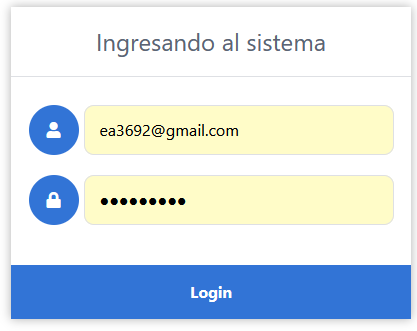
# Requerimientos

* [Postman](https://www.postman.com/downloads/)
* Visual Studio Code
* NodeJs

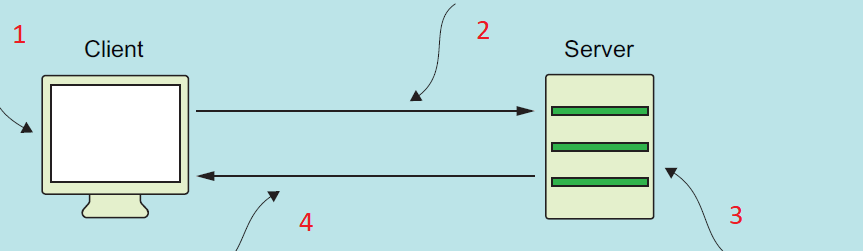
# Introducción

Crear una aplicación que permita al usuario registrarse



[Node.js](https://nodejs.org/en/about/) es un programa asíncrono en tiempo de ejecución de JavaScript basado en eventos.

* NodeJS utiliza el motor de JavaScript
* NodeJS utiliza Chrome V8 Google
* Cuando se inicia una aplicación JavaScript, todo el código de esa aplicación se carga en memoria.
* JavaScript es asíncrono por naturaleza.



1. El cliente representa al navegador Web sobre el cual el cliente puede ver tu aplicación.
2. El cliente envía una petición al servidor, sabe que datos le gustaría enviar o recibir.
3. El servidor es donde tu aplicación se ejecuta y maneja algún dato enviado por el usuario.
4. El servidor responde al cliente por cargar una página o enviar datos.

Crear un proyecto express con handlebars

Crear un proyecto con la vista [Handlebars](https://handlebarsjs.com/)

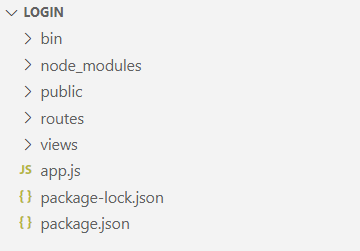
**express login --view=hbs**

Cambiarse al directorio recien creado con el siguiente comando

**cd login**

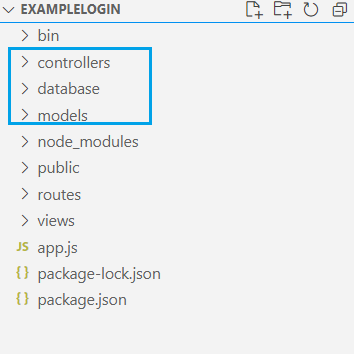
Instalar dependencias definidas en archivo package.json

**npm install**



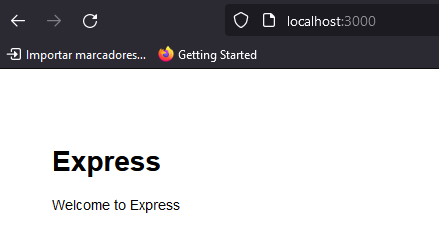
# Definir la estructura del proyecto mediante el patrón MVC

Agregar tres carpetas para conexión a BD, controladores y modelos



Iniciar la aplicación

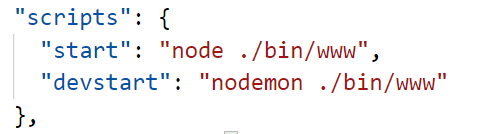
**SET DEBUG=login:\* & npm start**



# Habilitar el reinicio del servidor con Nodemon

Instalar el módulo

**npm install --save-dev nodemon**

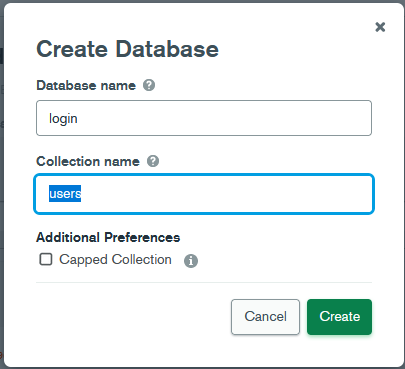


Ejecutar nuevamente con la nueva instrucción

**SET DEBUG=login:\* & npm run devstart**

# Configurando base de datos MongoDB

Crear una base de datos en MongoDb Atlas y llamarla login y una colección llamada Users



