

MANUAL DE USUARIO

PROYECTO

Kevin Estuardo Secaida Molina
Gabriel Enrique Perez Meza
César André Ramírez Dávila
Dayana Alejandra Reyes Rodríguez

Desktop Records

El sitio web de Desktop Records es capaz de procesar, manipular y presentar información sobre los empleados de la empresa, discos y países que manejan.

El sitio web está conformado por backend y frontend, y maneja la información con archivos xml integrados en el programa, el formato de estos archivos es:

xml de empleados:

El nombre del archivo es empleados.xml

```
<empresa>
  <departamento departamento="Contabilidad">
    <empleado id="1">
      <nombre>José Ernesto</nombre>
      <puesto>Gerente general</puesto>
      <salario>4000.00</salario>
    </empleado>
    <empleado id="2">
      <nombre>Daniel Pérez</nombre>
      <puesto>Contador</puesto>
      <salario>2000.00</salario>
    </empleado>
  </departamento>
  <departamento departamento="RRHH">
    <empleado id="3">
      <nombre>Marta Torres</nombre>
      <puesto>Reclutadora</puesto>
      <salario>3000.00</salario>
    </empleado>
  </departamento>
</empresa>
```

xml de discos:

El nombre del archivo es discos.xml

```
<catalog>
  <cd>
    <title>empire burlesque</title>
    <artist>bob dylan</artist>
    <country>usa</country>
    <company>columbia</company>
    <price>10.90</price>
    <year>1985</year>
  </cd>
  <cd>
    <title>greatest hits</title>
    <artist>dolly parton</artist>
    <country>usa</country>
    <company>rca</company>
    <price>9.90</price>
    <year>1982</year>
  </cd>
  <cd>
    <title>"empire burlesque"</title>
    <artist>bee gees</artist>
    <country>uk</country>
    <company>polydor</company>
    <price>10.90</price>
    <year>1998</year>
  </cd>
</catalog>
```

xml países:

El nombre del archivo es: mundo.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<mondo>
  <continente name="Europa">
    <pais moneda="Euro">
      <nombre>Monaco</nombre>
      <capital>Ciudad de monaco</capital>
      <idioma>Frances</idioma>
      <poblacion year="2019" unit="thousands">38.964</poblacion>
    </pais>
  </continente>
  <continente name="America">
    <pais moneda="Quetzal">
      <nombre>Guatemala</nombre>
      <capital>Ciudad de Guatemala</capital>
      <idioma>Español</idioma>
      <poblacion year="2019" unit="millions">16.6</poblacion>
    </pais>
  </continente>
  <continente name="Oceania">
    <pais moneda="Dólar australiano">
      <nombre>Australia</nombre>
      <capital>Canberra</capital>
      <idioma>Ingles</idioma>
      <poblacion year="2020" unit="millions">25.69 </poblacion>
    </pais>
  </continente>
</mondo>
```

La información manejada con empleados es:

Ver empleados:

Muestra el listado de todos los empleados de cada departamento en el sistema.

The screenshot shows a REST client interface for a project named "Proyecto IPC 2 / Verempleados". A GET request to `http://127.0.0.1:7000/empleados` was sent successfully. The response is a JSON object showing the structure of employee data, including company, department, and individual employee details.

Request: GET `http://127.0.0.1:7000/empleados`

Response Body (JSON):

```
{
  "empresa": {
    "departamento": [
      {
        "departamento": "Contabilidad",
        "empleado": [
          {
            "id": "1",
            "nombre": "José Ernesto"
          },
          {
            "id": "2",
            "nombre": "Daniel Pérez"
          }
        ]
      }
    ]
  }
}
```

Buscar según sueldo

Muestra el listado de todos los empleados de cada departamento con el sueldo indicado.

POST `http://127.0.0.1:7000/Bsueldo` **Send**

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** Beautify

```
1 {
2   "salario": "4000.00"
3 }
```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 15 ms Size: 331 B Save Response

Pretty Raw Preview Visualize **JSON**

```
1 {
2   "empleados": [
3     {
4       "departamento": "Contabilidad",
5       "id": "1",
6       "nombre": "José Ernesto",
7       "puesto": "Gerente general",
8       "salario": "4000.00"
9     }
10  ]
11 }
```

Agregar empleado

Se ingresan los datos del nuevo empleado y se agregará al archivo xml

POST `http://127.0.0.1:7000/agregarEmpleado` **Send**

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** Beautify

```
1 {
2   "departamento": "Prueba",
3   "id": "12",
4   "nombre": "ultimo agregado",
5   "puesto": "gerente",
6   "salario": "12000.00"
7 }
```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 33 ms Size: 230 B Save Response

Pretty Raw Preview Visualize **HTML**

