



UNIVERSIDAD SANTO TOMÁS  
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA  
T U N J A

**PLATAFORMA CRAI USTA TUNJA CONMUTACION BIBLIOGRAFICA  
PROYECTO RIBID (RED INTERBIBLIOTECARIA DE INTERCAMBIO DE  
DOCUMENTOS)**

**PROPONENTE**

Julián Daniel Franco Alvarado  
CC. 1054096225  
Cod. 2251207

**DIRECTOR**

Iván Fernando Fonseca Barinas

Tunja  
29 de noviembre del 2021

## CONTENIDO

|                                    |                                      |           |
|------------------------------------|--------------------------------------|-----------|
| <b><u>1.</u></b>                   | 4                                    |           |
| <b><u>2.</u></b>                   | 5                                    |           |
| <b><u>3.</u></b>                   | 6                                    |           |
| <b><u>4.</u></b>                   | 7                                    |           |
| <b>OBJETIVO GENERAL</b>            |                                      | <b>7</b>  |
| <b>OBJETIVOS ESPECÍFICOS</b>       |                                      | <b>7</b>  |
| <b><u>5.</u></b>                   | <b>¡Error! Marcador no definido.</b> |           |
| <b><u>6.</u></b>                   | <b>¡Error! Marcador no definido.</b> |           |
| <b>6.1. ANÁLISIS DE REQUISITOS</b> |                                      | <b>13</b> |
| ■                                  | 8                                    |           |
| ○                                  | 10                                   |           |
| ○                                  | 10                                   |           |
| ■                                  | 12                                   |           |
| ■                                  | 15                                   |           |
| 6.2.4. INTERFAZ DE USUARIO         |                                      | 19        |
| <b>6.3. FASE 3. IMPLEMENTACIÓN</b> |                                      | <b>21</b> |
| ■                                  | 18                                   |           |
| ■                                  | 19                                   |           |
| ■                                  | 19                                   |           |
| <b><u>7.</u></b>                   | 22                                   |           |

**8.** 25

o 25

o 25

o 26

**9.** 27

**10.** 28

**11.** 30

■ 30

■ 30

■ 30

■ 30

■ 30

■ 30

## ÍNDICE DE TABLAS

|                                       |    |
|---------------------------------------|----|
| Tabla 1 Objetivos específicos         | 10 |
| Tabla 3 Fase 2. Diseño y Arquitectura | 24 |
| Tabla 4 Colores                       | 26 |
| Tabla 5 Fase 3. Implementación        | 33 |
| Tabla 6 Cronograma de actividades     | 36 |
| Tabla 7 Resumen del presupuesto       | 38 |
| Tabla 8 Recurso humano                | 38 |
| Tabla 9 Recurso material y/o equipos  | 39 |

|                                                                                                                                                                                                                           |                                                                      |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|
| Título                                                                                                                                                                                                                    |                                                                      |
| Nombre Estudiante                                                                                                                                                                                                         | Julián Daniel Franco Alvarado                                        |
| Documento estudiante                                                                                                                                                                                                      | 1054096225                                                           |
| Correo electrónico                                                                                                                                                                                                        | Julian.franco@usantoto.edu.co                                        |
| Director                                                                                                                                                                                                                  | Iván Fernando Fonseca Barinas                                        |
| Lugar de ejecución del proyecto                                                                                                                                                                                           | Universidad Santo Tomás Tunja                                        |
| Duración                                                                                                                                                                                                                  | 12 meses                                                             |
| Costo                                                                                                                                                                                                                     | \$30.215.000                                                         |
| Palabras claves                                                                                                                                                                                                           | Documentos, compartir, plataforma, convenios, recuperar, visualizar. |
| <p>Los abajo firmantes confirman que todos los datos incluidos en la presente propuesta son correctos y verídicos, que no incumplen ninguna ley o norma vigente (incluir nombres y firmas de estudiantes y director).</p> |                                                                      |
| <p>Firma del autor</p> <p>Julián Daniel Franco Alvarado</p>                                                                                                                                                               |                                                                      |
| <p>Firma del director</p> <p>Iván Fernando Fonseca Barinas</p>                                                                                                                                                            |                                                                      |

## **2. PLANTEAMIENTO DEL PROBLEMA**

El Centro de Recursos para el Aprendizaje la Investigación (CRAI) de la Universidad Santo Tomas seccional Tunja, crea un sistema de intercambio de documentos que opera con procesos manuales en el que estudiantes, investigadores y docentes realizan solicitudes de forma presencial y en formularios rústicos de papel bond, lo cual dicho proceso, en muchos casos se convierte en una labor incomoda, puesto que cada vez que se realiza un solicitud de algún documento el dar notificación de que ya está disponible el mismo es más complejo, y en la mayoría de los casos no se completan las solicitudes, por falta de agilidad en la entrega de la misma documentación.

