DOCUMENTACIÓN:

TECNOLOGÍAS:

NODEJS + EXPRESS

POSTGRESS SQL 15

REACTJS + VITE

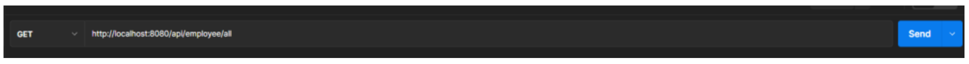
IMPORTANTE: Se requiere el comando **npm install** tanto para el cliente como el servedor para instalar los modulos necesarios del software

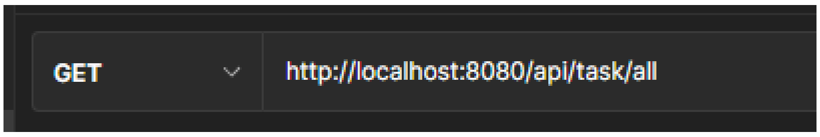
ESTRUCTURA SQL POSTGRESS:

```
create table employee (
id serial primary key,
f_name varchar(25),
l_name varchar(25)
);

create table task (
id serial primary key,
tittle varchar(25),
description varchar(45)
);

alter table employee add column fk_task serial;

alter table employe add constraint fk_task foreign key (fk_task) references task (id);
```

RUTAS API:

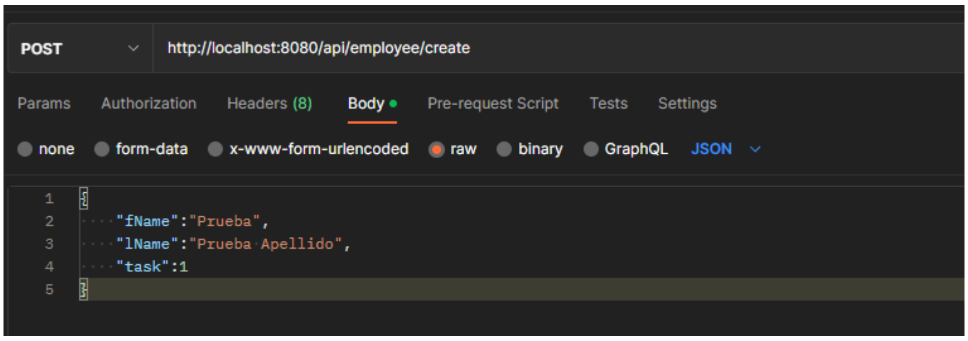
GET:


GET http://localhost:8080/api/employee/all


GET http://localhost:8080/api/task/all

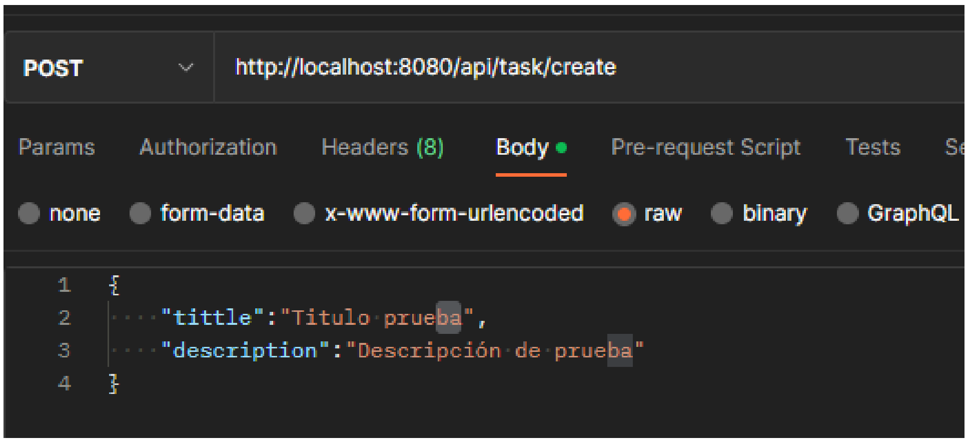
POST:

Controlador para crear empleados:


POST http://localhost:8080/api/employee/create

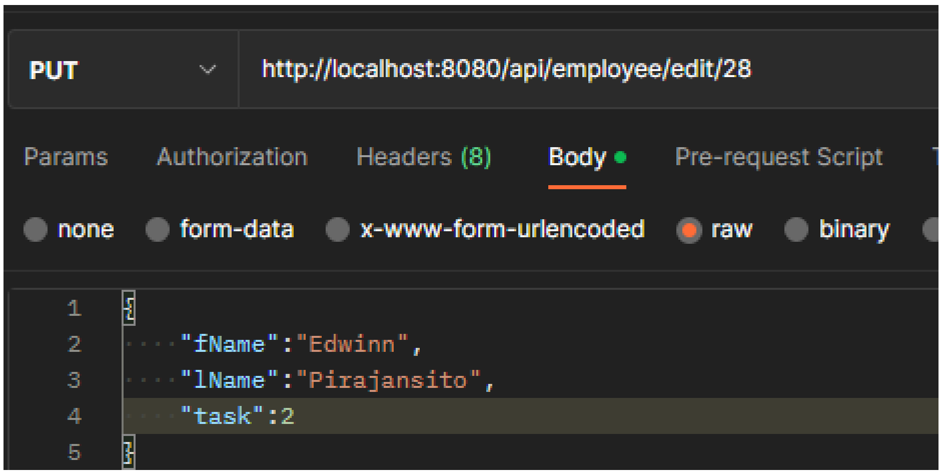
```
{
    "fName":"Prueba",
    "lName":"Prueba Apellido",
    "task":1
}
```

Controlador para crear cargos:


POST http://localhost:8080/api/task/create

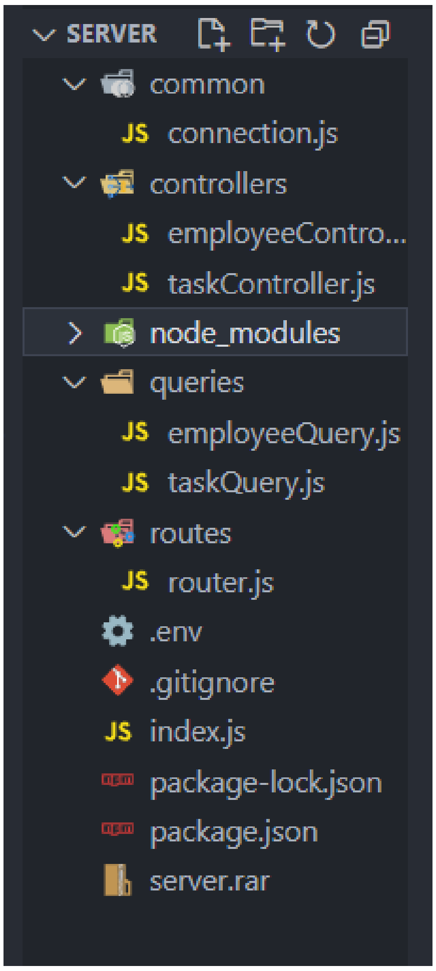
```
{
    "tittle":"Titulo prueba",
    "description":"Descripción de prueba"
}
```

Controlador para editar empleados:


PUT http://localhost:8080/api/employee/edit/28

```
{
    "fName":"Edwinn",
    "lName":"Pirajansito",
    "task":2
}
```

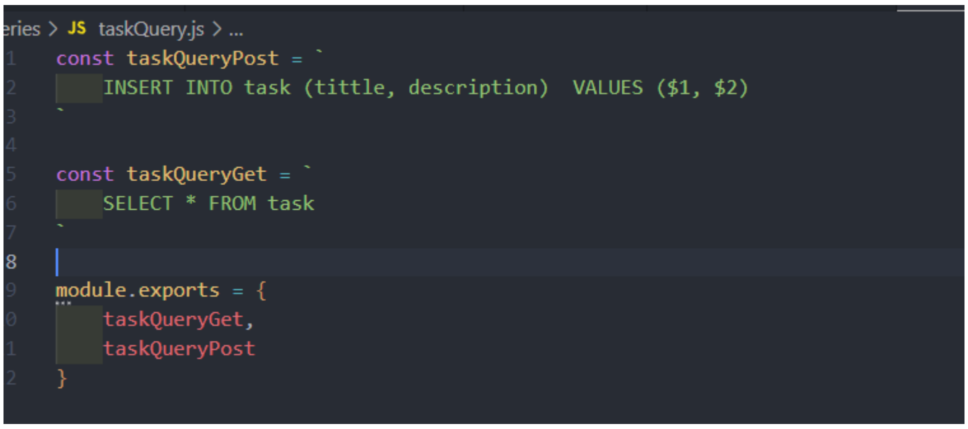
ESTRUCTURA DEL PROYECTO:



```
SERVER
  common
    connection.js
  controllers
    employeeContro...
    taskController.js
  node_modules
  queries
    employeeQuery.js
    taskQuery.js
  routes
    router.js
  .env
  .gitignore
  index.js
  package-lock.json
  package.json
  server.rar
```