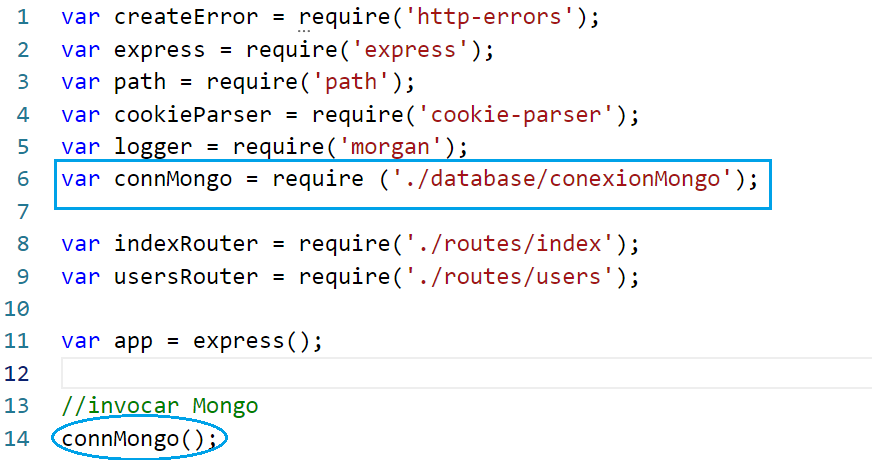
Instalar módulo [Mongoose](https://mongoosejs.com/docs/schematypes.html) para conexión a MongoDB Atlas

**npm install mongoose**

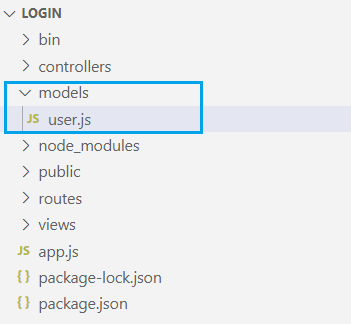
Agregar el archivo database/conexionMongo.js

|  |
| --- |
| //Configurar conexión Mongoose  function connMongo() {      var mongoose = require('mongoose');      var mongoDB = 'mongodb+srv://gds05:gds05@cluster0.jnmbt.mongodb.net/login?retryWrites=true&w=majority';      mongoose.connect(mongoDB, { useNewUrlParser: true , useUnifiedTopology: true});      var db = mongoose.connection;      db.on('error', console.error.bind(console, 'Error de Conexion a MongoDB'));  }  module.exports = connMongo; |