Dicho lo anterior, es fundamental la automatización y optimización de dicho proceso, entendiendo que se requiere fundamentalmente del proceso de intercambio de documentos para el correcto funcionamiento del CRAI, prestando un servicio óptimo para la comunidad educativa y evitar que la falta de agilidad en la entrega de los documentos requeridos se convierta en un obstáculo para la buena labor y las metas del CRAI.

### **3. JUSTIFICACIÓN**

Este proyecto es de fundamental importancia para el Centro de Recursos para el Aprendizaje y la Investigación de la Universidad Santo Tomas seccional Tunja, ya que busca aliviar lo tedioso que pueden ser los procesos manuales de intercambio de documentos y que pueden ser facilitados a través de herramientas tecnológicas, así mismo, permite que factores como el tiempo, la entrega oportuna de documentos, y la facilidad de visualización de los mismo por una plataforma digital, puedan favorecer a la comunidad educativa que más solicita documentos. Por ende el proyecto brinda un alto valor para facilitar tanto los procesos para los estudiantes e investigadores que solicitan documentos como para los funcionarios del CRAI encargados de dar funcionamiento manual al dicho proceso, mejorando su calidad de vida.

#### 4. OBJETIVOS

##### Objetivo General

1. Mejorar el proceso de intercambio de documentos entre la comunidad educativa por medio de la construcción un sistema informático enfocado en la web, utilizando el patrón de arquitectura de software MVC (modelo, vista, controlador), junto con el paradigma orientado a objetos, y tecnologías web más usadas en la industria.
- 2.

##### Objetivos específicos

1. Tabla 1. Objetivos específicos

| Nro. | Objetivo específico                                                                                                                                                                                                                                     |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | Diseñar los diagramas del sistema MVC (modelo, vista, controlador) según el sistema propuesto usando el estándar UML.                                                                                                                                   |
| 2    | Desarrollar una aplicación web que cumpla con cada uno de los requerimientos, y que cumpla con los diseños elaborados, usando tecnologías de software libre, herramientas de ingeniería de software y el paradigma de programación orientada a objetos. |
| 3    | Implementar la aplicación web al servicio de la comunidad educativa 24/7 para un libre y responsable, cumpliendo los términos y condiciones de uso del sistema.                                                                                         |

Fuente: Autor

## 5. METODOLOGIA

A continuación, se detalla cada una de las actividades realizadas en las diferentes fases del proyecto:

### 6.1. Análisis de requisitos

El desarrollo de la fase de análisis de requisitos estuvo por las actividades de licitación, validación de requisitos y posterior validación de los mismos, teniendo en cuenta el objetivo de comprender la lógica de negocio que debía cumplir el software

#### ■ 6.1.1. Análisis de planteamiento de arquitectura

El primer acercamiento fue a través de una serie de entrevistas con el director del CRAI Carlos Canoles donde se plantearon preguntas sobre el cómo se quería el funcionamiento de la plataforma, los actores involucrados y los diferentes procesos que se llevan a cabo en las áreas del intercambio de documentos.

A partir de la recolección de la información en el respectivo documento de requerimientos se listan las historias de usuario y se realizó el primer diseño lógico de la base datos según los requerimientos encontrados.

Para luego realizar la respectiva verificación de cada una de las tablas contenidas en el modelo entidad relación, y comprender a detalle los requisitos funcionales que

el software debía cumplir, y seguir con la fase de comparación de los requisitos y que se estuviera de acuerdo al planteamiento de los mismos.

### 6.1.2. Integración y creación repositorios en Github

Se dispuso un sistema compartido de código como Github donde se administraron las distintas versiones del software en este caso en concreto backend y frontend en donde se realiza la identificación de las ramas de trabajo y por qué usuarios van a ser administradas.

También se realizó la identificación y organización de cada una de las ramas de desarrollo donde estará alojado el código fuente.

A continuación, se muestran las entradas, instrumentos/herramientas, entregables y tiempo de la fase de análisis de historias de usuario.