CONSULTAS SQL:

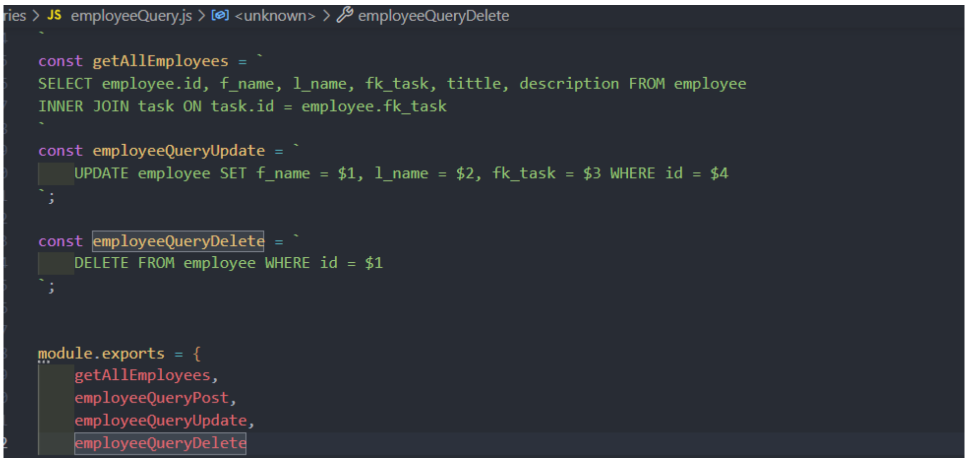
Task Query:



eries > JS taskQuery.js > ...

```
const taskQueryPost = `
    INSERT INTO task (tittle, description)  VALUES ($1, $2)
`

const taskQueryGet = `
    SELECT * FROM task
`

module.exports = {
    taskQueryGet,
    taskQueryPost
}
```

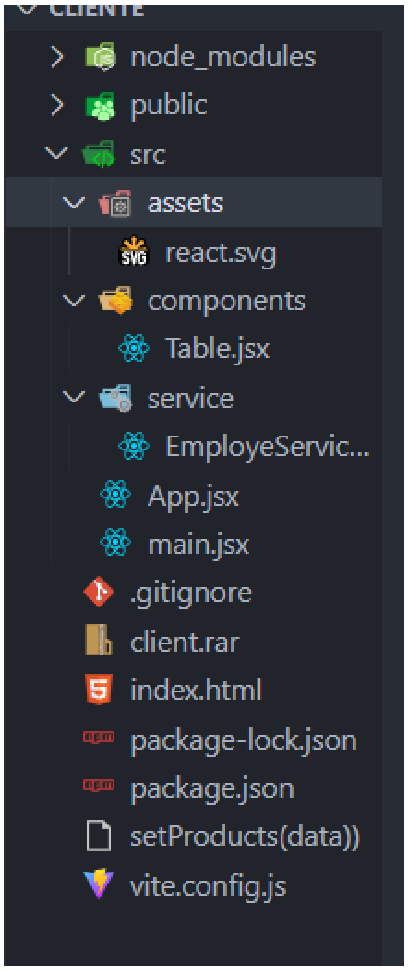
EmployeeQuery:



ries > JS employeeQuery.js > <unknown> > employeeQueryDelete

```
`
const getAllEmployees = `
SELECT employee.id, f_name, l_name, fk_task, tittle, description FROM employee
INNER JOIN task ON task.id = employee.fk_task
`
const employeeQueryUpdate = `
    UPDATE employee SET f_name = $1, l_name = $2, fk_task = $3 WHERE id = $4
`;

const employeeQueryDelete = `
    DELETE FROM employee WHERE id = $1
`;


module.exports = {
    getAllEmployees,
    employeeQueryPost,
    employeeQueryUpdate,
    employeeQueryDelete
```

ESTRUCTURA CLIENTE:



```
CLIENTE
  > node_modules
  > public
  v src
    v assets
        react.svg
    v components
        Table.jsx
    v service
        EmployeServic...
      App.jsx
      main.jsx
    .gitignore
    client.rar
    index.html
    package-lock.json
    package.json
    setProducts(data))
    vite.config.js
```