



DESARROLLO BACKEND EVALUACION N°3

PROFESOR: VICENTE ZAPATA

ALUMNO: ARTURO CHACÓN

Documentación de la API – Evaluación Unidad 3

Este proyecto corresponde a la evaluación de la unidad 3 para el ramo de Desarrollo Backend. Se desarrolla una API RESTful enfocada en el perfil de Reclutador, permitiendo gestionar las entidades **oferta laboral, usuario, postulación, antecedentes laborales y antecedentes académicos**.

Incluye documentación interactiva con **Swagger**.



1) Tecnologías utilizadas

- Node.js
- Express.js
- MySQL
- dotenv
- Swagger UI
- Thunder Client (para pruebas)
- Visual Studio Code

2) Estructura del proyecto

```
eva3_backend/  
├── config/  
│   ├── .env  
│   ├── db.js  
│   └── database.sql  
├── controllers/  
│   ├── ofertaLaboralController.js  
│   ├── usuarioController.js  
│   ├── postulacionController.js  
│   ├── antecedentesLaboralesController.js  
│   └── antecedentesAcademicosController.js  
├── models/  
│   ├── ofertaLaboral.js  
│   ├── usuario.js  
│   ├── postulacion.js  
│   ├── antecedentesLaborales.js  
│   └── antecedentesAcademicos.js  
├── routes/  
│   ├── ofertaLaboralRoutes.js  
│   ├── usuarioRoutes.js  
│   ├── postulacion.js  
│   ├── antecedentesLaborales.js  
│   └── antecedentesAcademicos.js  
├── app.js  
├── package.json  
├── swagger.js  
└── README.md
```

3) Instalación

1. Clona el repositorio y navega a la carpeta del proyecto.

2. Instala las dependencias:

```
npm install
```

3. Configura la base de datos:

- Importa el archivo [database.sql](#) en tu gestor MySQL.
- Asegúrate de que las tablas usuario, ofertalaboral, postulacion, antecedenteslaborales y antecedentesacademicos existan.

4. Configura las variables de entorno en config/.env:

```
DB_HOST=localhost
```

```
DB_USER=root
```

```
DB_PASSWORD=
```

```
DB_NAME=cliente_feliz
```

```
DB_PORT=3306
```

```
PORT=3000
```

5. Iniciar el servidor:

```
node app.js
```

4) Endpoints creados

Ofertas Laborales

Método	Ruta	Descripción
POST	/api/ofertaLaboral	Crear una oferta laboral
GET	/api/ofertaLaboral	Listar todas las ofertas
GET	/api/ofertaLaboral/:id	Obtener una oferta por ID
PUT	/api/ofertaLaboral/:id	Editar una oferta
DELETE	/api/ofertaLaboral/:id	Eliminar una oferta

Usuarios

Método	Ruta	Descripción
POST	/api/usuarioRoutes	Crear un usuario
GET	/api/usuarioRoutes	Listar todos los usuarios
GET	/api/usuarioRoutes/:id	Obtener un usuario por ID
PUT	/api/usuarioRoutes/:id	Editar un usuario
DELETE	/api/usuarioRoutes/:id	Eliminar un usuario

Postulaciones

Método	Ruta	Descripción
POST	/api/postulacion	Crear una postulación
GET	/api/postulacion	Listar todas las postulaciones
PUT	/api/postulacion	Editar una postulación
DELETE	/api/postulacion	Eliminar una postulación

Antecedentes Laborales

Método	Ruta	Descripción
POST	/api/antecedentesLaborales	Crear un antecedente laboral
GET	/api/antecedentesLaborales	Listar todos los antecedentes
GET	/api/antecedentesLaborales/:id	Obtener un antecedente por ID
PUT	/api/antecedentesLaborales/:id	Editar un antecedente



Antecedentes Académicos

Método	Ruta	Descripción
POST	/api/antecedentesAcademicos	Crear un antecedente académico
GET	/api/antecedentesAcademicos	Listar todos los antecedentes
GET	/api/antecedentesAcademicos/:id	Obtener un antecedente por ID
PUT	/api/antecedentesAcademicos/:id	Editar un antecedente
DELETE	/api/antecedentesAcademicos/:id	Eliminar un antecedente

5) Documentación Swagger

- Accede a la documentación interactiva en:
<http://localhost:3001/api-docs>

6) Pruebas realizadas con

- **Thunder Client** para probar cada endpoint.
- Se incluye archivo .json con la colección de pruebas.

Desarrollado por:

Arturo Chacón Ríos

Instituto Profesional IPSS – 2° Año

Ramo: Desarrollo Backend