



## Desarrollo de FullStack

Arturo Daniel Marroquin Fuentes AI07116862

Diego Villarreal Martinez AI07064821

Profesor: Francisco Gomez Rubio

# Links que se utilizarán a lo largo del proyecto:

<https://github.com/WebbiestStew/acts-fullstack.git> <- Repositorio de github

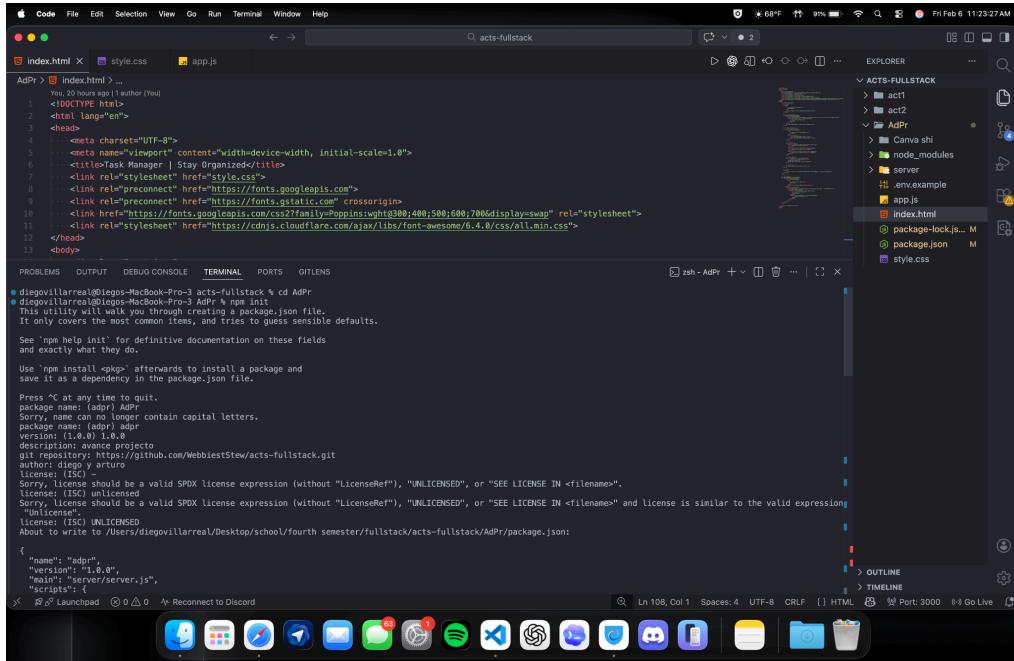
<https://webbieststew.github.io/acts-fullstack/AdPr> <- Pagina desplegada en la nube

# Introducción

Se nos dio de Avance de proyecto que hagamos una página que incluya una estructura bien organizada, las operaciones CRUD (create, read, update, delete), autenticación con JWT, una interfaz atractiva, la aplicación también será desplegada localmente, y no fue requerido, pero también será desplegada en la nube, mediante la herramienta de deploy de Github (se usará para la parte del Frontend), y de una página llamada “Render” (Que se usará para el backend), en este documento se dará una detallada explicación de lo que viene siendo todos los archivos que contiene este proyecto (a excepción de las carpetas de los módulos de Node)

# A)Configuración del proyecto

Primero se inicializo el proyecto con “npm init”:



```
You 20 hours ago | 1 author (You)
  You 20 hours ago | 1 author (You)
  <!DOCTYPE html>
  <html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task Manager | Stay Organized</title>
    <link rel="stylesheet" href="style.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
  </head>
  <body>
```

See 'npm help init' for definitive documentation on these fields and exactly what they do.

Use 'npm install <package>' afterwards to install a package and save it as a dependency in the package.json file.

