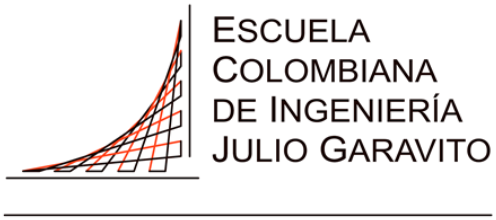
UNIVERSIDAD ESCUELA COLOMBIANA DE

INGENIERÍA JULIO GARAVITO



UNIVERSIDAD

DOCUMENTO DE ARQUITECTURA
ECI-Bienestar

Equipo Diamante

AUTOR

Vicente Garzón Ríos

MODULO

Gestión de Turnos para Servicios de Bienestar Universitario

PROGRAMA

Ingeniería de Sistemas

ASIGNATURA

Ciclos de Vida del Desarrollo de Software (CVDS)

PROFESORES

Ing. Rodrigo Humberto Gualtero Martínez

Ing. Andrés Martín Cantor Urrego

Descripción del Modulo

Este módulo permite a los miembros de la comunidad universitaria (estudiantes, docentes, administrativos y personal de servicios generales) gestionar y visualizar turnos para atención en los servicios de bienestar institucional: medicina general, odontología y psicología.

El sistema contempla la asignación de turnos desde tablets de autoservicio, control administrativo por parte del personal autorizado y seguimiento por parte de los profesionales de la salud.

Tecnologías a Usar

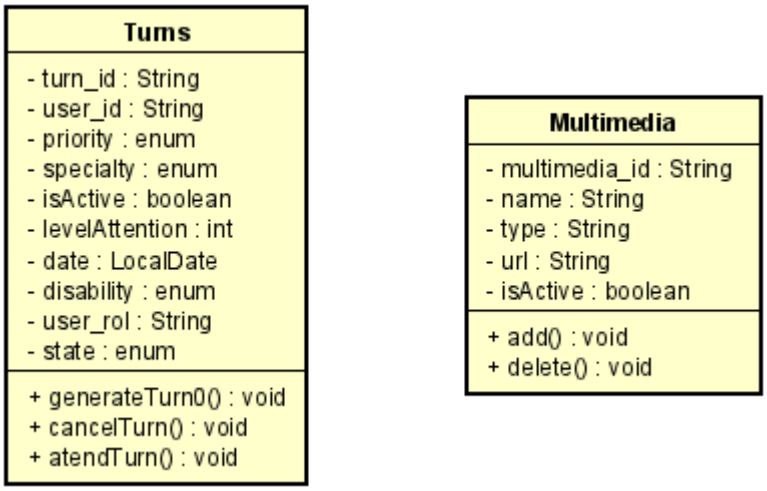
Categoría	**Tecnología**	**Justificación**
Lenguaje	Java 17	Estabilidad, soporte LTS (Long Term Support).
Framework Principal	Spring Boot	Estandarizado para microservicios RESTful.
Base de Datos	PostgreSQL	Robusto, relacional y gratuito. Ideal para consistencia transaccional.
Mensajería/Bus de Eventos	Apache Kafka + Spring Cloud Bus	Comunicación asíncrona y eventos de emergencia / actualización de estados.
Seguridad	JWT (JSON Web Token)	Autenticación segura, stateless.
ORM / Utilidades	Lombok	Para reducir código boilerplate (constructores, getters/setters automáticos).
Gestor de Proyecto	Maven	Construcción de proyectos Java estandarizada.
Actualización en Tiempo Real	WebSocket	Para actualizar la pantalla de turnos dinámicamente sin recargar.
Herramientas adicionales	JaCoCo, SonarQube	Control de calidad de código y cobertura de pruebas.

Comparativa de Tecnologías Alternativas Evaluadas:

Node.js vs Java → Se eligió Java por robustez y soporte empresarial.

MongoDB vs PostgreSQL → Se prefirió modelo relacional por estructura rígida del negocio (usuarios, turnos).