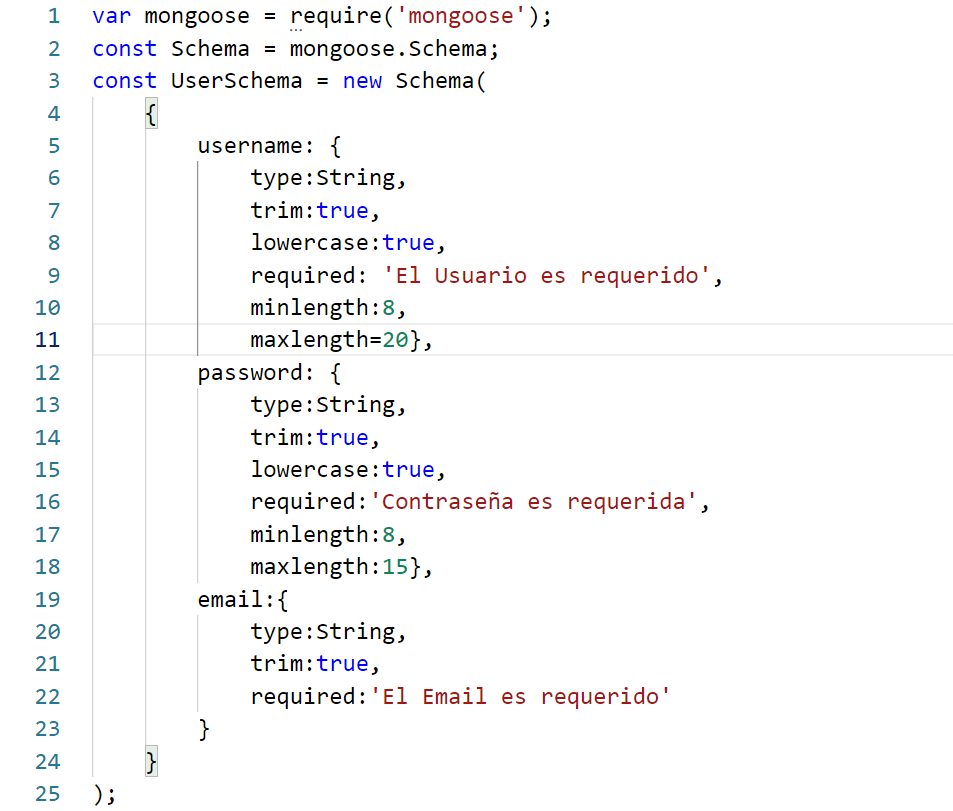
Modificar el archivo app.js



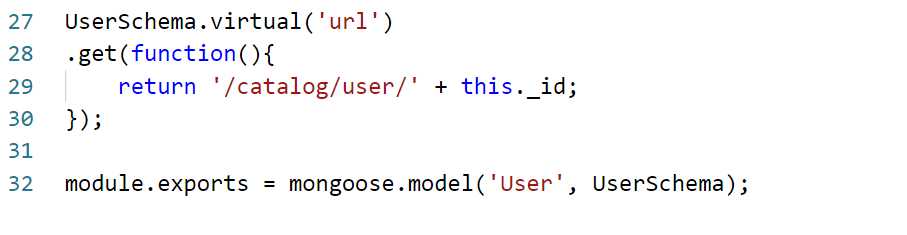
Crear el archivo **models/user.js**



Definiendo esquema para modelo user



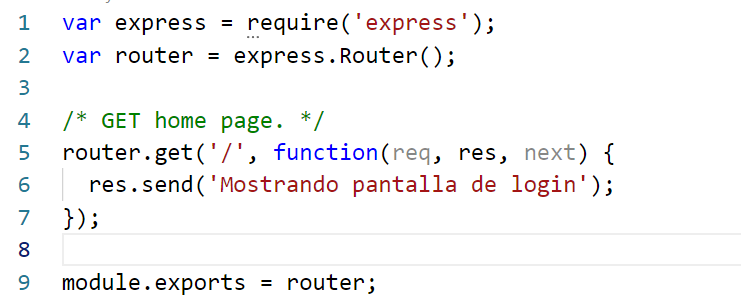
Agregando función virtual



# **Rutas y Controladores**



Modificar el archivo **routes/index.js**



Desplegando la vista

Véase referencia [Handlebars](https://handlebarsjs.com/)

Modificar plantilla views/layout.hbs

|  |
| --- |
| <!DOCTYPE html>  <html>    <head>      <title>{{title}}</title>      <link rel='stylesheet' href='/stylesheets/style.css' />      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">      <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">    </head>    <body>      {{{body}}}      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>    </body>  </html> |

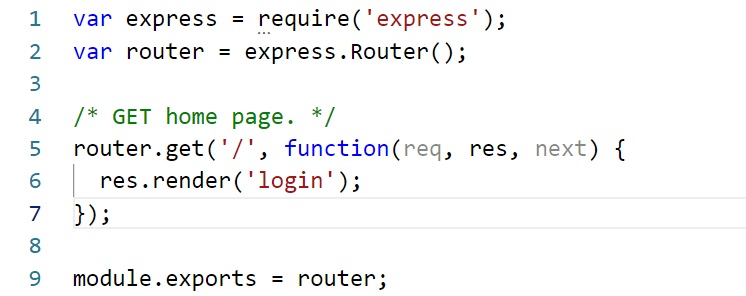
Agregar la hoja de estilo para la fuente