Press `C` at any time to quit.

package name: (adPr) AdPr  
Sorry, name can't contain capital letters.  
package name: (adPr) adpr  
version (1.0.0) 1.0.0  
description (optional)  
url repository: https://github.com/WebbiestStew/acts-fullstack.git  
author: diego y arturo  
license: (ISC)  
Sorry, license should be a valid SPDX license expression (without "LicenseRef"), "UNLICENSED", or "SEE LICENSE IN <filename>".  
license: (ISC) unlicensed  
Sorry, license should be a valid SPDX license expression (without "LicenseRef"), "UNLICENSED", or "SEE LICENSE IN <filename>" and license is similar to the valid expression  
"unlicensed"  
license: (ISC) UNLICENSED  
About to write to /Users/diegovillarreal/Desktop/school/fourth semester/fullstack/acts-fullstack/adPr/package.json:

```
{
  "name": "adpr",
  "version": "1.0.0",
  "main": "server/server.js",
  "scripts": {
    "start": "node -r module-merge server/server.js",
    "dev": "node -r module-merge server/server.js",
    "test": "echo \"No tests yet\" && exit 0"
  },
  "keywords": [],
  "author": "diego y arturo",
  "license": "UNLICENSED",
  "description": "avance proyecto",
  "dependencies": {
    "bcryptjs": "2.4.3",
    "express": "4.18.2",
    "dotenv": "17.2.3",
    "express-validator": "9.0.3",
    "pg": "8.18.0"
  },
  "devDependencies": {},
  "repository": {
    "type": "git",
    "url": "git+https://github.com/WebbiestStew/acts-fullstack.git"
  },
  "bugs": {
    "url": "https://github.com/WebbiestStew/acts-fullstack/issues"
  },
  "homepage": "https://github.com/WebbiestStew/acts-fullstack/readme"
}
```



```
AdPr > index.html > ...
You 20 hours ago | 1 author (You)
  <!DOCTYPE html>
  <html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task Manager | Stay Organized</title>
    <link rel="stylesheet" href="style.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
  </head>
  <body>
```

Ln 108, Col 1 Spaces: 4 UTF-8 CRLF [ ] HTML ⚡ Port: 3000 🔍 Go Live

Despues optamos por usar la base de datos PostgreSQL por su simplicidad:

```
zsh: command not found: dotenv
● diegovillarreal@Diegos-MacBook-Pro-3 AdPr % npm install pg

  up to date, audited 98 packages in 1s

  25 packages are looking for funding
    run `npm fund` for details

  found 0 vulnerabilities
○ diegovillarreal@Diegos-MacBook-Pro-3 AdPr %
```

Despues de instalo todas las dependencias que se mencionaron:

```
● diegovillarreal@Diegos-MacBook-Pro-3 AdPr % npm install express

  up to date, audited 98 packages in 1s

  25 packages are looking for funding
    run `npm fund` for details

  found 0 vulnerabilities
● diegovillarreal@Diegos-MacBook-Pro-3 AdPr % npm install jsonwebtoken

  up to date, audited 98 packages in 909ms

  25 packages are looking for funding
    run `npm fund` for details

  found 0 vulnerabilities
● diegovillarreal@Diegos-MacBook-Pro-3 AdPr % npm install bcryptjs

  up to date, audited 98 packages in 816ms

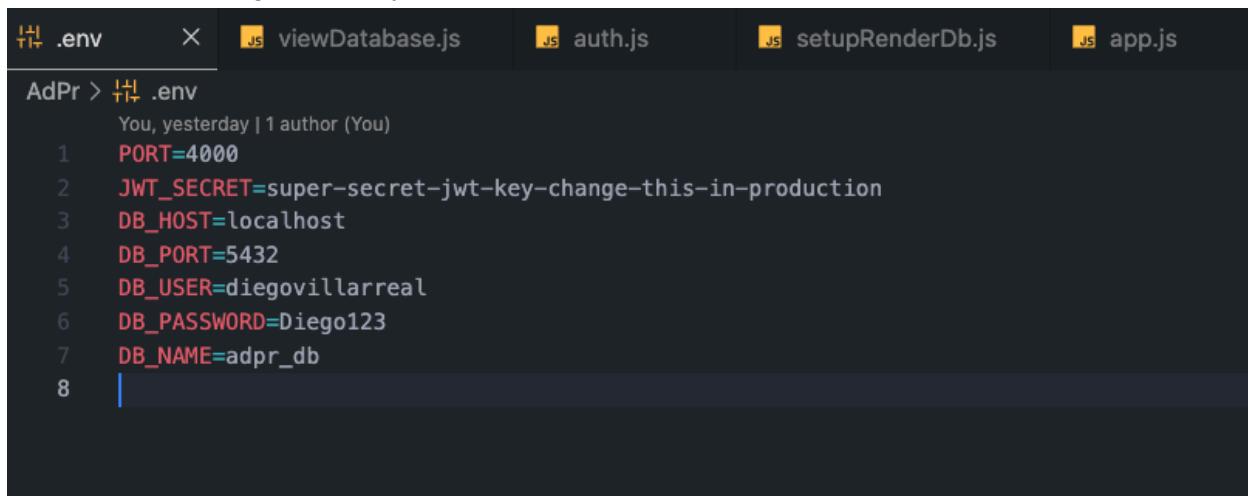
  25 packages are looking for funding
    run `npm fund` for details

  found 0 vulnerabilities
● diegovillarreal@Diegos-MacBook-Pro-3 AdPr % npm install cors

  up to date, audited 98 packages in 821ms

  25 packages are looking for funding
    run `npm fund` for details
```