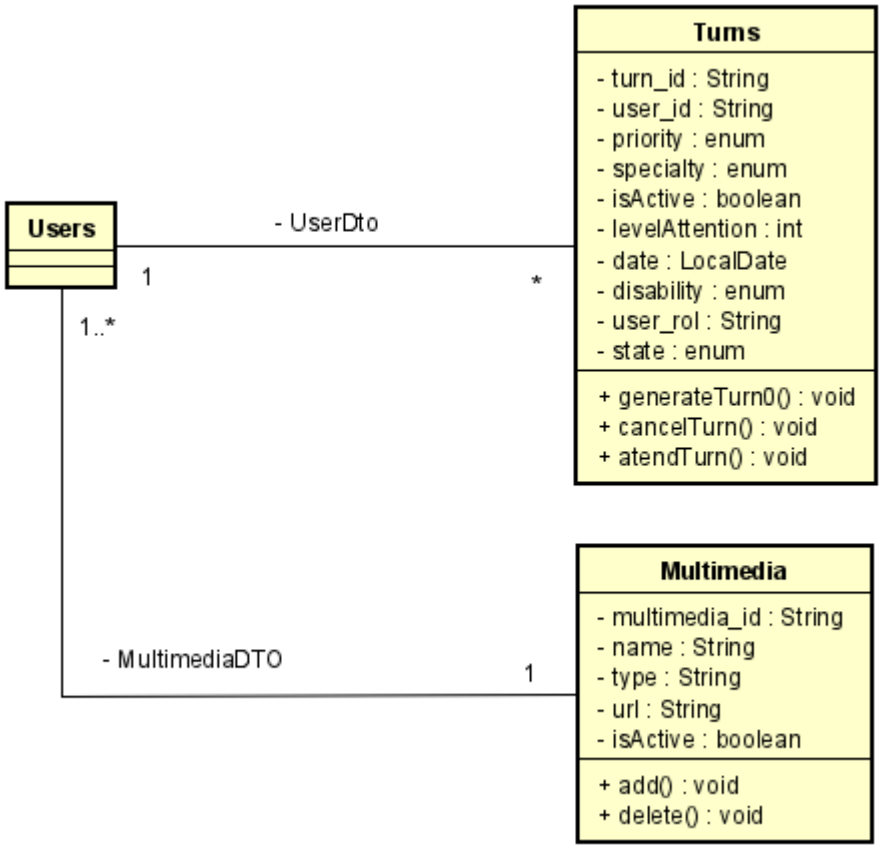
Diagrama de Datos



Inicialmente solo tenemos previsto utilizar una tabla Turns en una base de datos relacional, esto debido a que el manejo de los usuarios esta dado para otro modulo, pero en caso de ser necesario guardamos la información del usuario con un user_id (El cual llegaría a ser nuestra fk en dicho caso), la tabla guarda la información general del turno, un atributo booleano para diferenciar de si el turno está activo aun o si ya termino y un estado para ver que esta haciendo actualmente.

La tabla multimedia es para los diferentes elementos informativos que se vayan a asignar dependiendo del administrador

Diagrama de Clases



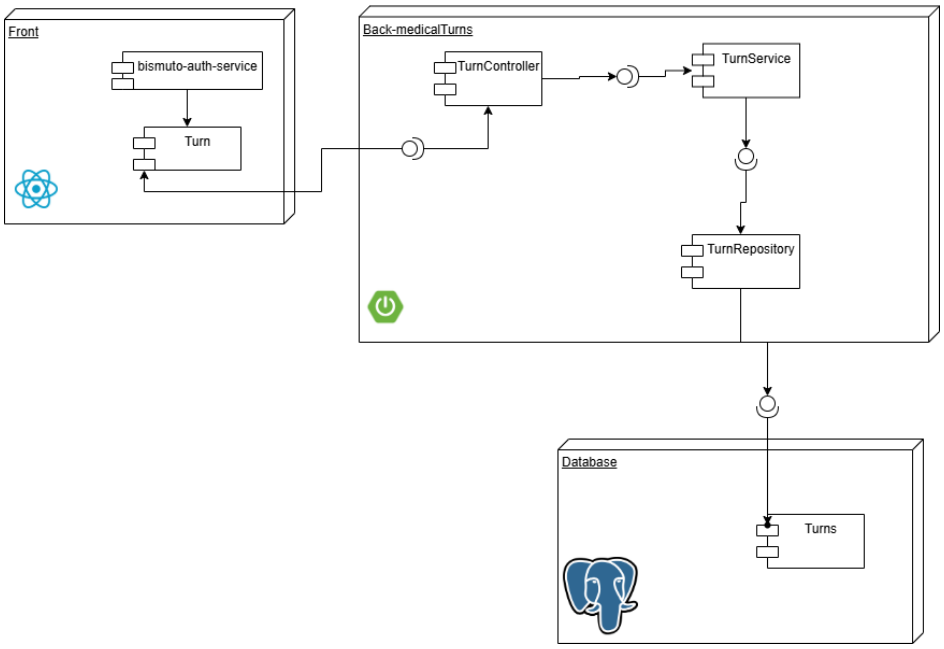
En resumen, la relación que mantenemos es que un usuario puede tener 1 o muchos turnos, en nuestro caso el UserDTO que necesitamos son los datos básicos de ingreso solicitados, como lo son el id del usuario, la prioridad y la especialidad