```
1 Empleado ha sido agregado
```

```
<departamento departamento="Prueba">
  <empleado id="12">
    <nombre>ultimo agregado</nombre>
    <puesto>gerente</puesto>
    <salario>12000.00</salario>
  </empleado>
</departamento>
```

Eliminar empleado

Se ingresa el ID del empleado y se elimina del archivo xml

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:9000/eliminarEmpleado`. The request body is a JSON object: `{ "id": "12" }`. The response status is `200 OK` with a time of `28 ms` and a size of `243 B`. The response body is a JSON object: `{ "mensaje": "Empleado eliminado", "status": 200 }`.

```
POST http://127.0.0.1:9000/eliminarEmpleado
```

```
{
  "id": "12"
}
```

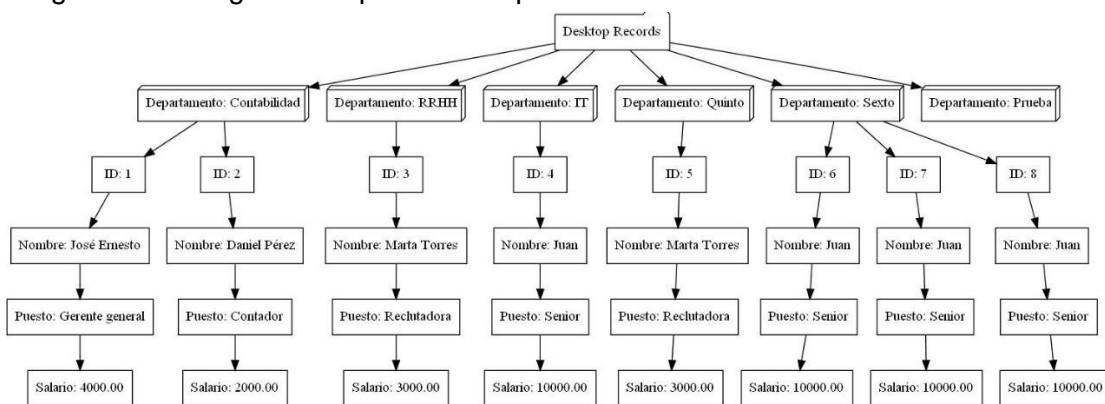
Status: 200 OK Time: 28 ms Size: 243 B

```
{
  "mensaje": "Empleado eliminado",
  "status": 200
}
```

```
<departamento departamento="Prueba">
</departamento>
```

Reporte empleados

Se genera la imagen del reporte de empleados.



La información manejada con discos es:

Ver discos

Muestra el listado de todos los discos.

Buscar discos según nombre

Muestra el listado de todos los discos que tengan el nombre indicado.

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:9000/discoTitulo`. The request body is a JSON object: `{ "title": "red" }`. The response is a JSON object with the following fields: `artist: "the communards", company: "london", country: "uk", mensaje: "Disco encontrado", price: "7.80", title: "red", year: "1987"`. The status is 200 OK, time is 63 ms, and size is 332 B.

```
POST http://127.0.0.1:9000/discoTitulo
```

```
{
  "title": "red"
}
```

```
{
  "artist": "the communards",
  "company": "london",
  "country": "uk",
  "mensaje": "Disco encontrado",
  "price": "7.80",
  "title": "red",
  "year": "1987"
}
```

Buscar discos según año

Muestra el listado de todos los discos del año indicado.

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:9000/yearDisco`. The request body is a JSON object: `{ "year": "1968" }`. The response is a JSON array containing one object with the following fields: `anio: "1968", artista: "otis redding", ciudad: "usa", compania: "stax records", precio: "7.90", titulo: "the dock of the bay"`. The status is 200 OK, time is 9 ms, and size is 340 B.

```
POST http://127.0.0.1:9000/yearDisco
```

```
{
  "year": "1968"
}
```

```
[
  {
    "anio": "1968",
    "artista": "otis redding",
    "ciudad": "usa",
    "compania": "stax records",
    "precio": "7.90",
    "titulo": "the dock of the bay"
  }
]
```

Buscar discos según artista

Muestra el listado de todos los discos del artista indicado.

POST `http://127.0.0.1:9000/discoArtista` Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Beautify

```
1 [{"artist": "bonnie tyler"}]
```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 15 ms Size: 451 B Save Response

Pretty Raw Preview Visualize **JSON** Copy Search

