Proyecto

Backend desarrollado con NestJS y MongoDB para el template GPI de la Universidad de Valparaíso. Expone una API RESTful para autenticación y gestión de usuarios, pensada para integrarse con un frontend React.

Tecnologías principales

- NestJS + TypeScript
- MongoDB con Mongoose
- JWT y Passport para autenticación
- Class Validator y bcrypt

Puesta en marcha rápida

1) Instala dependencias: `pnpm install`
2) Configura variables en `.env` (URI de MongoDB, `JWT_SECRET`, etc.).
3) Inicia MongoDB y levanta el servidor: `pnpm start:dev` (por defecto en `http://localhost:3000/api`).

Datos útiles

- Backup de la base: `backups/gpi_dump` (usa `mongorestore --drop ... ./backups/gpi_dump/gpi_database`).
- Usuario de prueba tras restaurar: `admin@admin.com / Admin1234`.
- Scripts de producción: `pnpm build` y `pnpm start:prod`.

Autenticación y uso de token

- Flujos soportados: credenciales (`/api/auth/login`), registro (`/api/auth/register`) y Google OAuth (`/api/auth/google` → callback).
- El login devuelve un `accessToken`. Guárdalo en memoria (no en localStorage si quieres evitar XSS) y envíalo en cada petición protegida con `Authorization: Bearer <token>`.
- El backend deriva el usuario desde el token; no enviamos el ID en cuerpo para rutas protegidas.
- Rutas que requieren bearer según el Swagger: usuarios y roles (create/list/get/update/delete). Para trabajar seguro, usa el token en TODAS las rutas salvo `login`, `register` y `google`/`google/callback`.
- Ejemplo con token:

```
curl -H "Authorization: Bearer $TOKEN" http://localhost:3000/api/users
```

Endpoints (qué hacen y qué esperan)

Auth

- `POST /api/auth/register`: alta de usuario (name, lastName, email, password).
- `POST /api/auth/login`: login con email/password → entrega accessToken.

- `GET /api/auth/me`: perfil del usuario autenticado (requiere bearer).
- `GET /api/auth/google`: redirige a Google OAuth.
- `GET /api/auth/google/callback`: callback OAuth; devuelve el token de sesión.

Usuarios (requiere bearer)

- `POST /api/users`: crea usuario (CreateUserDto: name, lastName, email, password, roles?, permisos?, isActive?).
- `GET /api/users`: lista todos los usuarios (solo admin).
- `GET /api/users/{id}`: obtiene usuario por ID (permiso `ver_usuario`).
- `PATCH /api/users/{id}`: actualiza usuario (permiso `editar_usuario`; usa UpdateUserDto).
- `DELETE /api/users/{id}`: elimina usuario (solo admin).

Roles (requiere bearer)

- `POST /api/roles`: crea rol (codigo, nombre, descripcion).
- `GET /api/roles`: lista roles.
- `GET /api/roles/{id}`: detalle de un rol.
- `PATCH /api/roles/{id}`: actualiza rol.
- `DELETE /api/roles/{id}`: elimina rol.

Permisos

- `POST /api/permisos`: crea permiso (CreatePermisoDto).
- `GET /api/permisos`: lista permisos.
- `GET /api/permisos/{id}`: detalle de permiso.
- `PATCH /api/permisos/{id}`: actualiza permiso.
- `DELETE /api/permisos/{id}`: elimina permiso.

Roles-Permisos

- `POST /api/roles-permisos`: asigna permiso a rol (CreateRolePermisoDto).
- `GET /api/roles-permisos`: lista asignaciones.
- `GET /api/roles-permisos/{id}`: detalle de asignación.
- `PATCH /api/roles-permisos/{id}`: actualiza asignación (UpdateRolePermisoDto).
- `DELETE /api/roles-permisos/{id}`: elimina asignación.

Verificaciones de correo

- `POST /api/verificaciones-correo`: crea/verifica solicitud de verificación (CreateVerificacionCorreoDto).
- `GET /api/verificaciones-correo`: lista verificaciones.
- `GET /api/verificaciones-correo/{id}`: detalle de verificación.

- `PATCH /api/verificaciones-correo/{id}`: actualiza verificación (UpdateVerificacionCorreoDto).
- `DELETE /api/verificaciones-correo/{id}`: elimina verificación.

Vendor Accreditations (requiere bearer)

- `POST /api/vendor-accreditations`: crea solicitud de acreditación de vendedor. Body (`CreateVendorAccreditationDto`):
  - `nombre_tienda` (string)
  - `telefono_contacto` (string)
  - `rut_empresa` (string)
    El backend asocia el `usuario_id` desde el token (`req.user.userId`), no enviar el ID en el body.
- `GET /api/vendor-accreditations`: lista todas las solicitudes (solo rol `admin`, JwtAuthGuard + RolesGuard).
- `DELETE /api/vendor-accreditations/{id}`: elimina una solicitud (solo rol `admin`).