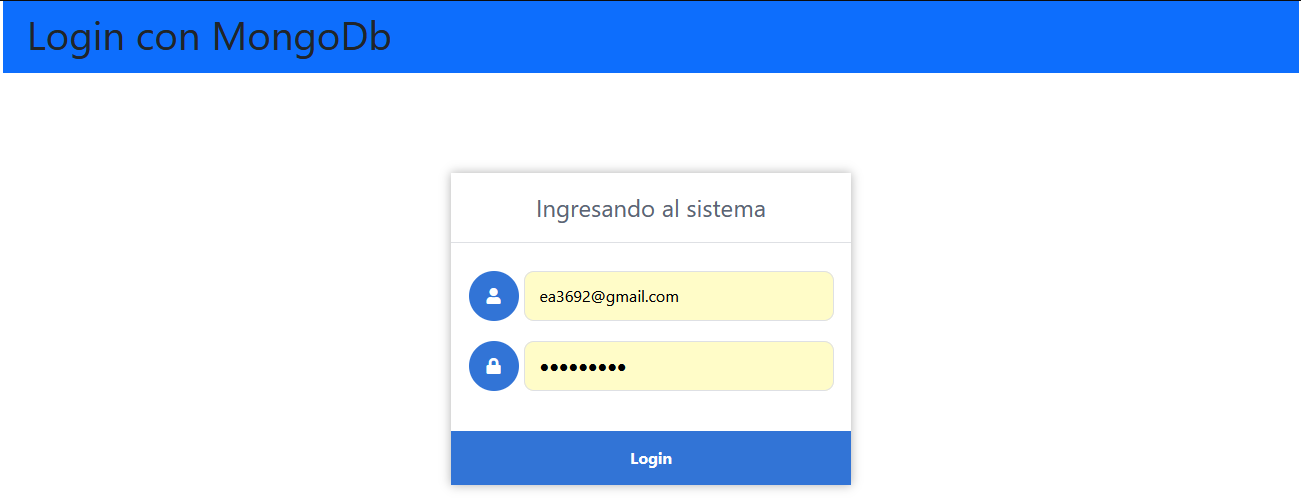
<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">

Creando la vista Login **views/login.hbs**

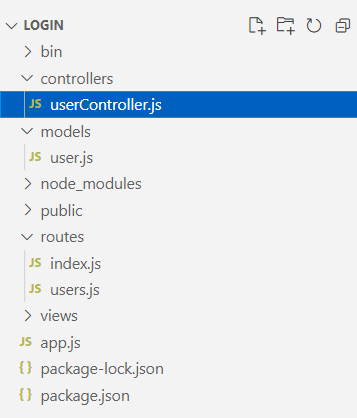
|  |
| --- |
| <div class="container login">  <h1>Ingresando al sistema</h1>    <form action="/login/verify" method="post">             <!--  usuario -->            <label for="username"> <i class="fas fa-user"></i></label>            <input type="text" name="username" placeholder="Usuario" id="username" required>            <!-- Contraseña -->            <label for="password"><i class="fas fa-lock"></i></label>            <input type="password" name="password" placeholder="Contraseña" id="password" required>            <input type="submit" value="Login">          </form>  </div> |