Tabla 2 Fase 1. Herramientas de uso principal

| Tipo                          | Elementos                                                                                                          |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Característica                | <ul style="list-style-type: none"><li>● Planteamiento arquitectura</li></ul>                                       |
| Instrumentos y/o herramientas | <ul style="list-style-type: none"><li>● Computador</li><li>● Conexión a internet</li><li>● Navegador Web</li></ul> |
| Entregables                   | Links Github de repositorios                                                                                       |
| Tiempo                        | 1 semana                                                                                                           |

Fuente: Autor

○ **6.2. Fase 2. Diseño general**

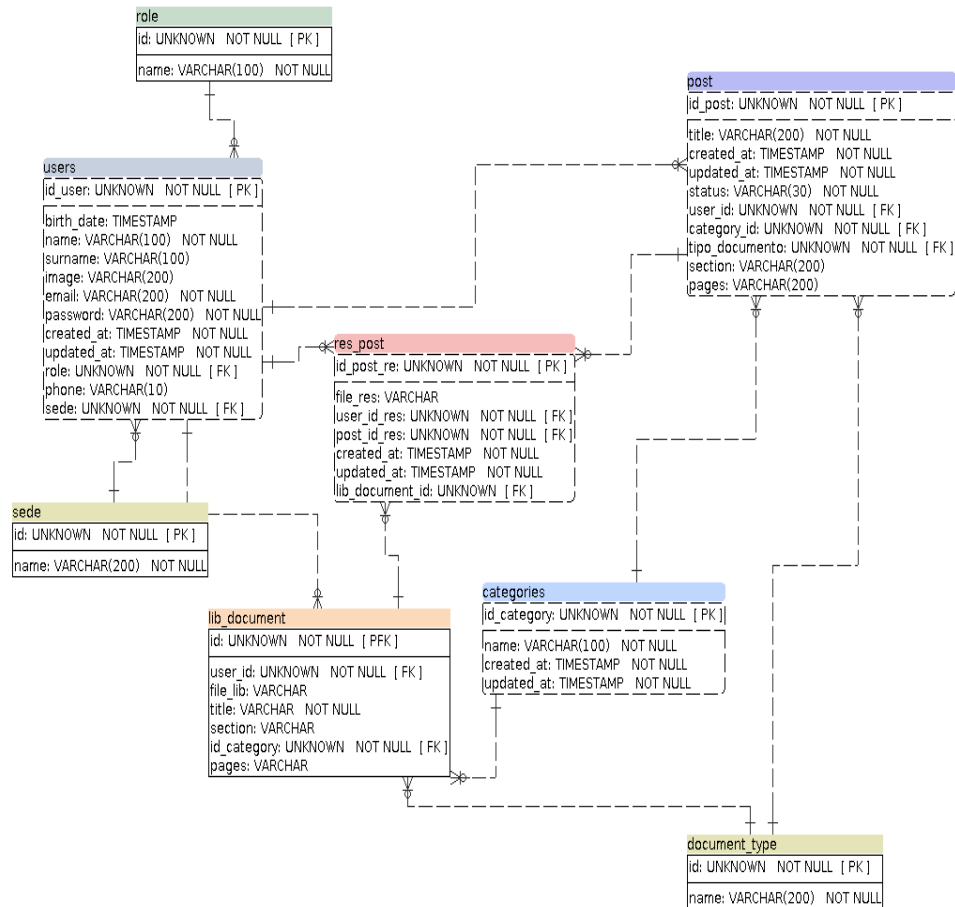
El desarrollo de la fase de diseño y arquitectura estuvo realizado por las siguientes actividades principales: el diseño del modelo entidad relación y el diseño de la arquitectura.

**a. 6.2.1. Diseño de la base de datos**

A continuación, se presenta el diseño entidad-relación de la base de datos, está compuesto por las siguientes tablas:

- role
- users
- sede
- Res\_post
- Lib\_document
- post
- categories
- Document\_type

### Ilustración 1 Diseño de la base de Datos



Fuente: Autor

Determinando que cada una de las tablas cumplieran con una función estratégica para el funcionamiento adecuado de la lógica de negocio que se planteó como solución para la distribución eficiente de documentos y la oportunidad de una escalabilidad mucho más sencilla en la ampliación y agregación de otras tablas que complementen más lógica de negocio.

### ■ 6.2.2. Diseño de la arquitectura

Se determinó que la mejor opción para la construcción del Backend era el uso del lenguaje de programación PHP, junto con el Framework de desarrollo empresarial Laravel usado en entornos empresariales y de gran facilidad para crear servicios REST.

Respecto al motor de base de datos se eligió PostgreSQL. Esta decisión se realizó debido a que este motor de base de datos es relacional, gratuito, seguro, ampliamente usado en el mundo empresarial, además cuenta con buena documentación y soporte.

También implementando un sistema para autenticar como JWT que es sencillo, rápido y muy seguro para la comunicación con el cliente y servidor, además de un

uso escalado en caso de que se requiera escalabilidad y como copia de seguridad espejo de la base de datos de PostgreSQL.

