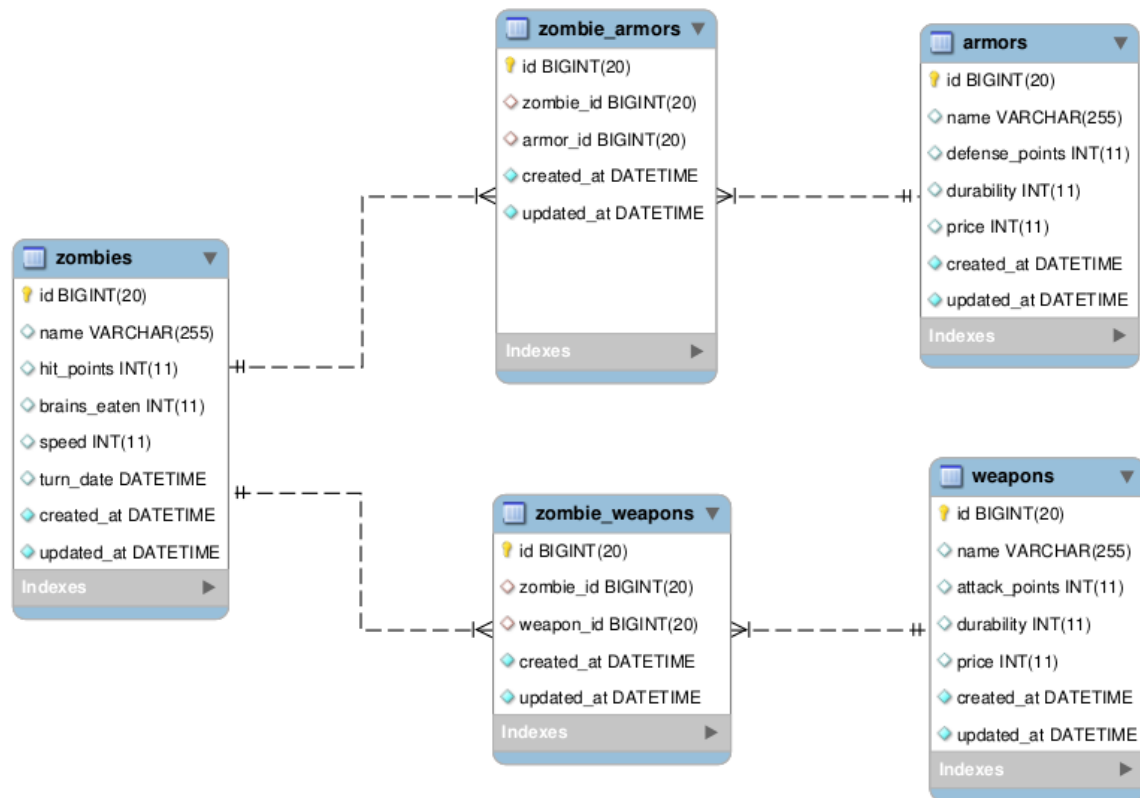


Modelo E-R



Routes

Routes match in priority from top to bottom

Helper Path / Url	HTTP Verb	Path	Controller#Action
zombies_path	GET	/zombies(.:format)	v1/zombies#index
	POST	/zombies(.:format)	v1/zombies#create
zombie_path	GET	/zombies/:id(.:format)	v1/zombies#show
	PATCH	/zombies/:id(.:format)	v1/zombies#update
	PUT	/zombies/:id(.:format)	v1/zombies#update
	DELETE	/zombies/:id(.:format)	v1/zombies#destroy
weapons_path	GET	/weapons(.:format)	v1/weapons#index
	POST	/weapons(.:format)	v1/weapons#create
weapon_path	GET	/weapons/:id(.:format)	v1/weapons#show
	PATCH	/weapons/:id(.:format)	v1/weapons#update
	PUT	/weapons/:id(.:format)	v1/weapons#update
	DELETE	/weapons/:id(.:format)	v1/weapons#destroy
armors_path	GET	/armors(.:format)	v1/armors#index
	POST	/armors(.:format)	v1/armors#create
armor_path	GET	/armors/:id(.:format)	v1/armors#show
	PATCH	/armors/:id(.:format)	v1/armors#update
	PUT	/armors/:id(.:format)	v1/armors#update
	DELETE	/armors/:id(.:format)	v1/armors#destroy
search_path	GET	/search(.:format)	v1/zombies#search

Search

El requerimiento de buscar zombies por atributos lo desrrolle de la siguiente formatter