Modificar el archivo routes/index.js

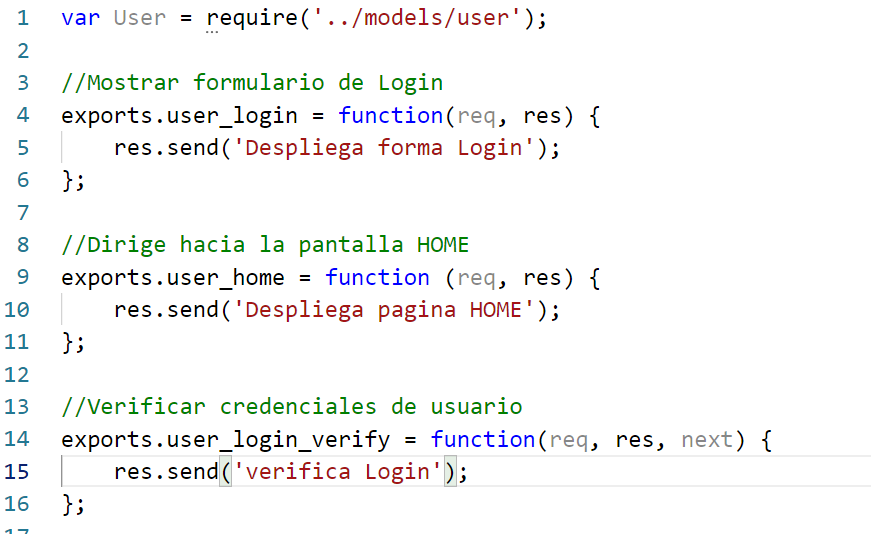




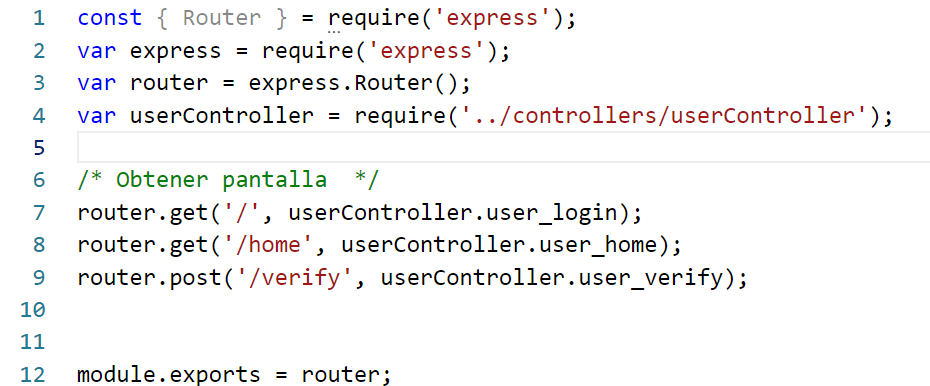
Crear el controlador **controllers/userController.js**



Definiendo funcionalidades de Controlador



Modificar el archivo routes/index.js

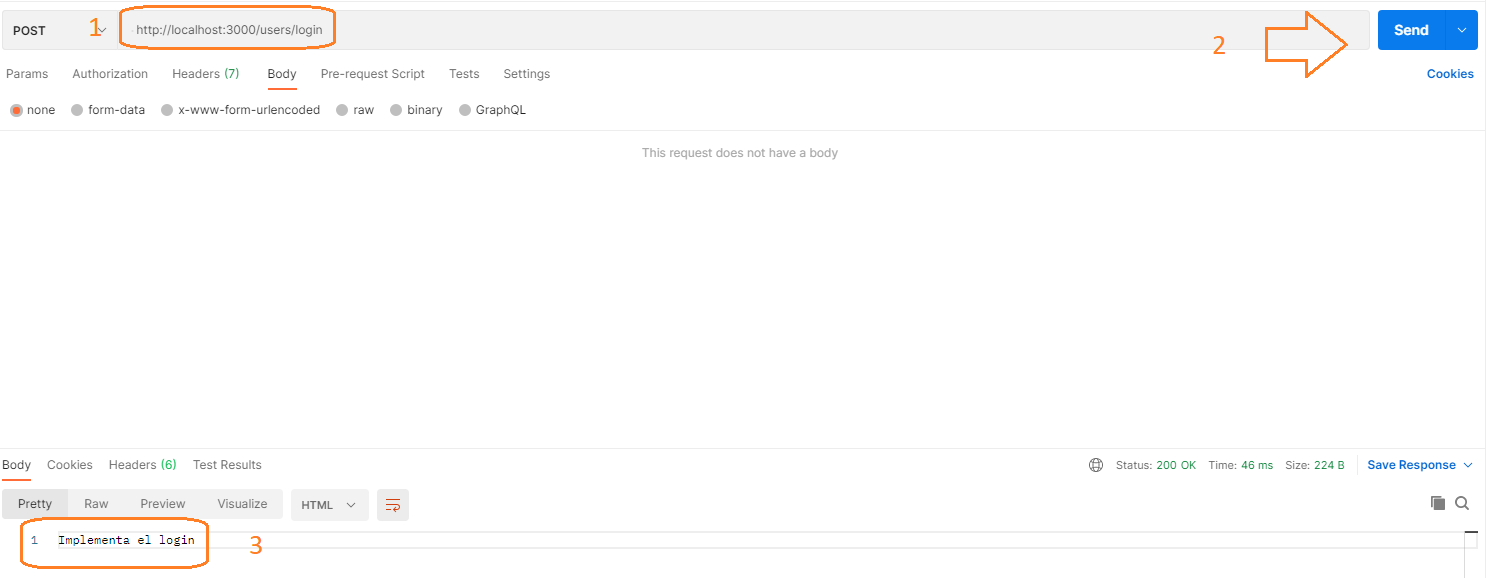


Prueba las rutas

[http://localhost:3000/](http://localhost:3000/users)

