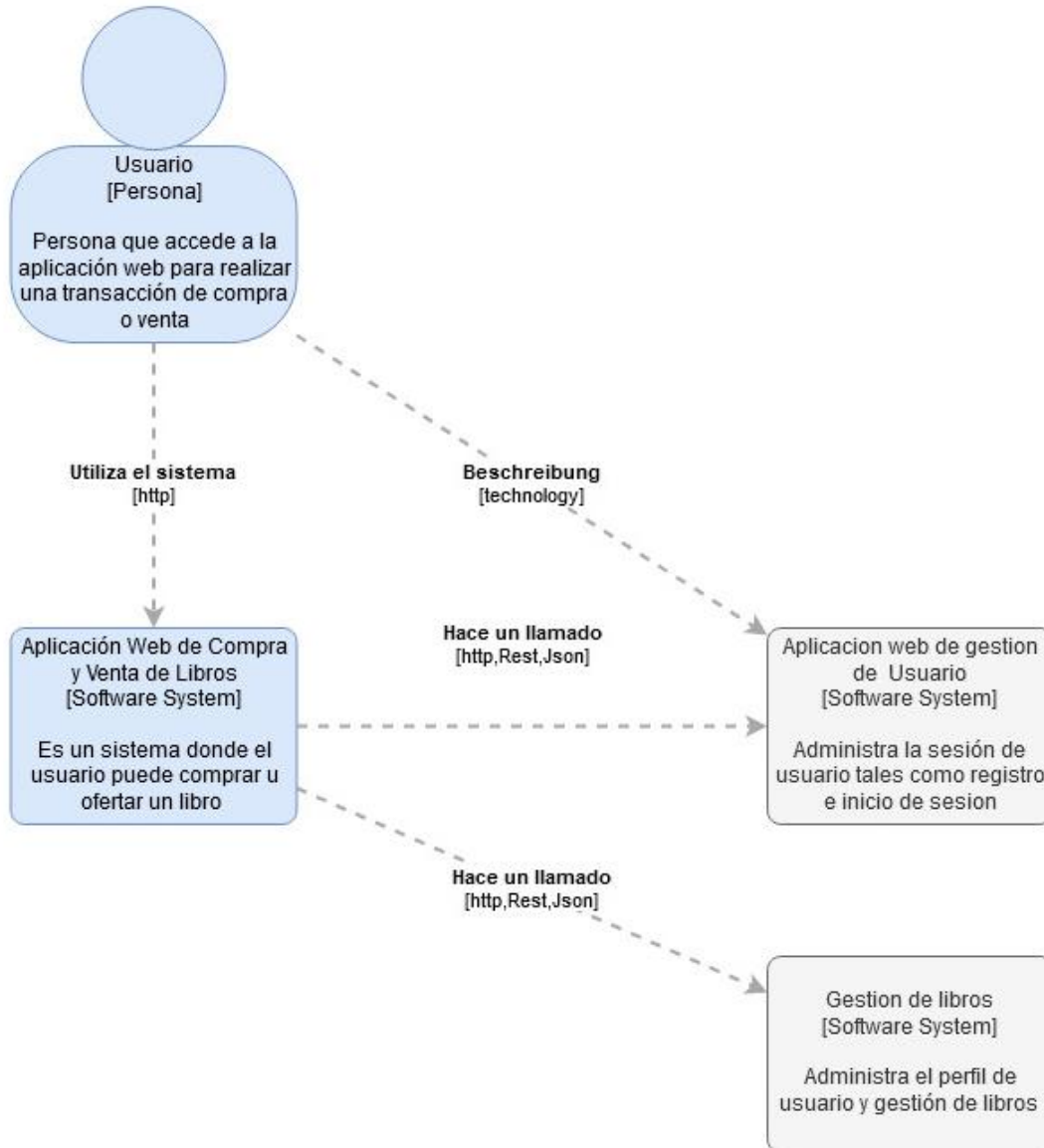
4. Descripción de los elementos del sistema

Para hacer la descripción del sistema utilizaremos el modelo C4.

4.1 Diagrama de Contexto

El proyecto tendrá como aplicativo una aplicación web donde el usuario puede ofertar y comprar libros, dicha aplicación hace uso de dos componentes principales, Autenticación de usuario y Perfil de usuario, el componente de Autenticación de usuario gestionará la sesión del usuario tales como el registro (sign up) y el inicio de sesión (log in), el componente Perfil de usuario administrará los datos de perfil de usuario tales como el nombre completo, ciudad de origen, entre otros.