```
1 [{"artistas": [
2   {
3     "artist": "bonnie tyler",
4     "company": "cbs records",
5     "country": "uk",
6     "price": "9.90",
7     "title": "hide your heart",
8     "year": "1988"
9   },
10  ],
11  ]}]
```

Modificar disco

Se ingresan los datos del disco y se editará en el archivo xml

POST `http://127.0.0.1:9000/modificarDisco` Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Beautify

```
1 [{"title": "picture book",
2   "artist": "Alguien diferente",
3   "country": "Guatemala",
4   "company": "elektra",
5   "price": "7.20",
6   "year": "2022"}]
```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 14 ms Size: 241 B Save Response

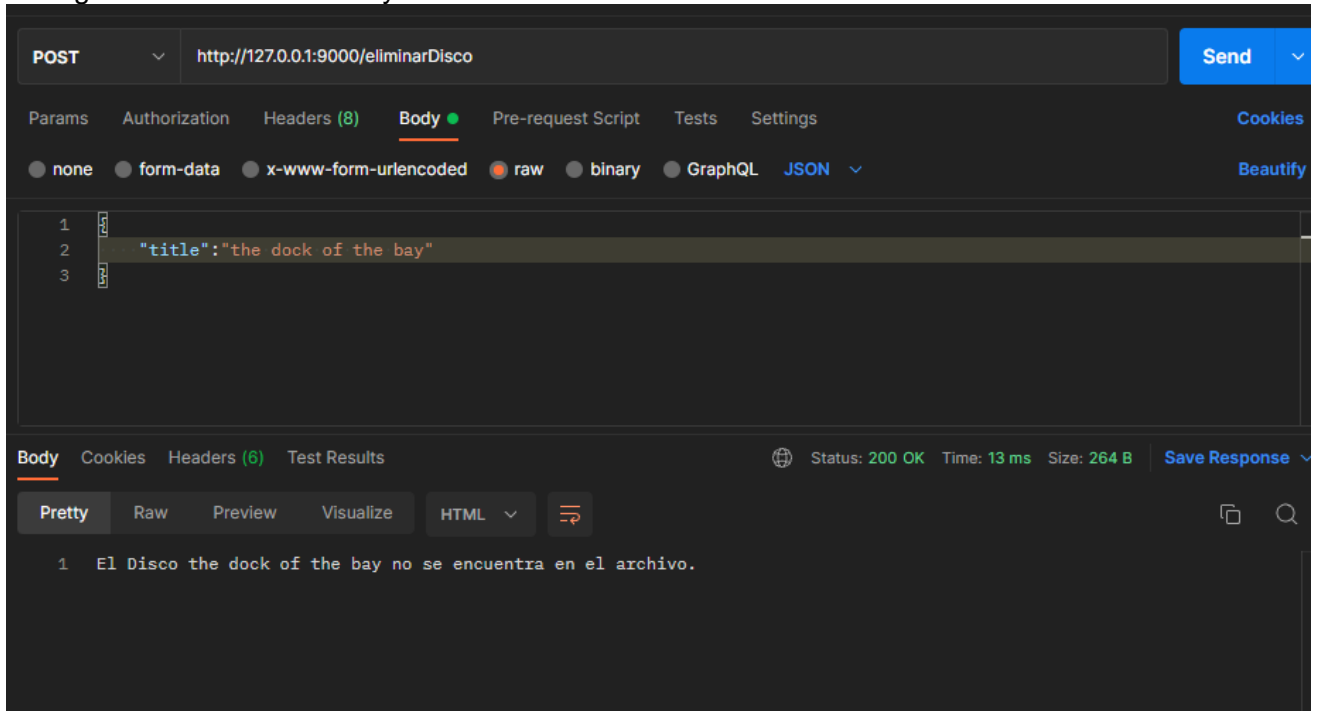
Pretty Raw Preview Visualize **JSON** Copy Search