Este tipo de arquitectura tiene ventajas empresariales y de escalabilidad como:

- Fácil acceso a endpoints del servidor para que cualquier cliente pueda acceder al uso de la plataforma sea desde una aplicación web adaptado para dispositivos móviles
- Se tienen registros de seguridad más confiables con el uso del sistema para autenticar con jwt, teniendo una mejor calidad y rapidez en las respuestas del servidor.
- Creando un sistema de seguridad completo con métodos completamente reservados en helpers de Lavarvel para mantener la confidencialidad de los métodos de autenticación.
- Teniendo en cuenta métodos para la completa validación del token y evitar la falsificación o suplantación del mismo, manteniendo la integridad en seguridad para el usuario que ha ingresado al sistema.

Ilustración 2 Código fuente de la función “Signup”

```
public function signup($email, $password, $getToken = null)
{
    $user = User::where([
        'email' => $email,
        'password' => $password
    ])->first();

    $signup = false;
    if (is_object($user)) {
        $signup = true;
    }

    if ($signup) {
        $token = array(
            'sub' => $user->id,
            'email' => $user->email,
            'name' => $user->name,
            'surname' => $user->surname,
            'image' => $user->image,
            'role' => $user->role,
            'sede' => $user->sede,
            'phone' => $user->phone,
            'birth_date' => $user->birth_date,
            'iat' => time(),
            'exp' => time() + (7 * 24 * 60 * 60)
        );

        $jwt = JWT::encode($token, $this->key, 'HS256');
        $decoded = JWT::decode($jwt, $this->key, ['HS256']);

        if (is_null($getToken)) {
            $data = $jwt;
        } else {
            $data = $decoded;
        }
    } else {
        $data = array(
            'status' => "error",
            'code' => 400,
            'message' => 'Login Incorrecto'
        );
    }
    return $data;
}
```

Fuente: Autor

### ■ 6.2.3. Módulos

Dado que el sistema busca suplir de una forma eficiente y rápida la solicitud y entrega de documentos entre la comunidad educativa según los roles que interpretan dentro del sistema se obtuvieron los siguiente módulos:

#### **Módulo de registro de usuarios.**

Existen tres tipos de usuarios en la plataforma como:

**Bibliotecario:** Tiene los permisos de responder solicitudes de documentos hechos por docentes y estudiantes según su sede de la universidad.

**Docente:** Tiene los permisos para editar su cuenta y crear tantas solicitudes como le sea necesario, además de descargar los documentos.

**Estudiante:** Tiene los permisos para editar su cuenta y crear tantas solicitudes como le sea necesario.

#### **Módulo de solicitud de documentos.**

**Estudiante y docente:** El estudiante o docente registrado tiene la opción en la plataforma de solicitar un documento mediante un formulario donde se especifique el nombre del documento, el tipo del documento, y la categoría del mismo con el fin de obtener el material específico.

#### **Módulo de visualización de documentos.**

**Estudiante y docente:** En este módulo se le muestra al estudiante o docente, por parte del bibliotecario de la sede a la que se le fue solicitado el documento, el

material que le fue solicitado, mostrando la pantalla como un archivo tipo PDF renderizado dentro de la misma plataforma.

#### ▪ 6.2.4. Interfaz de usuario

Entendiendo las necesidades en el apartado de las solicitudes y entregas de documentos se plantea un UI/UX de lo más eficiente en la parte visual, con características responsivas para el uso en múltiples dispositivos con diferentes tamaños de pantalla para una accesibilidad apta para cualquier tipo de usuario sin importar el tamaño de pantalla a la que acceda al sistema.

Planteando en el UX una experiencia sencilla de navegabilidad donde el usuario pueda tener fácil acceso a cualquier característica que tenga disponible según su papel dentro del sistema.

Partiendo desde una interfaz que integre lo más minimalista posible, se planteó el uso del Framework de interfaz Bootstrap con una mezcla de colores claros para el fondo, y colores oscuros predominantes sobre el blanco, mejorando la accesibilidad, dando detalles con colores azules predominantes para elementos de acción para el usuario como botones, etiquetas, textos, entre otros.