<http://localhost:3000/home>

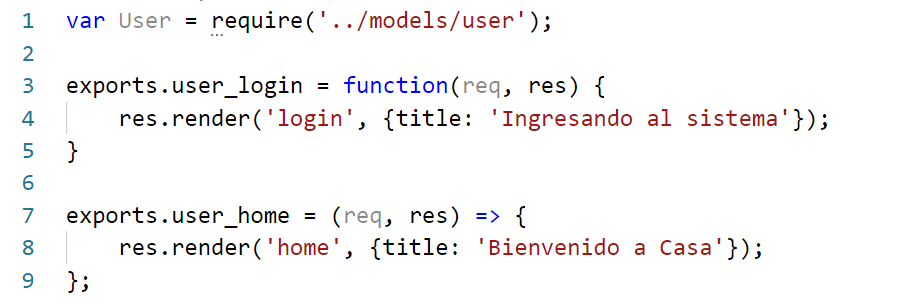
<http://localhost:3000/verify>



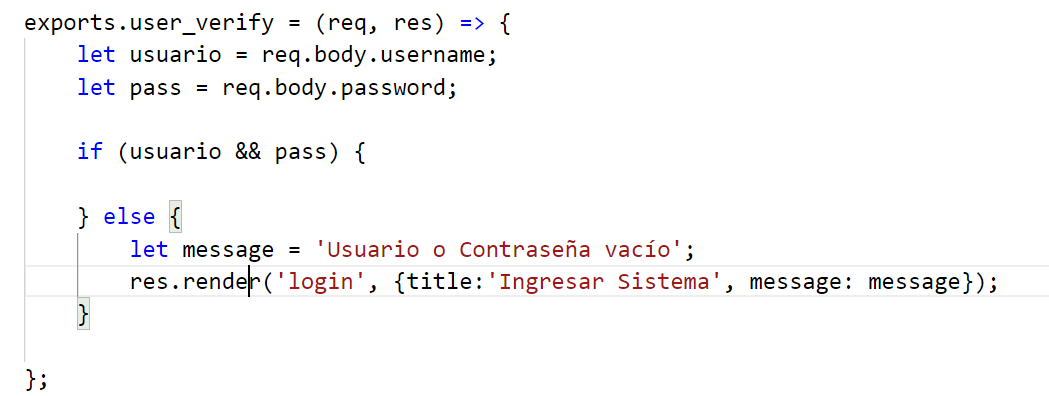
Instalar modulo

**npm install --save express-validator**

Modificar el controlador **controllers/userController**.js para desplegar vistas de login y home



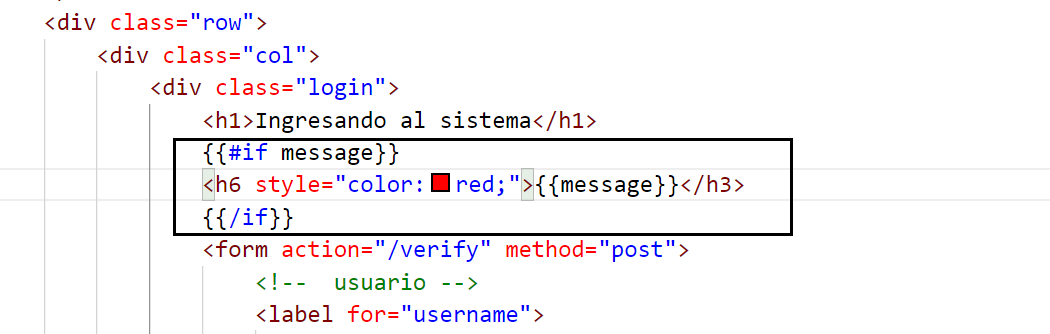
Maneja verificación de login que no sean datos vacíos