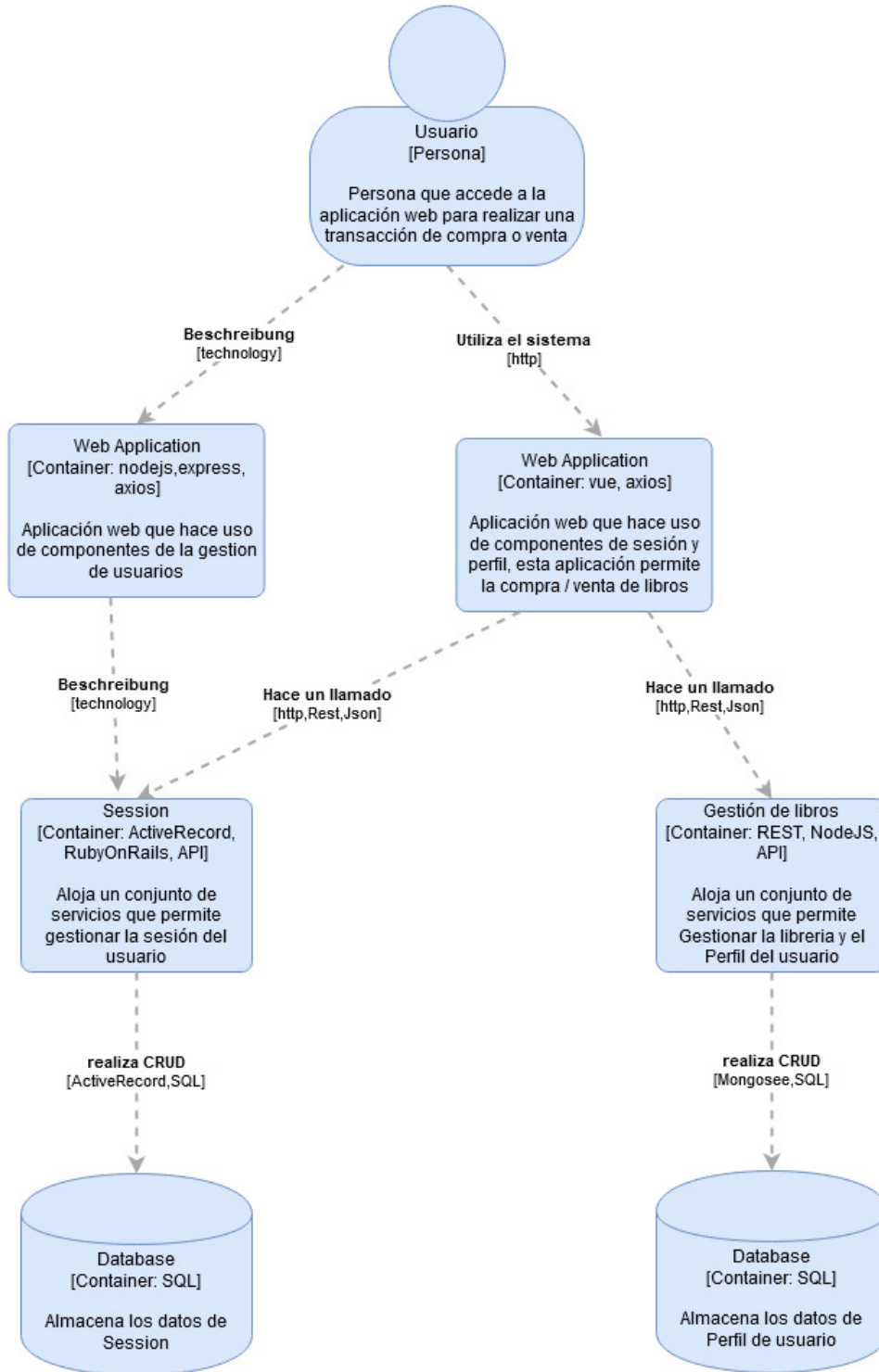
Ilustración 1 Diagrama de Contexto



4.2 Diagrama de Contenedor

El siguiente diagrama muestra la interacción que tiene el sistema entre contenedores, Sesion y gestión de libros y las aplicaciones web, dichos Componentes son independientes al sistema e interactúan entre sí de manera indirecta, ambos componentes fueron desarrollados en diferentes tecnologías, la aplicación web consume dichos componentes a través de servicios a través del protocolo HTTP.

