# **Documentación de Proyecto**

Logotipo

Descripción generada automáticamente

**Discovery Dish ©**

Integrantes:

Sergio Daniel Dueñas Godínez

Manuel Alejandro Murguía Orantes

Andre Yahir González Cuevas

Contenido

[**Documentación de Proyecto** 1](#_gjdgxs)

[Descripción Extensa del Proyecto 3](#_30j0zll)

[Stack de Tecnologías a Utilizar 4](#_1fob9te)

[Alcance 4](#_3znysh7)

[Diagramas 5](#_2et92p0)

[Entidades 7](#_tyjcwt)

[Planeación 8](#_3dy6vkm)

[Diagrama de Árbol 8](#_2s8eyo1)

[Captura de Pantalla de Bloques de Código 8](#_1t3h5sf)

[Lista de EndPoints 9](#_4d34og8)

## Descripción Extensa del Proyecto

La concepción de nuestro proyecto implica el desarrollo de un sitio web con contenido enfocado en recetas culinarias. Este sitio se construirá utilizando tecnologías como HTML, CSS y JavaScript. Nuestra intención es que el acceso a la página sea completamente abierto, sin requerir credenciales de usuario ni contraseña para ingresar. Los usuarios podrán explorar las diversas opciones de platos disponibles a través de la página, cada uno de los cuales estará acompañado por una imagen relacionada, el nombre del plato y posiblemente una breve descripción.

En la página, los usuarios tendrán la libertad de buscar platos por nombre o ingredientes. Cuando un plato sea seleccionado, los usuarios serán redirigidos a una página dedicada específicamente a ese plato. En esta página, se proporcionarán detalles sobre los ingredientes necesarios, las cantidades requeridas y se incluirán imágenes de referencia para asistir en la preparación del plato.

Contemplamos la posibilidad de incluir enlaces a videos de plataformas externas como YouTube, con el propósito de ofrecer una visualización más detallada del proceso de elaboración de los platos.

Otra idea que estamos considerando, aunque no está completamente definida, es la implementación de un sistema de recomendación que sugiera platos similares cuando los usuarios consulten una receta específica. Por ejemplo, si alguien busca "Fetuccini", el sistema podría mostrar recetas relacionadas como Espagueti, Lasaña, Pene, entre otros. Es importante destacar que esta idea se encuentra en una fase de desarrollo incipiente y solo se concretaría si disponemos de tiempo suficiente para su implementación.

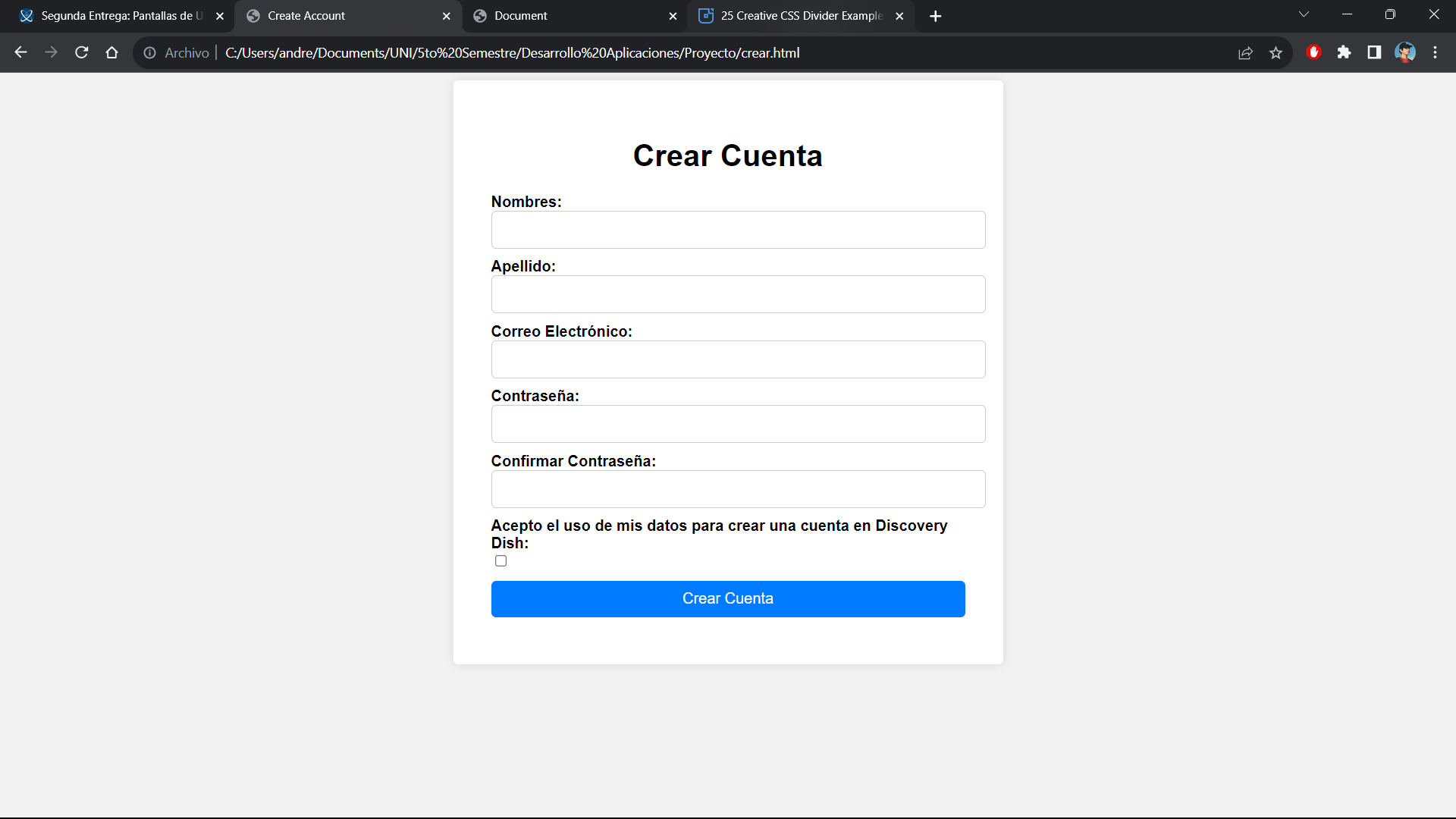
Además de las características mencionadas, planeamos incorporar un sistema de inicio de sesión y registro. Tras iniciar sesión con éxito, los usuarios podrán enviar comentarios, solicitar nuevas recetas y contribuir a la diversidad de contenidos en la página. Los comentarios podrán abordar diversos aspectos, como ofrecer consejos, brindar asistencia, así como notificar posibles errores en las páginas.

