

Tarea T02.03

Construcción de
aplicación de software

the Past 10 Days

On the scales, am I becoming more confident or less confident? Why?

I force myself to go outside my Comfort Zone, I became more confident

Do I give myself more time? 

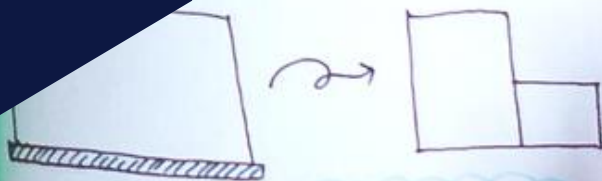
Less time worrying about failure
What time to think of new ideas!

What things I learned over the past 10 days?

AFRAID to ask for help
Solutions are ALWAYS best
15 min each day to CLEAN UP

GET IT?

portable whiteboard



Folds up for on
the go meetings!

1. **Fecha:** 16-07-2023
2. **Grupo base:** Grupo XX
3. **Integrantes:**

Apellidos y Nombres	Calificación
JENNIFFER LISSETTE GRANDA RODRIGUEZ	

4. **Objetivos específicos** que persigue la realización del trabajo.
 - 4.1. Crear aplicaciones utilizando herramientas de Ingeniería de software.
 - 4.2. Comprender el manejo de un equipo de desarrollo de software.
5. **Tareas específicas** que se deben realizar en este ejercicio.
 - 5.1. Seguir las instrucciones del ejercicio.
6. **Entregables** que deberán ser cargados en el AVAC.
 - 6.1. Archivo en PDF con el siguiente nombre: *SE_Tarea_T02_03_GrupoXX.pdf*
 - 6.2. Aplicación de Software almacenado en repositorio de código.
 - 6.3. Incluso si la Tarea se realiza en grupo, la carga a la plataforma debe ser individual.
7. **Instrucciones**
 - 7.1. Crear un repositorio de código y definir tareas de seguimiento que serán asignadas a cada integrante del grupo.
 - 7.2. Desarrollar un software que cumpla con los requerimientos de la Tarea 02.01 y las descripciones de diseño de la Tarea 02.02.
 - 7.3. El desarrollo consistirá en la implementación del backend de la aplicación, pudiendo utilizar los siguientes frameworks:
 - 7.3.1.Springboot
 - 7.3.2.FastAPI
 - 7.3.3.Flask
 - 7.3.4.Django DRF
 - 7.4. Se deberá publicar servicios con documentación mediante swagger.
 - 7.5. Para el desarrollo utilizar un esquema modelo/repositorio/controlador/servicio similar al definido en la Figura 1.