Y despues se configuro el proyecto con el dotenv



The screenshot shows a terminal window with several tabs at the top: '.env' (selected), 'viewDatabase.js', 'auth.js', 'setupRenderDb.js', and 'app.js'. The '.env' tab contains the following configuration variables:

```
AdPr > .env
You, yesterday | 1 author (You)
1 PORT=4000
2 JWT_SECRET=super-secret-jwt-key-change-this-in-production
3 DB_HOST=localhost
4 DB_PORT=5432
5 DB_USER=diegovillarreal
6 DB_PASSWORD=Diego123
7 DB_NAME=adpr_db
8 |
```

# B)Backend

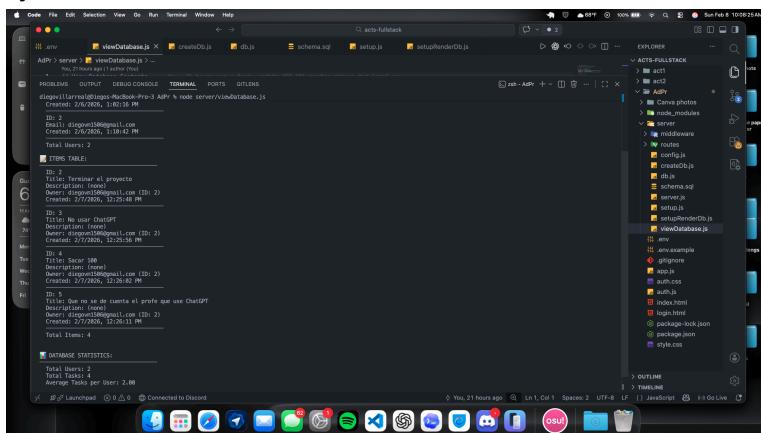
Se hizo un documento llamado “[server.js](#)” para poder configurar el servidor que se utilizara para poder visualizar la pagina:

```
You, 2 days ago | 1 author (You)
1 const express = require('express');      You, 2 days ago * feat: initiali...
2 const cors = require('cors');
3 const { validateEnv, config } = require('./config');
4 const authRoutes = require('./routes/auth');
5 const itemRoutes = require('./routes/items');
6 const { notFound, errorHandler } = require('./middleware/errorHandler');
7
8 validateEnv();
9
10 const app = express();
11
12 app.use(cors());
13 app.use(express.json());
14
15 app.get('/health', (req, res) => {
16   res.json({ status: 'ok' });
17 });
18
19 app.use('/api/auth', authRoutes);
20 app.use('/api/items', itemRoutes);
21
22 app.use(notFound);
23 app.use(errorHandler);
24
25 app.listen(config.port, () => {
26   console.log(`Server running on port ${config.port}`);
27 });
28
```