Tabla 4 Explicación Paleta de Colores

| Color  | Función                                                                    |
|--------|----------------------------------------------------------------------------|
| Gris   | Resalta sobre el color blanco del fondo para una mejor armonía de colores. |
| Blanco | Evoca pureza, inocencia y positividad.<br>Ayuda a limpiar el diseño        |
| Azul   | Evoca calma, confianza, honestidad.                                        |

Fuente: Autor

Por consiguiente, los anteriores colores se utilizaron para:

- Morado: botones, acciones actuales, títulos principales
- Fondo blanco
- Gris: resaltar selecciones, opción de acciones y menús

Un ejemplo del uso de los colores se muestra en la siguiente imagen:

Ilustración 3 Diseño del formulario de autenticación

hace

er y s

ótitulos

**Centro de Recursos para el Aprendizaje**

Universidad Santo Tomas

Nombre

El nombre no es valido

Correo

Contraseña

Al registrarse, acepta nuestros [Terminos de uso](#).

**REGISTRAME**

o

¿Ya tienes una cuenta? [Inicia aqui](#)

V - Jul

V - P

Fuente: Autor

### 6.3. Fase 3. Implementación

Esta arquitectura se compone por la arquitectura SaaS implementando en el “Backend” un servicio que gestiona cada uno de los requerimientos para proveer la información y un “Frontend” o cliente que hace peticiones y consume datos del

servicio según las solicitudes del mismo. El desarrollo en la fase de implementación estuvo compuesto de las siguientes actividades:

#### ■ 6.3.1. Desarrollo del Servicio

Toda la implementación del servicio se construirá sobre el framework Laravel el cual permitió modularizar el servicio sobre diferentes rutas, controladores y modelos que estaban serializados respectivamente de acuerdo con las tablas de la base de datos, permitiendo un manejo simple y escalable de este framework empresarial.

La arquitectura de servicio que brinda las respuestas en los respectivos adjetivos de la arquitectura REST son los siguientes:

- Los modelos: gestionan las relaciones entre las tablas de la base de datos y el Framework.
- Controladores: Permiten crear, consultar y almacenar los resultados de las consultas de la base de datos con instancias de modelo y respuestas respectivas.
- Rutas: Son las direcciones a las que el cliente se va dirigir para obtener la información según lo necesite, todo en formato JSON.

Según lo requerimientos será necesario de un servicio para que en futuro otros clientes puedan acceder a la información y hacer su respectiva manipulación, dado que puede escalar a tecnologías móviles, la mejor y más eficiente arquitectura era

SaaS, dando una buena implementación en seguridad y confiabilidad en la misma escalabilidad

#### ■ 6.3.1.1 Buenas prácticas de programación

Siguiendo las buenas prácticas de programación se hizo uso de los principios SOLID que de acuerdo con las normas fundamentales de la programación y el diseño orientados a objetos es una práctica confiable. (Rubira, 2021).

Un Ejemplo del principio SOLID se puede observar en el siguiente fragmento de código.

Ilustración 4 Código fuente de la clase “ResPost”