A continuación, presentamos un esquema preliminar que proporciona una visión general de la estructura de la página, aunque es importante señalar que aún queda mucho por desarrollar y agregar a fin de completar el proyecto.



Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente



## Stack de Tecnologías a Utilizar

* Front-End
* Back-End
* Bases de Datos
* Sistemas de recomendación con integración de 3ros (.py) (Si nos da tiempo lo añadimos) (si no no)

## Alcance

* Front-End
  + Login/Registro
  + Autenticación de Usuarios
  + Acceso a la REST API
  + Interfaz Responsiva
  + Validación en formularios
  + Control de Errores
  + Interfaz de Consultas
  + Uso de Imágenes
* Back-End
  + Conexión a Bases de Datos
  + REST API con operaciones CRUD
  + Validaciones antes de almacenar datos
  + Control de Errores

## Diagramas

Diagrama de HMTL (Front-End)

Diagrama

Descripción generada automáticamente

Diagrama entre Stacks

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

## Entidades

Usuario (Libre Acceso)

Este usuario tendrá los permisos para ver la página web, podrá navegar a través de las diferentes recetas sin ningún problema. No tendrá más accesos más que la libertad de navegar sin barreras.

Usuario (Registrado)

A diferencia de los usuarios de libre acceso, los usuarios registrados gozarán los privilegios de dejar comentarios & votaciones en las recetas. Podrán mandar correos a los administradores para señalar posibles errores que presenten las páginas, o bien dejar sugerencias acerca de nuevas recetas. También podrán editar sus perfiles, cambiar nombre de usuario, foto de perfil, y uno que otro detalle que se nos llegué a ocurrir. El correo no podrá ser cambiado una vez creada la cuenta.

Administrador (Nosotros)

Los administradores, como su nombre lo dicen, podrán administrar con total libertad la página y como se muestra toda la información. Podrán añadir y quitar recetas y podrán regular el comportamiento de los usuarios.

## Planeación

| Actividades | Tiempo de Duración | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Noviembre | | | | Diciembre | | | | |
| Semana 8 | Semana 9 | Semana 10 | Semana 11 | Semana 12 | Semana 13 | Semana 14 | Semana 15 | Semana 16 |
| Diseño de la Arquitectura del Sistema |  |  |  |  |  |  |  |  |  |
| Unión c/ JS y Base de Datos |  |  |  |  |  |  |  |  |  |
| Investigación y Selección de Mecanismos de Autenticación |  |  |  |  |  |  |  |  |  |
| Añadir Registro de Usuarios & Autenticación |  |  |  |  |  |  |  |  |  |
| Diseño y testeo de CRUD’s |  |  |  |  |  |  |  |  |  |
| Retoques finales + atrasos + opcionales |  |  |  |  |  |  |  |  |  |
| Pruebas Finales |  |  |  |  |  |  |  |  |  |
| Presentación Final |  |  |  |  |  |  |  |  |  |