Después se hicieron los archivos para la base de datos en PostgreSQL:  
 Primero para ver la base de datos en consola:

```
You, 21 hours ago | author (You)
1 // View Database Contents      You, 21 hours ago + feat: update API URL configuration
2 // Run this with: node server/viewDatabase.js
3
4 const { query } = require('./db');
5
6 async function viewDatabase() {
7   console.log('=====');
8   console.log(` DATABASE CONTENTS`);
9   console.log('=====\\n');
10
11   try {
12     // View all users
13     console.log(` USERS TABLE:`);
14     console.log(`-----`);
15     const users = await query('SELECT id, email, created_at FROM users ORDER BY id');
16
17     if (users.rows.length === 0) {
18       console.log(` (No users found)`);
19     } else {
20       users.forEach(user => {
21         console.log(` - ID: ${user.id}`);
22         console.log(` - Email: ${user.email}`);
23         console.log(` - Created: ${new Date(user.created_at).toLocaleString()}`);
24         console.log(` -----`);
25       });
26     }
27     console.log(` Total Users: ${users.rows.length}\\n`);
28
29     // View all items
30     console.log(` ITEMS TABLE:`);
31     console.log(`-----`);
32     const items = await query(
33       `SELECT items.id, items.title, items.description,
34        items.owner_id, users.email as owner_email,
35        items.created_at
36       FROM items
37      JOIN users ON items.owner_id = users.id
38      ORDER BY items.id
39    `);
40
41     if (items.rows.length === 0) {
42       console.log(` (No items found)`);
43     } else {
44       items.forEach(item => {
45         console.log(` - ID: ${item.id}`);
46         console.log(` - Title: ${item.title}`);
47         console.log(` - Description: ${item.description || '(none)'} `);
48         console.log(` - Owner: ${item.owner_email} (ID: ${item.owner_id})`);
49         console.log(` - Created: ${new Date(item.created_at).toLocaleString()}`);
50         console.log(` -----`);
51       });
52     }
53     console.log(` Total Items: ${items.rows.length}\\n`);
54
55     // Show table structures
56     console.log(`\\n DATABASE STATISTICS:`);
57     console.log(`-----`);
58     console.log(` - Total Users: ${users.rows.length}`);
59     console.log(` - Total Tasks: ${items.rows.length}`);
60
61     if (users.rows.length > 0) {
62       const avgItems = (items.rows.length / users.rows.length).toFixed(2);
63       console.log(` - Average Tasks per User: ${avgItems}`);
64     }
65
66     console.log(`\\n=====\\n`);
67
68     process.exit(0);
69   } catch (error) {
70     console.error(` Error viewing database: ${error.message}`);
71     process.exit(1);
72   }
73 }
```

Ejecución:



```
Terminal
You, 21 hours ago | author (You)
$ node server/viewDatabase.js
Total Users: 2
Total Tasks: 4
Average Tasks per User: 2.00
-----
```

| ID | Title                                   | Description | Owner                         | Created                 |
|----|---|-------------|-------------------------------|-------------------------|
| 1  | Test item 1                             | (none)      | disguisedog@gmail.com (ID: 2) | 2023-07-26, 18:00:18 PM |
| 2  | Test item 2                             | (none)      | disguisedog@gmail.com (ID: 2) | 2023-07-26, 18:00:42 PM |
| 3  | Me, No ChatGPT                          | (none)      | disguisedog@gmail.com (ID: 2) | 2023-07-26, 18:00:42 PM |
| 4  | Dear PM                                 | (none)      | disguisedog@gmail.com (ID: 2) | 2023-07-26, 18:00:42 PM |
| 5  | Que no se acuerda de que fue un ChatGPT | (none)      | disguisedog@gmail.com (ID: 2) | 2023-07-26, 18:00:42 PM |

```
File Browser
EVENDBP
  |- ACTS-FULLSTACK
  |  |- act1
  |  |- act2
  |  |- config.js
  |  |- createDb.js
  |  |- db.js
  |  |- schema.sql
  |  |- setup.js
  |  |- server.js
  |  |- viewDatabase.js
  |
  |- env
  |- lib
  |- utils
  |- client
  |  |- assets
  |  |- auth.css
  |  |- auth.js
  |  |- index.html
  |  |- login.html
  |  |- package-lock.json
  |  |- package.json
  |  |- style.css
  |
  |- server
  |  |- middleware
  |  |- routes
  |  |- config.js
  |  |- db.js
  |  |- schema.sql
  |  |- setup.js
  |  |- server.js
  |  |- viewDatabase.js
```