```
1 {"mensaje": "Disco modificado",
2   "status": 200}
```

```
<cd>
  <title>picture book</title>
  <artist>Alguien diferente</artist>
  <country>Guatemala</country>
  <company>elektra</company>
  <price>7.20</price>
  <year>2022</year>
</cd>
```

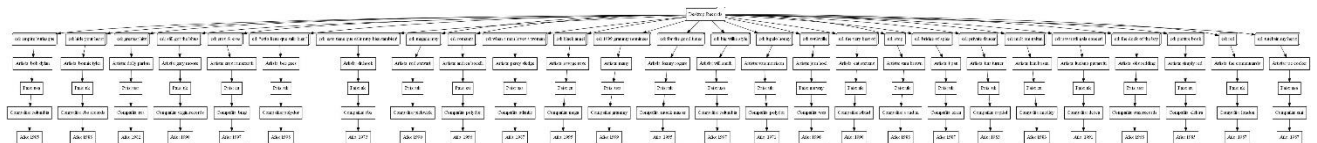
Eliminar disco

Se ingresa el título del disco y se eliminará del archivo xml



Reporte discos

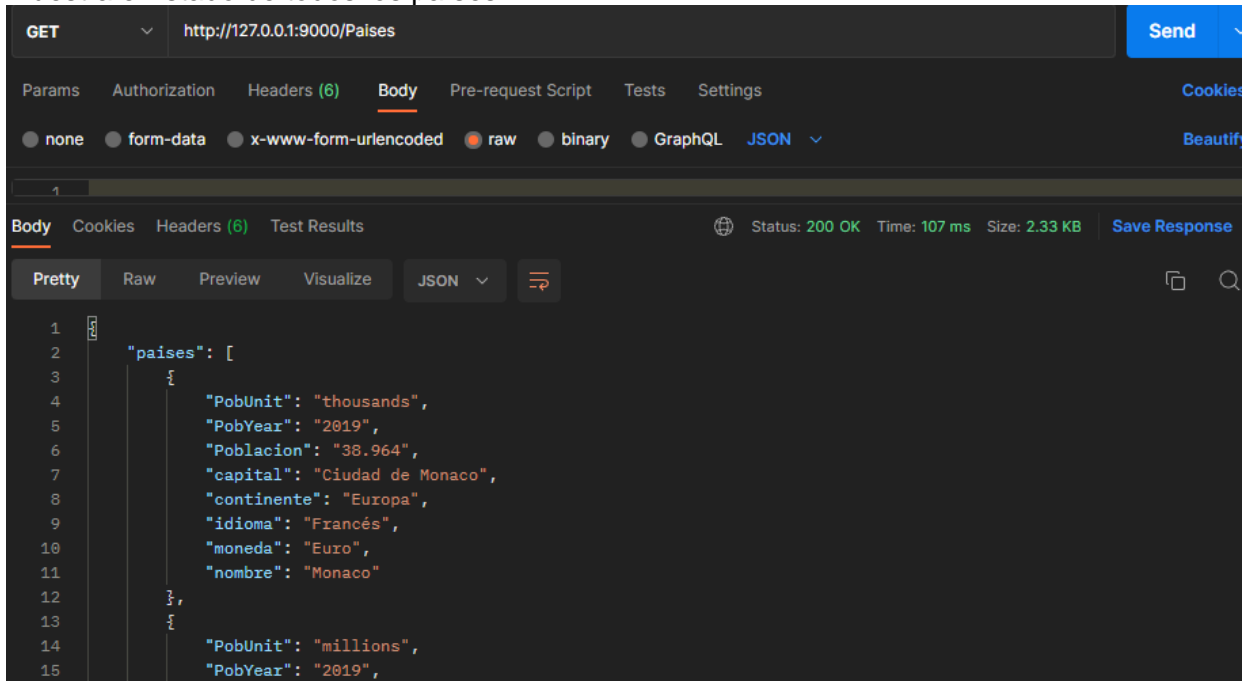
Se generará la imagen del reporte de discos



La información manejada con países es:

Ver regiones

Muestra el listado de todos los países



Buscar países según moneda

Muestra el listado de todos los países con la moneda indicada.

POST `http://127.0.0.1:9000/paisMoneda` Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   ... "moneda": "Dolar"
3 }
```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 8 ms Size: 483 B Save Response

Pretty Raw Preview Visualize JSON ≡

```
2 {
3   "monedas": [
4     {
5       "capital": "Washington DC",
6       "continente": "America",
7       "idioma": "Inglés",
8       "moneda": "Dolar",
9       "nombre": "Estados unidos",
10      "poblacion": "328.2"
11    },
12    {
13      "capital": "San Salvador",
14      "continente": "America",
```

Cookies Capture requests Bootcamp Runner Trash

Buscar países según continente

Muestra el listado de todos los países del continente indicado.

POST `http://127.0.0.1:9000/continente` Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   ... "continente": "Europa"
3 }
```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 8 ms Size: 826 B Save Response

Pretty Raw Preview Visualize JSON ≡

```
1 {
2   "paises": [
3     {
4       "capital": "Ciudad de Monaco",
5       "idioma": "Francés",
6       "moneda": "Euro",
7       "pais": "Monaco",
8       "poblacion": "38.964"
9     },
10    {
11      "capital": "Viena",
12      "idioma": "Alemán",
13      "moneda": "Euro"
```

Cookies Capture requests Bootcamp Runner Trash

Reporte regiones

Se generará la imagen del reporte de los países de cada continente.