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class ResPost extends Model
9 {
10     use HasFactory;
11
12     protected $table = "res_post";
13
14     protected $fillable = [
15         "id",
16         "file_res",
17         "user_id_res",
18         "post_id_res",
19         "lib_document_id"
20     ];
21
22     public function user(){
23         return $this->belongsTo( related: "App\Models\User", foreignKey: "user_id_res");
24     }
25
26     public function post(){
27         return $this->belongsTo( related: "App\Models\Post", foreignKey: "post_id_res");
28     }
29
30     public function libDocument(){
31         return $this->belongsTo( related: "App\Models\LibDocument", foreignKey: "lib_document_id");
32     }
33 }
```

Fuente: Autor

#### 6.3.2. Desarrollo del Frontend

Para la arquitectura Frontend del cliente web se construirá usando el Framework para manejo de vistas Angular, siguiendo buenas prácticas de

programación teniendo en cuenta los elementos que componen el marco de trabajo como:

- `app.component`: Contiene el componente principal donde se va a situar toda la single page app y el routing.
- Componentes: En este directorio se van situar todos y cada uno de los componentes que se van a mostrar al usuario.
- Servicios: En este directorio se van a ubicar todos los servicios que van a hacer peticiones y van a gestionar la información en los componentes.
- `App.module.ts`: Este módulo situado como uno de los más importantes ya que permitirá administrar los módulos de librerías externas que se agreguen a la aplicación.
- `App.routing.ts`: Este archivo nos permitirá manejar todo lo relacionado con las rutas y sus distintas utilidades para con la aplicación.
- `Styles.css`: estilos CSS globales personalizados que se utilizan en componentes y páginas.
- Assets: Este directorio en particular nos permite almacenar las utilidades estáticas como imagen etc, para el uso en la aplicación.

A continuación, se muestran las entradas, instrumentos/herramientas, entregables y tiempo de la fase de Implementación.

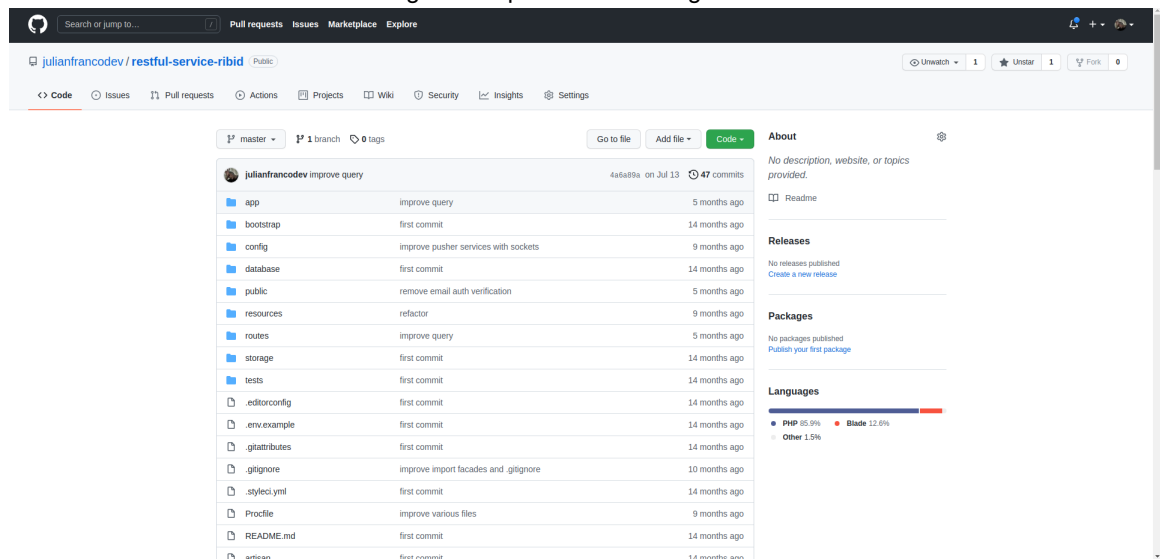
Tabla 5 Fase 3. Implementación

| Tipo                          | Elementos                                                                                                                                                                                                |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Entradas                      | <ul style="list-style-type: none"> <li>● Archivo .ldm del diseño entidad-relación de la base de datos</li> </ul>                                                                                         |
| Instrumentos y/o herramientas | <ul style="list-style-type: none"> <li>● Computador</li> <li>● Conexión a internet</li> <li>● Navegador Web</li> <li>● Github</li> <li>● SQL POWER ARCHITEC</li> <li>● Git</li> <li>● Postman</li> </ul> |
| Entregables                   | Software el cual cumple con las necesidades y requerimientos especificados por la empresa                                                                                                                |
| Tiempo                        | 9 semanas                                                                                                                                                                                                |

Fuente: Autor

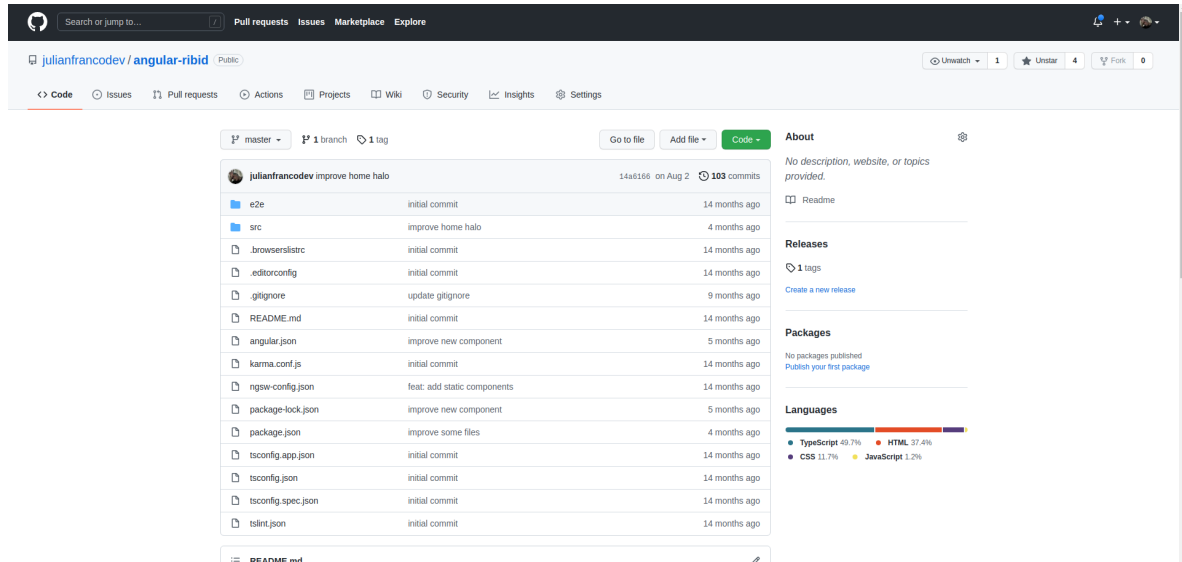
Esta fase tiene como resultado el código fuente dispuesto en el repositorio remoto tanto el servidor como el cliente.

Ilustración 5 Lugar del repositorio del código fuente del Backend



Fuente: Autor

Ilustración 6 Lugar del repositorio del código fuente del Frontend



Fuente: Autor

Adicionalmente se entrega el código fuente de cada uno de los servicios a la división respectiva del CRAI para su libre uso y modificación.