Despues para crear la DB, con su puerto indicado, el host (en este caso local):

The screenshot shows a terminal window with several tabs at the top: '.env', 'createDb.js X', 'db.js', 'schema.sql', 'setup.js', and 'setupRenderDb.js'. The current tab is 'createDb.js'. The code in the terminal is as follows:

```
You, yesterday | 1 author (You)
1 const { Client } = require('pg');      You, yesterday * feat: implement user authentication with login
2
3 async function createDatabase(){
4   const client = new Client({
5     host: 'localhost',
6     port: 5432,
7     user: 'diegovillarreal',
8     database: 'postgres'
9   });
10
11   try {
12     await client.connect();
13
14     // Check if database exists
15     const result = await client.query(
16       "SELECT 1 FROM pg_database WHERE datname='adpr_db'"
17     );
18
19     if (result.rows.length === 0) {
20       await client.query('CREATE DATABASE adpr_db');
21       console.log('✓ Database "adpr_db" created successfully!');
22     } else {
23       console.log('✗ Database "adpr_db" already exists');
24     }
25
26     await client.end();
27   } catch (error) {
28     console.error('✗ Error:', error.message);
29     process.exit(1);
30   }
31 }
32
33 createDatabase();
34
```

The output of the command shows the creation of the database:

```
✓ Database "adpr_db" created successfully!
```

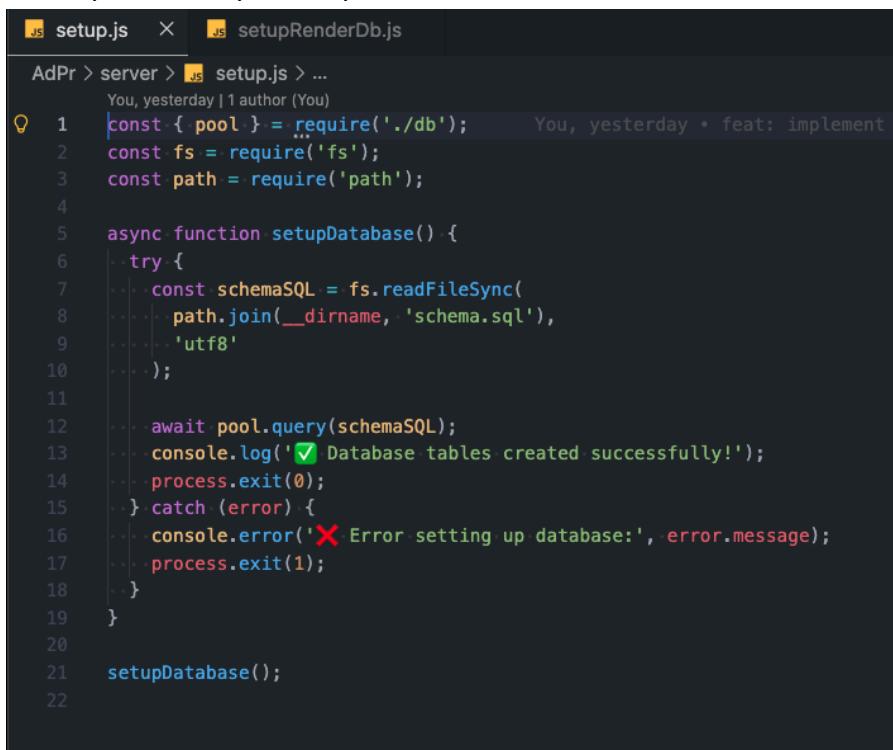
Despues la base de datos como tal:

```
AdPr > server > db.js > poolConfig > connectionString
You, 20 hours ago | 1 author (You)
1 const { Pool } = require('pg');
2 const { config } = require('./config');
3
4 // Use DATABASE_URL if available (production), otherwise use individual settings (local)
5 const poolConfig = config.databaseUrl
6 ... ?
7 ... connectionString: config.databaseUrl, You, 20 hours ago • Fix: Support DATABASE_URL fo
8 ... ssl: { rejectUnauthorized: false }
9 ...
10 ...
11 ...
12 const pool = new Pool(poolConfig);
13
14 async function query(text, params) {
15   return pool.query(text, params);
16 }
17
18 module.exports = { pool, query };
19
```