Diagrama de componentes

Microservicios

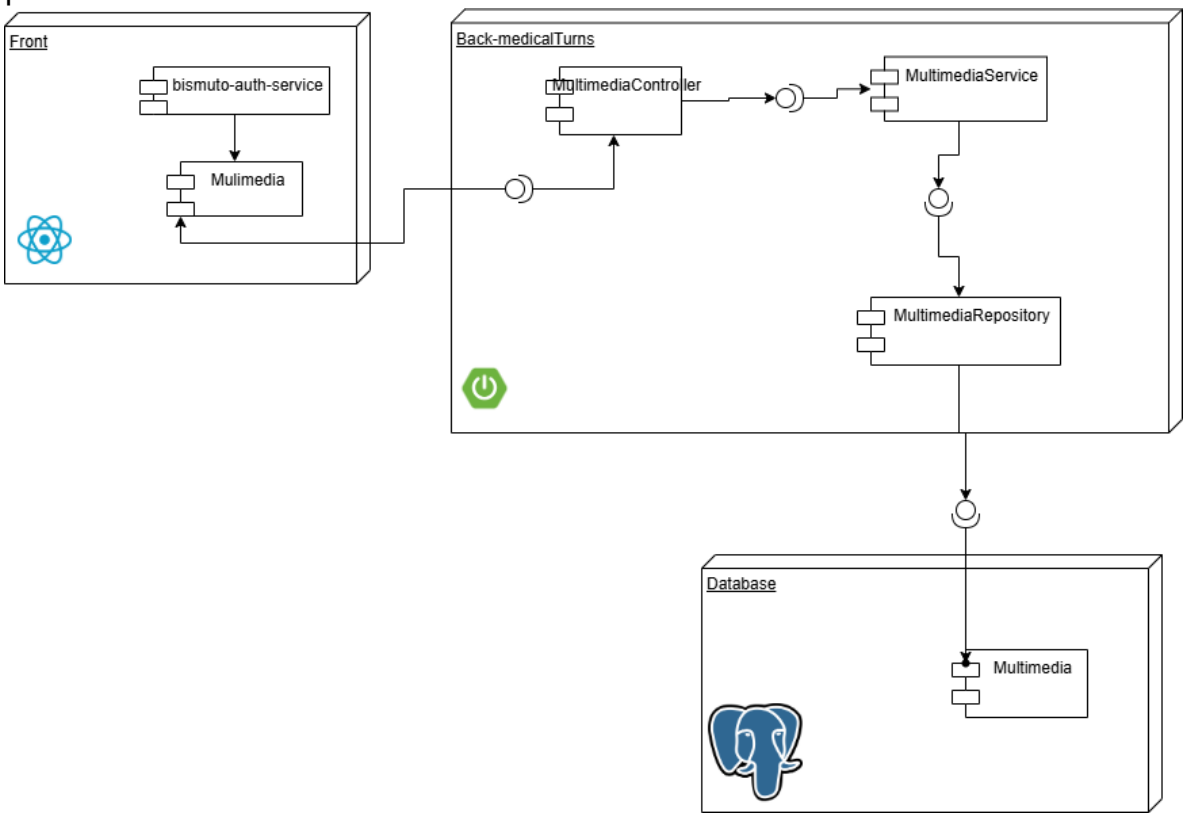
1. Turn Management Service

El siguiente microservicio se basa en el mantenimiento de los diferentes turnos, ya sea por admin o usuario común, se comunica con la clase controlador el cual dependiendo de la solicitud realiza una u otra acción, esta se conecta con el service de turn el cual mantiene la lógica separada para cada una de las solicitudes, finalmente se conecta con el repositorio de turn para guardar los datos o modificarlos de la base de datos.