## 6. CRONOGRAMA DE ACTIVIDADES

Cronograma de actividades de acuerdo a los parámetros definidos en la metodología.

[illegible]

Fuente: Autor

Fuente: Autor

## 7. PRESUPUESTO

### ○ Resumen del presupuesto

Presupuestos de los recursos necesarios para desarrollar el proyecto.

Tabla 7 Resumen del presupuesto

| Rubros                       | Fuentes de Financiación |               |                    | Total        |
|------------------------------|-------------------------|---------------|--------------------|--------------|
|                              | Propias                 | Contrapartida | Otras <sup>1</sup> |              |
| Recurso Humano               | x                       |               |                    | \$50.840.000 |
| Recurso Material y/o Equipos | x                       |               |                    | \$10.375.000 |
| Total                        |                         |               |                    | \$61.215.000 |

Fuente: Autor

En las siguientes tablas se detallan los recursos necesarios para el proyecto en términos del capital humano y material.

### ○ Recurso Humano

A continuación, se indican los datos de los estudiantes y director que trabajaran en el proyecto.

Tabla 8 Recurso humano

| Nombre                        | Rol        | Dedicación Horas | Valor Hora | Fuentes de Financiación |               |       | Total        |
|-------------------------------|------------|------------------|------------|-------------------------|---------------|-------|--------------|
|                               |            |                  |            | Propias                 | Contrapartida | Otras |              |
| Julian Daniel Franco Alvarado | Estudiante | 576              | \$20.000   | x                       |               |       | \$20.520.000 |
| Total                         |            |                  |            |                         |               |       | \$20.520.000 |

Fuente: Autor

○ **Recurso Material y/o Equipos**

Tabla 9 Recurso material y/o equipos

| Recurso             | Justificación                                                            | Fuentes Financiación |               |       | Total        |
|---------------------|--------------------------------------------------------------------------|----------------------|---------------|-------|--------------|
|                     |                                                                          | Propios              | Contrapartida | Otras |              |
| 3 computadores      | Desarrollo de la plataforma.                                             | X                    |               |       | \$10.400.000 |
| Conexión a internet | Para investigar, y desarrollar la plataforma.                            | X                    |               |       | \$1.640.000  |
| Hosting             | Servidor donde se aloja la plataforma para publicarla en internet.       | X                    |               |       | \$1.300.000  |
| Dominio             | Dirección de internet donde los usuarios podrán acceder a la plataforma. | X                    |               |       | \$35.000     |
| Total               |                                                                          |                      |               |       | \$13.375.000 |

Fuente: Autor

## 8. Conclusiones

- Llevando a la practica la metodología para el diseño de diagramas MVC, se realizan de manera eficiente los diagramas UML propuestos para estructura del sistema. Es así que con la información del estándar UML se logró la exportación de cada diagrama específico cumpliendo con los objetivos establecidos.
- Es así que con la toda la información obtenida de las herramientas y en concordancia con la metodología, se consiguió terminar el software de acuerdo a las necesidades del CRAI.