Un Zombie puede ser ubicado de acuerdo a los siguientes parametros en el querystring

key puede tomar los valores propios del atributo del Zombie “hit_points”, “brains_eaten” y “speed”
value contendra el valor buscado según la key seleccionada

Como parametros adicionales opcionales podemos filtrar por id de las Armaduras (armorid) o de las Armas (weaponid)

Ejemplo:

<http://localhost:3000/search?key=speed&value=1&weaponid=1>

Selecciona los Zombies que tenga speed=1 y tengan una Armadura con Id=1

<http://localhost:3000/search?key=speed&value=1&armorid=20>

Selecciona los Zombies que tenga speed=1 y tengan una Arma con Id=20

http://localhost:3000/search?key=hit_points&value=1&armorid=20

Selecciona los Zombies que tenga hit_points=1 y tengan una Arma con Id=20

http://localhost:3000/search?key=hit_points&value=1&armorid=1&weaponid=1

Selecciona los Zombies que tenga hit_points=1 y tengan una Arma con Id=1 y una Armadura con Id=1

<http://localhost:3000/search>

Busca todos los Zombies es equivalente a <http://localhost:3000/zombies>

Nota: En los requerimientos no estaba especificado como se realizarian las busquedas, estas se pueden realizar según sea necesaria por ejemplo en vez de buscar por Id del Arma buscar por Name del Arma. Todas estas especificaciones deben estar bien claras ya que implican cambios en el modelo de datos posibles indices alternos para hacer mejor la busqueda, validaciones para no permitir null, etc.

Los otros metodos son los normales en un CRUD, estos se desarrollaron para Armor, Weapon y Zombie, detalle para Zombie es similar para las otras entidades

INDEX

localhost:3000/zombies

Listar todos los Zombies

```
[
  {
    "id": 2,
    "name": "Cyborg Stealy",
    "hit_points": 5,
    "speed": 9,
    "brains_eaten": 30,
    "turn_date": "2016-08-10T00:00:00.000Z"
  },
  {
    "id": 3,
    "name": "Cyborg Mr. Meeseeks",
    "hit_points": 2,
    "speed": 4,
    "brains_eaten": 29,
    "turn_date": "2017-11-25T00:00:00.000Z"
  },
  .....
  {
    "id": 52,
    "name": "Antonio Franceskin1",
    "hit_points": 10,
    "speed": 10,
    "brains_eaten": 10,
```

```
    "turn_date": "2018-03-14T00:00:00.000Z"
  }
]
```

Show

localhost:3000/zombies/2 con el metodo GET

Lista el Zombie con Id=2

```
{
  "zombie": {
    "id": 2,
    "name": "Cyborg Stealy",
    "hit_points": 5,
    "speed": 9,
    "brains_eaten": 30,
    "turn_date": "2016-08-10T00:00:00.000Z"
  }
}
```

Update

localhost:3000/zombies/2 con el metodo PATCH pasandole como parametros

```
{
  "zombie": {
    "id": 2,
    "name": "Cyborg Stealy Change",
    "hit_points": 5,
    "speed": 9,
    "brains_eaten": 30,
    "turn_date": "2016-08-10T00:00:00.000Z"
  }
}
```

Resultado

```
{
  "zombie": {
    "id": 2,
    "name": "Cyborg Stealy Change",
    "hit_points": 5,
    "speed": 9,
    "brains_eaten": 30,
    "turn_date": "2016-08-10T00:00:00.000Z"
  }
}
```

Destroy

<localhost:3000/zombies/2> con el metodo DELETE pasandole como parametros

```
{
  "zombie": {
    "id": 2,
    "name": "Cyborg Stealy Change",
    "hit_points": 5,
    "speed": 9,
    "brains_eaten": 30,
    "turn_date": "2016-08-10T00:00:00.000Z"
  }
}
```

Elimina el registro

Create

localhost:3000/zombies con el metodo POST con los parametros

```
{
  "zombie": {
    "name": "Cyborg Stealy Change",
    "hit_points": 5,
    "speed": 9,
    "brains_eaten": 30,
    "turn_date": "2016-08-10T00:00:00.000Z"
  }
}
```

Inserta el registro con Id=53

```
{
  "zombie": {
    "id": 53,
    "name": "Cyborg Stealy Change",
    "hit_points": 5,
    "speed": 9,
    "brains_eaten": 30,
    "turn_date": "2016-08-10T00:00:00.000Z"
  }
}
```

Nota General: Use Postman para probar toda la aplicación, se monto en GitHub en el siguiente enlace:

<https://github.com/AntonioFranceskin/coding-challenge-backend.git> donde esta toda la API

y en Heruko se monto la aplicación

<https://testbadi.herokuapp.com/>