

Guía de Pruebas de API REST con Postman

1. Empresas

GET /api/empresas

- Método HTTP: GET
- URL: `http://localhost:8080/api/empresas`
- Descripción: Obtiene una lista de todas las empresas.

GET /api/empresas/{id}

- Método HTTP: GET
- URL: `http://localhost:8080/api/empresas/{id}`
- Descripción: Obtiene una empresa por su ID.

POST /api/empresas

- Método HTTP: POST
- URL: `http://localhost:8080/api/empresas`
- Descripción: Crea una nueva empresa.
- Body (JSON):

```
{  
  "nombre": "Tech Innovations",  
  "descripcion": "Innovación tecnológica en desarrollo de software.",  
  "sector": "Tecnología",  
  "ubicacion": "Ciudad de México",  
  "correoContacto": "contacto@techinnovations.com",  
  "fechaRegistro": "2025-02-27"  
}
```

PUT /api/empresas/{id}

- Método HTTP: PUT
- URL: http://localhost:8080/api/empresas/{id}
- Descripción: Actualiza una empresa existente.
- Body (JSON):

```
{  
  "nombre": "Tech Innovations Updated",  
  "descripcion": "Innovación tecnológica en IA y desarrollo de software.",  
  "sector": "Tecnología",  
  "ubicacion": "Monterrey",  
  "correoContacto": "contacto@techinnovations.com",  
  "fechaRegistro": "2025-02-27"  
}
```

DELETE /api/empresas/{id}

- Método HTTP: DELETE
- URL: http://localhost:8080/api/empresas/{id}
- Descripción: Elimina una empresa por su ID.

2. Ofertas de Empleo

GET /api/ofertas

- Método HTTP: GET
- URL: http://localhost:8080/api/ofertas
- Descripción: Obtiene una lista de todas las ofertas de empleo.

GET /api/ofertas/{id}

- Método HTTP: GET
- URL: `http://localhost:8080/api/ofertas/{id}`
- Descripción: Obtiene una oferta de empleo por su ID.

POST /api/ofertas

- Método HTTP: POST
- URL: `http://localhost:8080/api/ofertas`
- Descripción: Crea una nueva oferta de empleo.
- Body (JSON):

```
{  
  "titulo": "Desarrollador Backend",  
  "descripcion": "Se busca desarrollador backend con experiencia en Spring Boot y bases de  
datos relacionales.",  
  "salarioMin": 40000,  
  "salarioMax": 60000,  
  "requisitos": "Experiencia con Java, Spring Boot, SQL.",  
  "tipoJornada": "Tiempo completo",  
  "modalidad": "Remoto",  
  "ubicacion": "Ciudad de México",  
  "fechaPublicacion": "2025-02-27",  
  "fechaExpiracion": "2025-03-27",  
  "estado": "Activa",  
  "empresa": {  
    "id": 1  
  }  
}
```

PUT /api/ofertas/{id}

- Método HTTP: PUT

- URL: http://localhost:8080/api/ofertas/{id}

- Descripción: Actualiza una oferta de empleo existente.

- Body (JSON):

```
{  
  "titulo": "Desarrollador Backend Senior",  
  "descripcion": "Buscamos un desarrollador backend senior con más de 5 años de experiencia  
en Spring Boot y bases de datos relacionales.",  
  "salarioMin": 50000,  
  "salarioMax": 80000,  
  "requisitos": "Experiencia avanzada con Java, Spring Boot, SQL y arquitectura de  
microservicios.",  
  "tipoJornada": "Tiempo completo",  
  "modalidad": "Remoto",  
  "ubicacion": "Ciudad de México",  
  "fechaPublicacion": "2025-02-27",  
  "fechaExpiracion": "2025-06-30",  
  "estado": "Activa",  
  "empresa": {  
    "id": 1  
  }  
}
```

DELETE /api/ofertas/{id}

- Método HTTP: DELETE

- URL: http://localhost:8080/api/ofertas/{id}

- Descripción: Elimina una oferta de empleo por su ID.

3. Candidatos

GET /api/candidatos

- Método HTTP: GET
- URL: <http://localhost:8080/api/candidatos>
- Descripción: Obtiene una lista de todos los candidatos.

GET /api/candidatos/{id}

- Método HTTP: GET
- URL: <http://localhost:8080/api/candidatos/{id}>
- Descripción: Obtiene un candidato por su ID.

POST /api/candidatos

- Método HTTP: POST
- URL: <http://localhost:8080/api/candidatos>
- Descripción: Crea un nuevo candidato.
- Body (JSON):

```
{  
  "nombre": "Juan",  
  "apellido": "Pérez",  
  "correoElectronico": "juan.perez@email.com",  
  "telefono": "555-1234567",  
  "cv": "http://example.com/cv/juan_perez.pdf",  
  "habilidades": "Java, Spring Boot, SQL",  
  "experienciaLaboral": "5 años como desarrollador backend",  
  "educacion": "Ingeniería en Sistemas Computacionales",
```

"fechaRegistro": "2025-02-27"

}

PUT /api/candidatos/{id}

- Método HTTP: PUT

- URL: http://localhost:8080/api/candidatos/{id}

- Descripción: Actualiza un candidato existente.

- Body (JSON):

{

"nombre": "Juan Carlos",

"apellido": "Pérez Gómez",

"correoElectronico": "juancarlos.perez@email.com",

"telefono": "555-9876543",

"cv": "http://example.com/cv/juan_carlos_perez.pdf",

"habilidades": "Java, Spring Boot, MongoDB",

"experienciaLaboral": "7 años como desarrollador backend",

"educacion": "Ingeniería en Sistemas Computacionales",

"fechaRegistro": "2025-02-27"

}

DELETE /api/candidatos/{id}

- Método HTTP: DELETE

- URL: http://localhost:8080/api/candidatos/{id}

- Descripción: Elimina un candidato por su ID.

4. Postulaciones

GET /api/postulaciones

- Método HTTP: GET
- URL: `http://localhost:8080/api/postulaciones`
- Descripción: Obtiene una lista de todas las postulaciones.

GET `/api/postulaciones/{id}`

- Método HTTP: GET
- URL: `http://localhost:8080/api/postulaciones/{id}`
- Descripción: Obtiene una postulación por su ID.

POST `/api/postulaciones`

- Método HTTP: POST
- URL: `http://localhost:8080/api/postulaciones`
- Descripción: Crea una nueva postulación.
- Body (JSON):

```
{  
  "candidato": {  
    "id": 1  
  },  
  "oferta": {  
    "id": 1  
  },  
  "fechaPostulacion": "2025-02-27",  
  "estado": "Enviada",  
  "comentarios": "Motivado por el proyecto y la oportunidad de crecimiento."  
}
```

PUT `/api/postulaciones/{id}`

- Método HTTP: PUT
- URL: `http://localhost:8080/api/postulaciones/{id}`
- Descripción: Actualiza el estado de una postulación.
- Body (JSON):

```
{  
  "estado": "En revisión",  
  "comentarios": "Esperando la respuesta de la empresa."  
}
```

DELETE `/api/postulaciones/{id}`

- Método HTTP: DELETE
- URL: `http://localhost:8080/api/postulaciones/{id}`
- Descripción: Elimina una postulación por su ID.