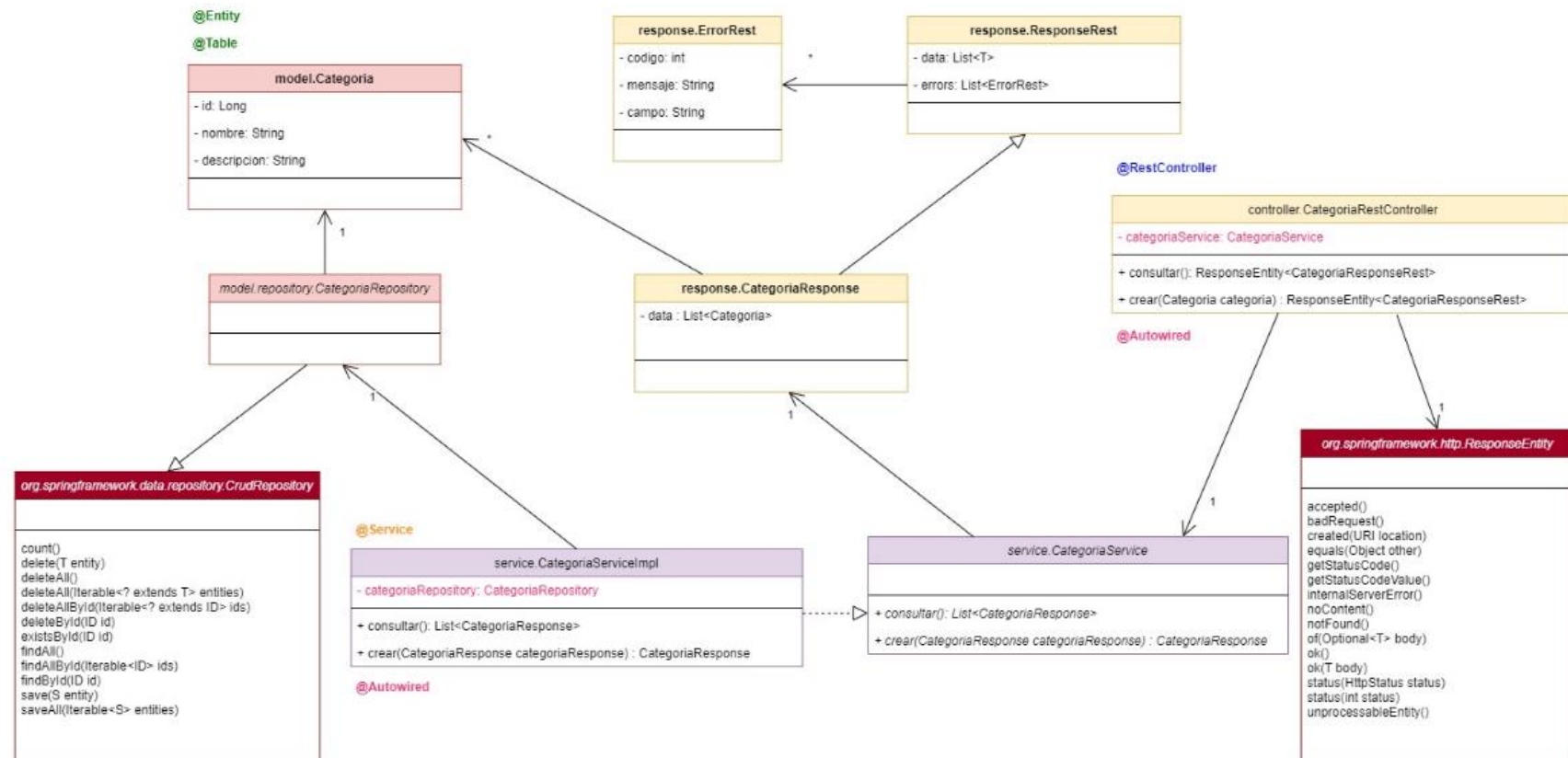


Figura 1: Estructura ejemplo para un backend

8. Código Fuente mediante un enlace del repositorio.

EJEMPLO: <https://github.com/mortizo/poo-servicio02-springboot.git>

DEBER:

<https://github.com/jenngran/citas>

9. Conclusiones de la Tarea en 200 palabras.

En este ejercicio, trabajamos en la creación de un backend utilizando Flask y PostgreSQL para administrar citas médicas y usuarios. Definimos tres clases principales: "Cita", "Historial Médico" y "Usuario", cada una con sus propios métodos y funcionalidades. En la clase "Cita", definimos una estructura que representa las citas médicas y su relación con el paciente, el médico y el motivo de la consulta. Además, implementamos la capacidad de guardar las citas en una base de datos PostgreSQL utilizando psycopg2.

En la clase "Historial Médico", desarrollamos un mecanismo para almacenar y recuperar las citas relacionadas con un paciente específico, lo que permite filtrar las citas por médico o fecha. También integramos la conexión a la base de datos mediante psycopg2 para asegurar la persistencia de los datos.

Por otro lado, en la clase "Usuario", creamos una estructura que representa a los usuarios con sus credenciales y roles. También implementamos la funcionalidad para agregar nuevos usuarios a la base de datos y autenticarlos.

En general, este ejercicio nos proporcionó una comprensión práctica de cómo utilizar Flask, PostgreSQL y la biblioteca psycopg2 para crear un backend completamente funcional y seguro. Además, utilizamos anotaciones de Swagger para documentar nuestras funciones y generar documentación de la API.

10. Rúbrica de calificación.

Criterios	No Presenta	Nivel Bajo	Nivel Medio	Nivel Alto	Total
Repositorio correctamente documentado con los requerimientos, diseño y tareas definidas por usuario.	0	0.5	1	2	
Al menos 20 commits en el repositorio que aseguren que han trabajado todos los compañeros.	0	0.2	0.5	1	
Servicios funcionales y correctamente desplegados.	0	0.2	0.5	1	