proyecto de base de datos

# software

Software usado para este Proyecto.

## Postgresql

PostgreSQL es un manejador de base de datos open source y compatible con muchas tecnologías web existentes, como Node.js y existen varios servicios en la web que tienen hosting gratis de esta base de datos (con limitaciones).

Tiene soporte multiplataforma e interfaces graficas gratis, lo que hace aún más fácil su uso en PC y Mac. Por eso es que elegimos este manejador.

## Node.js

Node.js es una plataforma de software que utiliza JavaScript como lenguaje principal, esta plataforma tiene paquetes (módulos) que se pueden usar en tu proyecto de node.js.

### Express.js

Node.js también se puede usar para crea run servidor HTTP, para este objetivo se utiliza el paquete Express.js, el cual tiene un pequeño framework para crear aplicaciones web.

### Pg

Pg es otro paquete que permite la conexión con un servidor PostgreSQL.

## Visual Studio 2015

Usamos Visual Studio 2015 por que tiene soporte para Node.js por medio de un plugin.

## heroku.com

Heroku es un proveedor de servicios web, nos permite echar a andar un servidor Node.js con una base de datos P0stgreSQL en la web. Estamos usando los servicios que son gratis.

# metodo de conexión

Heroku provee una cadena de conexión con la que podemos acceder a la base de datos de la web. Este es un ejemplo:

postgres://hqfhanijqvgone:6xJGcaWJwZjK8Zf6aPHMnaOu\_3@ec2-54-163-251-104.compute-1.amazonaws.com:5432/d5aaibrcs9n71b

De esta manera nuestra aplicación web que corre bajo Node.js se conecta usando el paquete de pg a la base de datos proveída por Heroku. (Para hacer tests tenemos una base de datos local en nuestras maquinas).

### Ejemplo

var connectionString = "postgres://hqfhanijqvgone:6xJGcaWJwZjK8Zf6aPHMnaOu\_3@ec2-54-163-251-104.compute-1.amazonaws.com:5432/d5aaibrcs9n71b";

/\* GET home page. \*/

router.get('/', function (req, res, next) {

var subjects = [];

// Get a Postgres client from the connection pool

pg.connect(connectionString, function (err, client, done) {

// Handle connection errors

if (err) {

done();

return res.status(500).json({ success: false, data: err });

}

var query = client.query(

"SELECT materia.clave, grupo.seccion, materia.nombre" +

" FROM grupo INNER JOIN materia ON grupo.materia\_clave = materia.clave ");

query.on('row', function (row) {

subjects.push(row);

});

query.on('error', function (err) {

done();

res.status(500).json({ err: err });

});

query.on('end', function (row) {

done();

res.render('tarea1', {

subjects: subjects,

subjectID: 0,

subject: 'Ninguna',

teacher: 'none',

section: 'none',

shedule: 'none',

students : [

]

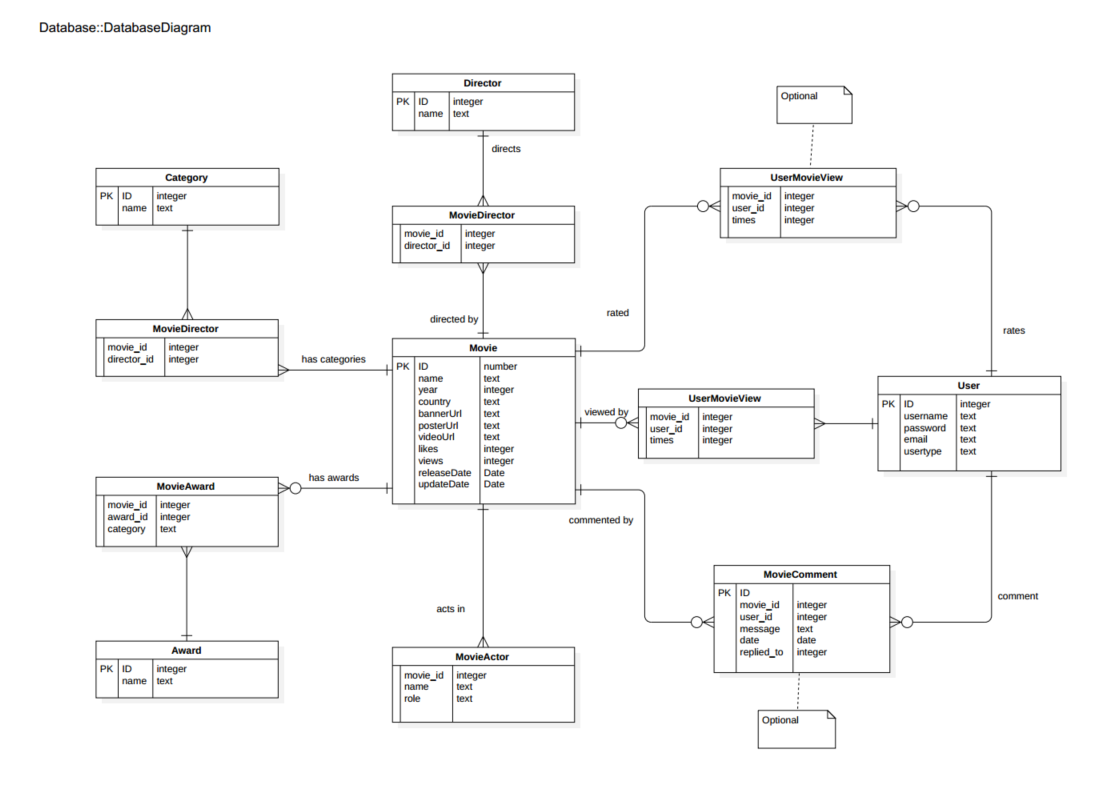
});

});

});

});

# Diagrama



# Pantallas