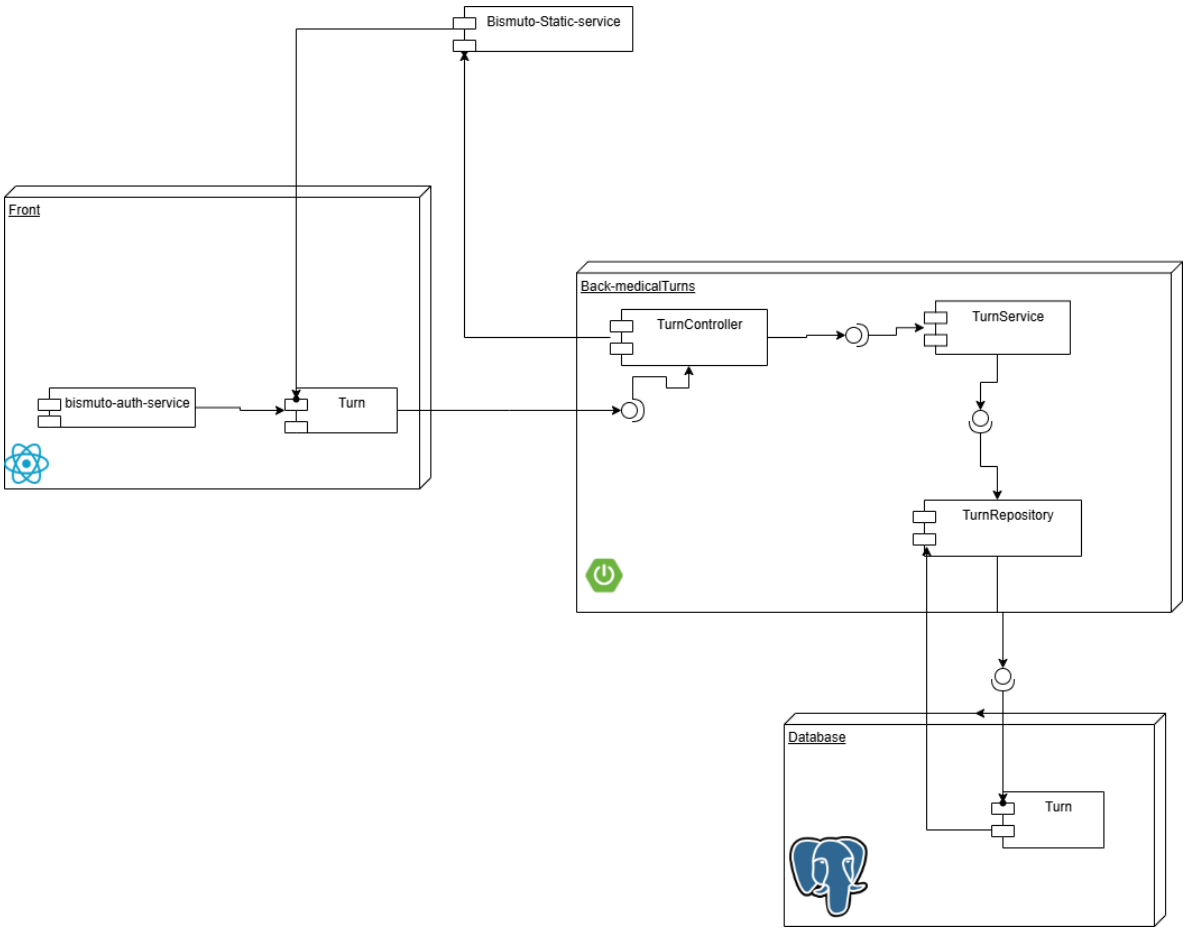
2. Multimedia Management Service

El siguiente microservicio funciona de manera similar que los turnos pero con los elementos multimedia que se usan a manera informativa, solo lo pueden usar los administradores