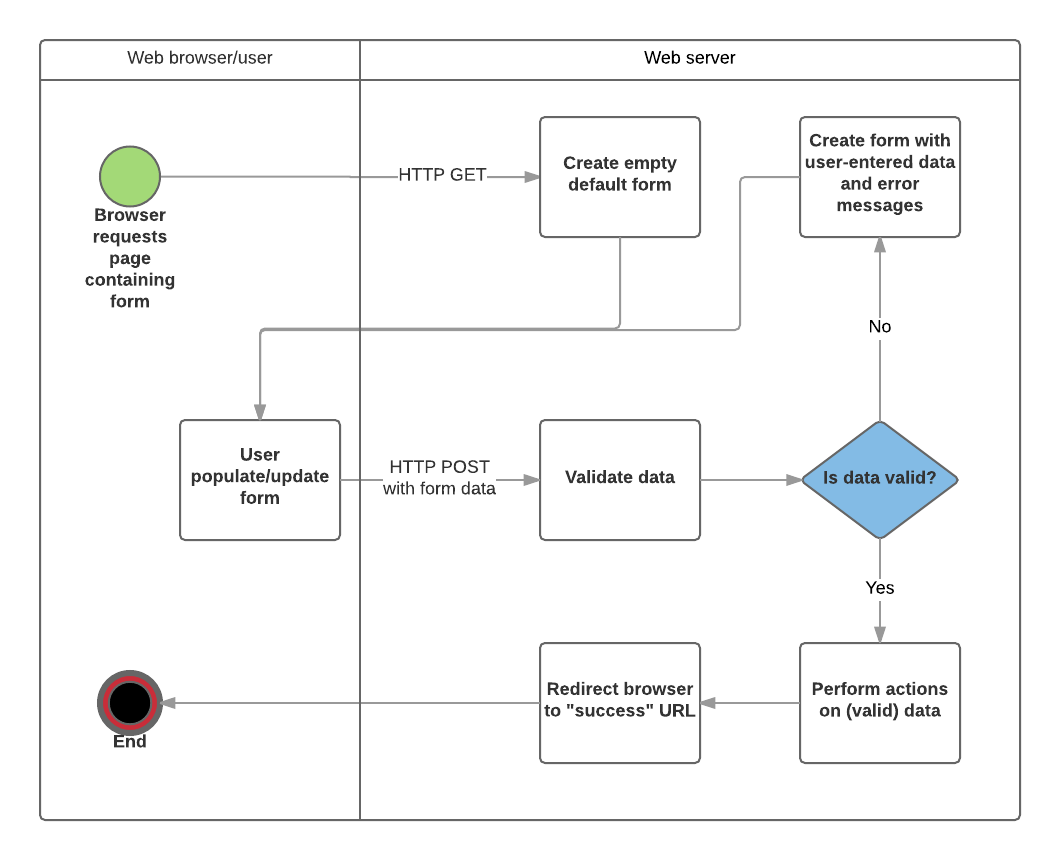
Accede a la base de datos

|  |
| --- |
| var User = require('../models/user');  const { body, validationResult } = require('express-validator');  exports.user\_login = function(req, res) {      res.render('login', {title: 'Ingresando al sistema'});  }  exports.user\_home = (req, res) => {      res.render('home', {title: 'Bienvenido a Casa'});  };  exports.user\_verify = (req, res) => {      let usuario = req.body.username;      let pass = req.body.password;      console.log('Usuario ' + usuario + ' Password: ' + pass);      if (usuario && pass) {          User.find({'username': usuario, 'password': pass}, function(error, results) {              if (error){                  let message = 'Hubo un error de conexion a Mongo';                  res.render('login', {title:'Ingresar Sistema', message:message});              }              if (results.length > 0) {                  res.render('index', {title:'Bienvenido ' + usuario});              } else {                  let mensaje = 'Usuario o contraseña incorrecto';                  let titulo = 'Bienvenido ' + usuario;                  res.render('login', {title:titulo, message: mensaje });              }          });      } else {          let message = 'Usuario o Contraseña vacío';          res.render('login', {title:'Ingresar Sistema', message: message});      }    }; |

Haz cambios en la vista



Trabajando con formas



Instalar modulo de validación

npm install express-validator

Ir al controlador controller/userController