## Estructura de Archivos

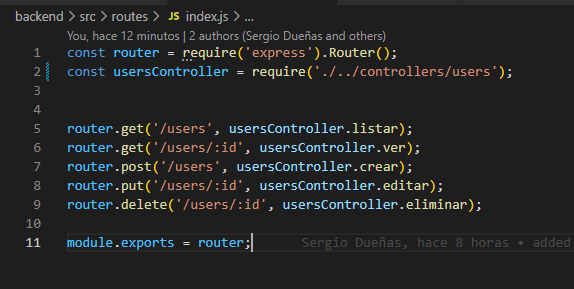
* DISCOVERY-DISH
  + backend
    - node\_modules
    - src
      * controllers
        + users.js
      * Models
        + user.js
      * routes
        + index.js
    - index.js
    - package-lock.json
    - package.json
  + frontend (Estos archivos deben ser modificados)
    - crear.html
    - estilos.css
    - index.css
    - index.html
    - login.html
    - main.js
    - nosotros.html

## Captura de Pantalla de Bloques de Código

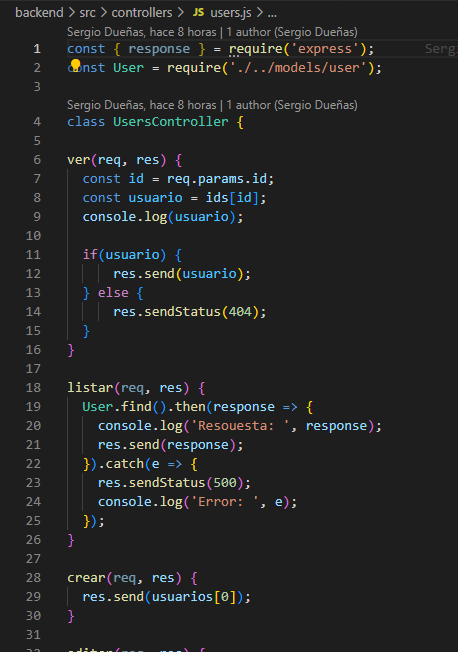
Inicio de Aplicación



Definición de Rutas



Métodos de un Controlador



Captura de pantalla de un celular

Descripción generada automáticamente

Modelos

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

## Lista de EndPoints

Verbo

Los endpoints para esta entrega están habilitados, pero no tienen función alguna, son simples vacíos para simular las acciones posteriores

1. GET
   1. Listar
   2. Ver
2. POST
   1. Crear
3. PUT
   1. Editar
4. DELETE
   1. Eliminar

URL

1. router.get('/users', usersController.listar);
2. router.get('/users/:id', usersController.ver);
3. router.post('/users', usersController.crear);
4. router.put('/users/:id', usersController.editar);
5. router.delete('/users/:id', usersController.eliminar);

PARAMETROS DE ENTRADA

1. GET – LISTAR
   1. NO SE NECESITAN PARÁMETROS
2. GET - VER
   1. ID DEL USUARIO
3. POST – CREAR
   1. ID DEL USUARIO
4. PUT - EDITAR
   1. ID DEL USUARIO
5. DELETE - DELETE
   1. ID DEL USUARIO

RESPUESTA

1. GET – LISTAR
   1. LISTA DE USUARIOS (IDS)
2. GET - VER
   1. DATOS DEL USUARIO
   2. USUARIO NO ENCONTRADO
3. POST – CREAR
   1. USUARIO CREADO
4. PUT - EDITAR
   1. USUARIO EDITADO
5. DELETE - DELETE
   1. USUARIO ELIMINADO

## Modelo de Recomendación

Para mejorar la página web hicimos un modelo de recomendación en base a los usuarios, el fin de este modelo era poder hacer la página más personalizada para el usuario. Para lograr esto usamos la librería de Surprise, donde elegimos hacer un modelo basado en SVD (singular value decomposition), el cúal está basado en content base o sea en base solo a las interacciones individuales del usuario.

El resultado fue el siguiente:



Después de crear el modelo, decidimos hacer varios para poder definir cuál era el mejor, para esto se usó Mlflow donde se puede llevar un versionamiento de los modelos con sus experimentos, el resultado fue este:

Elegimos usar SVD, implementando parámetros específicos con ayuda de Grid Search CV, librería la cúal se le entrega ciertos valores como parámetros y a través de ciclos y validación cruzada, te devuelve un array con los mejores parámetros para el modelo, el resultado en la API usando streamlit para visualizar fue el siguiente:



## Conclusiones

Tuvimos problemas con el proyecto en sí, debido a nuestra falta de conocimiento de html por eso decidimos implementar un modelo de machine learning para poder aplicar lo visto en nuestra carrera con la clase. Nos costó desplegar muchas páginas y contenido pero pudimos mostrar e implementar la base de nuestro proyecto, considerando la idea inicial y el alcance que podíamos lograr.