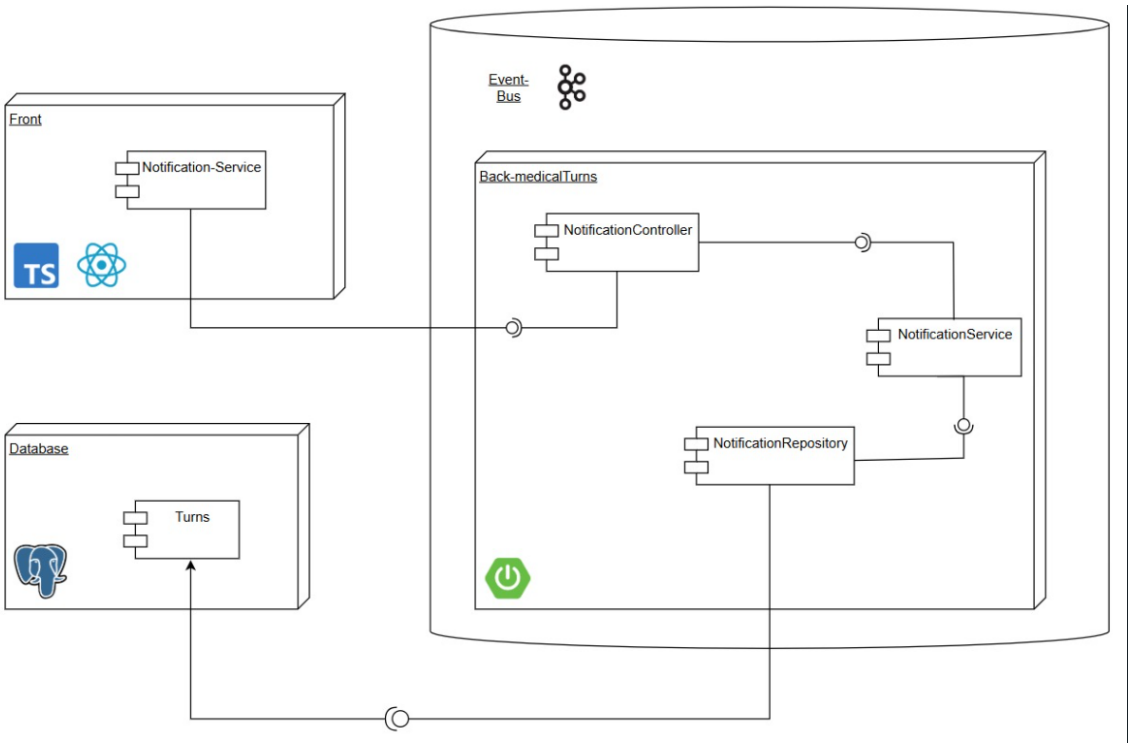
3. Report Service

El siguiente microservicio muestra de manera general el comportamiento de como se generan los reportes en nuestro modulo, se pide a través del front y nuestro back se encarga de mandar esos datos al modulo de static service del equipo bismuto, el cual retorna la información al front



4. Notification Service

El siguiente microservicio hace el seguimiento del funcionamiento del bus de eventos, para que se conecte al modulo de notificaciones en media de lo que sea necesario



Funcionalidades expuestas

Método	Endpoint	Descripción	Entrada	Salida
GET	/turns	Lista turnos	N/A	Lista de turnos
POST	/turns	Crear nuevo turno	Objeto JSON del turno	Turno creado
DELETE	/turns/{id}	Eliminar turno por ID	ID del turno	Confirmación
GET	/turns/{date}	Lista turnos en una fecha especifica	Fecha que se quiera revisar	Lista de turnos
GET	/turns/{user)	Lista de turnos para un usuario especifico	UserDto	Lista de turnos

Método	Endpoint	Descripción	Entrada	Salida
GET	/turns/{specialty}	Lista de turnos por especialidad	Especialidad que se quiera revisar	Lista de turnos
POST	/turns/enable	Habilita los turnos	N/A	Confirmación
POST	/turns/disable	Deshabilita los turnos	N/A	Confirmación
GET	/turns/lastTurn	Devuelve el ultimo turno llamado	N/A	Ultimo turno llamado
GET	/turns/pendientTurns	Devuelve todos los turnos que están pendientes	N/A	Lista de turnos que están pendientes
POST	/multimedia/{multimedia}	Sube un nuevo archivo multimedia	MultimediaDTO	Elemeto multimedia creado
GET	/multimedia	Devuelve una lista de todos los elementos multimedia que se hayan subido	N/A	Lista de elementos multimedia
GET	/multimedia/latest	Devuelve el último elemento multimedia, es decir el que se está mostrando actualmente	N/A	Elemento multimedia

Método	Endpoint	Descripción	Entrada	Salida
GET	/multimedia/{id}	Devuelve un elemento multimedia por su ID	ID del elemento multimedia	Elemento multimedia
DELETE	/multimedia/{id}	Elimina un elemento multimedia por su ID	ID del elemento	Confirmación
POST	/turns/disable/{specialty}	Deshabilita turnos de una especialidad especifica	Especialidad especifica	Confirmación
POST	/turns/enable/{specialty}	Habilita turnos de una especialidad especifica	Especialidad especifica	Confirmación

Manejo de errores

Código HTTP	Mensaje de error	Causa probable
400	"Datos de entrada inválidos"	Validaciones fallidas en el formulario
401	"Usuario no autenticado"	Token inválido o ausente
404	"Turnos no disponibles"	Los turnos están deshabilitados
404	“Especialidad no disponible”	Especialidad deshabilitada
500	"Error interno del servidor"	Fallo inesperado