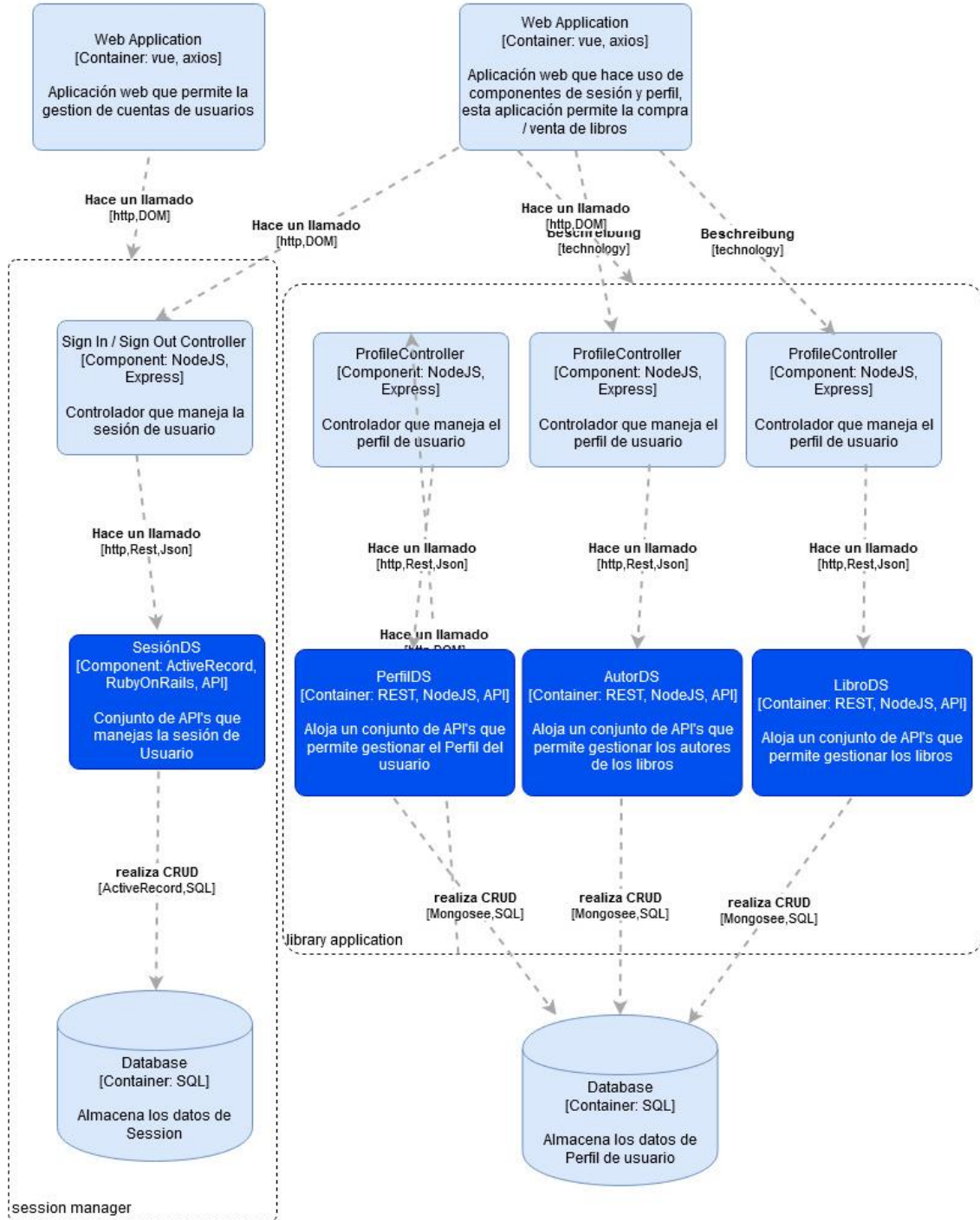
Ilustración 2 Diagrama de Contenedor



4.3 Diagrama de Componente

El siguiente diagrama describe los componentes a ser desarrollados, la aplicación Web consulte los componentes a través de controladores, SessionDS contiene el conjunto de APIs para gestionar la sesión, así mismo ProfileDS y libroDS, autorDs.

Ilustración 3 Diagrama de Componente



4.4 Diagrama de Código

Cada una de las imágenes representa a un componente de la aplicación.

Ilustración 5 Componente Perfil



Ilustración 4 Componente Autor

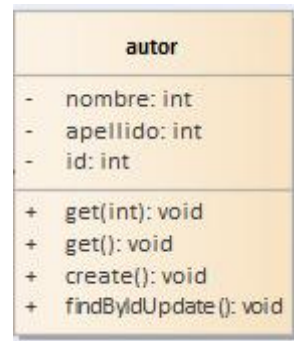
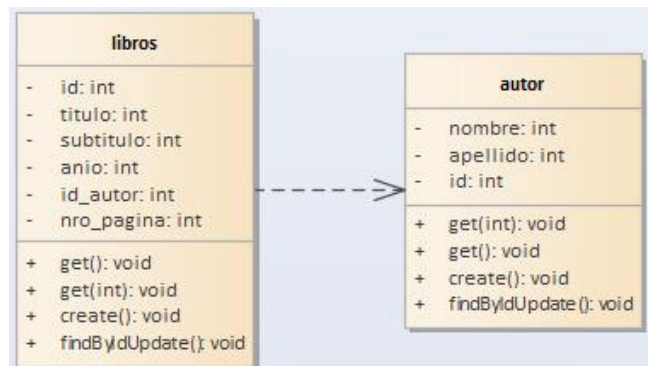


Ilustración 7 Componente Session



Ilustración 6 Componente Libro



Bibliografía

C4model.com. (2019). The C4 model for visualising software architecture. [online] Available at: <https://c4model.com/> [Accessed 16 Nov. 2019].

Vuejs.org. (2019). Introduction — Vue.js. [online] Available at: <https://vuejs.org/v2/guide/#What-is-Vue-js> [Accessed 16 Nov. 2019].

Node.js, F. (2019). Node.js. [online] Node.js. Available at: <https://nodejs.org/es/> [Accessed 16 Nov. 2019].

MongoDB. (2019). The most popular database for modern apps. [online] Available at: <https://www.mongodb.com/> [Accessed 16 Nov. 2019].