- Gracias a los buenos tiempo de desarrollo permitidos se logró implementar el software al servicio de la comunidad educativa de la USTA seccional Tunja.

## 9. REFERENCIAS

- [1] M. A. Vilchis Rodríguez, «“La Web 2.0 y la nube”», Universidad Autónoma del Estado de Hidalgo, 28/11/18..
- [2] J. Strickland, ««How Web 2.0 Works»,» 2007. [En línea]. Available: [computer.howstuffworks.com..](http://computer.howstuffworks.com..)
- [3] P. Cadena, «Introducción al uso de la Web 2.0 en el estado colombiano.», 2010.
- [4] M. vazquez, «¿Web 2.0? ¿web social? ¿qué es eso?», e-Lis e-prints in library & information science, 2014.
- [5] «. E. o. M. (. V. C. D. P. f. Abstraction».. [En línea]. Available: «Simple Example of MVC (Model View Controller) Design Pattern for Abstraction». .
- [6] T. Reenskaug, "A note on DynaBook requirements", 1979.
- [7] R. S. Pressman, Ingeniería del software: Un enfoque práctico,.
- [8] «testingeducation.org,» [En línea]. Available: [www.testingeducation.org](http://www.testingeducation.org).
- [9] « A bug tracker for PostgreSQL?,» [En línea]. Available: <https://lwn.net/Articles/660468/>.
- [10] <http://laravel.com>, ««Introduccion - Documentation Laravel PHP Framework»,» [En línea]. Available: <http://laravel.com> .
- [11] <https://www.php-fig.org/psr/>, ««PHP Standards Recommendations»,» [En ] línea]. Available: <https://www.php-fig.org/psr/>.
- [12] S. o. (Angular), «<https://angular.io/>,» [En línea]. Available: <https://angular.io/>.  
]
- [13] «Arquitectura de Angular - Ventajas y Desventajas,» [En línea]. Available: ] <https://aotaduy.github.io/area204/educaci%C3%B3n/2020/04/02/angular-intro.html>.
- [14] Docker.com, «[docker.readthedocs.org..](https://docker.readthedocs.org..),» «Docker Documentation: Kernel Requirements», [En línea]. Available: [docker.readthedocs.org..](https://docker.readthedocs.org..)

- [15 S. w. o. (Docker), «<https://www.docker.com/>,» [En línea]. Available:  
] <https://www.docker.com/>.
- [16 Heroku, «<https://dashboard.heroku.com/>,» [En línea]. Available:  
] <https://dashboard.heroku.com/>.
- [17 Heroku's Ruby Cloud Platform at Building 43,  
] «<https://web.archive.org/web/20140602201113/http://www.building43.com/videos/2010/04/20/herokus-ruby-cloud-platform/>,» 2010. [En línea]. Available:  
<https://web.archive.org/web/20140602201113/http://www.building43.com/videos/2010/04/20/herokus-ruby-cloud-platform/>.
- [18 GigaOm, «<https://gigaom.com/2010/11/09/can-heroku-become-the-official-cloud-of-facebook-apps/>,» 2020. [En línea]. Available:  
<https://gigaom.com/2010/11/09/can-heroku-become-the-official-cloud-of-facebook-apps/>.
- [19 Julian, «doc tesis,» 2022.  
]

## 10. ANEXOS

### ■ Anexo 1. Diagrama Entidad-Relación Base de Datos

Enlace: [https://drive.google.com/file/d/1H66hP7q2TzsReTKEyHMpv5mD\\_kE6xGu\\_/view?usp=sharing](https://drive.google.com/file/d/1H66hP7q2TzsReTKEyHMpv5mD_kE6xGu_/view?usp=sharing)

### ■ Anexo 2. Diseño completo Interfaz

Enlace: [https://docs.google.com/document/d/1sTHGDPby7R7PYkYNTAEyi\\_45sMdCF0F5erHVDUOq5Ng/edit?usp=sharing](https://docs.google.com/document/d/1sTHGDPby7R7PYkYNTAEyi_45sMdCF0F5erHVDUOq5Ng/edit?usp=sharing)

### ■ Anexo 3. Manual de Usuario

Enlace: [https://docs.google.com/document/d/181MEeHsE\\_NUFyl0oTfcwj1wzdY83T4a1CdkCkmTDPq0/edit?usp=sharing](https://docs.google.com/document/d/181MEeHsE_NUFyl0oTfcwj1wzdY83T4a1CdkCkmTDPq0/edit?usp=sharing)

### ■ Anexo 4. Descripción de Software por parte del crai

Enlace: <https://drive.google.com/file/d/1oAPpKLwYQ5jwHOMF4mILWsUjT9f5SkKR/view?usp=sharing>

### ■ Anexo 5. Carpeta que contiene los diagramas

<https://drive.google.com/drive/folders/1XQV965lGocQc01xmImCMY2ssAPeHpiOj?usp=sharing>

### ■ Anexo 6. Carpeta que contiene los requerimientos generales de ribid

[https://drive.google.com/drive/folders/1mM4yqf2r\\_iL5b2edAHV47Nm6EenFfw7n?usp=sharing](https://drive.google.com/drive/folders/1mM4yqf2r_iL5b2edAHV47Nm6EenFfw7n?usp=sharing)