El codigo SQL que se uso:

```
schema.sql X setup.js setupRenderDb.js
AdPr > server > schema.sql
You, 2 days ago | 1 author (You)
1 CREATE TABLE IF NOT EXISTS users (
2   id SERIAL PRIMARY KEY,
3   email TEXT UNIQUE NOT NULL,
4   password_hash TEXT NOT NULL,
5   created_at TIMESTAMP DEFAULT NOW()
6 );
7
8 CREATE TABLE IF NOT EXISTS items (
9   id SERIAL PRIMARY KEY,
10  title TEXT NOT NULL,
11  description TEXT,
12  owner_id INTEGER NOT NULL REFERENCES users(id) ON DELETE CASCADE,
13  created_at TIMESTAMP DEFAULT NOW()
14 );
15
```

El setup de la DB para su perfecto funcionamiento:



```
AdPr > server > setup.js > ...
You, yesterday | 1 author (You)
Q 1 const { pool } = require('../db');      You, yesterday * feat: implement
2 const fs = require('fs');
3 const path = require('path');

4
5 async function setupDatabase() {
6   try {
7     const schemaSQL = fs.readFileSync(
8       path.join(__dirname, 'schema.sql'),
9       'utf8'
10    );
11
12     await pool.query(schemaSQL);
13     console.log('✅ Database tables created successfully!');
14     process.exit(0);
15   } catch (error) {
16     console.error('❌ Error setting up database:', error.message);
17     process.exit(1);
18   }
19 }
20
21 setupDatabase();
22
```

El sistema de autenticación de JWT, para poder poner tareas, eliminarlas, o editarlas, ocupas iniciar sesion:

```
AdPr > auth.js > API_URL
You, 20 hours ago | 1 author (You)
1 // API Configuration
2 // Automatically use the right API based on where the app is running
3 const API_URL = window.location.hostname === 'localhost' || window.location.hostname === '127.0.0.1'
4 ? 'http://localhost:4000/api' // Local development
5 : 'https://acts-fullstack.onrender.com/api'; // Production
6 You, 20 hours ago * Update API URL to use Re
7
8 // DOM Elements
9 const loginForm = document.getElementById('loginForm');
10 const registerForm = document.getElementById('registerForm');
11 const loginFormElement = document.getElementById('loginFormElement');
12 const registerFormElement = document.getElementById('registerFormElement');
13 const showRegisterLink = document.getElementById('showRegister');
14 const showLoginLink = document.getElementById('showLogin');
15 const loadingSpinner = document.getElementById('loadingSpinner');
16
17 // Toggle between login and register forms
18 showRegisterLink.addEventListener('click', (e) => {
19   e.preventDefault();
20   loginForm.classList.add('hidden');
21   registerForm.classList.remove('hidden');
22   clearErrors();
23 });
24
25 showLoginLink.addEventListener('click', (e) => {
26   e.preventDefault();
27   registerForm.classList.add('hidden');
28   loginForm.classList.remove('hidden');
29   clearErrors();
30 });
31
32 // Password visibility toggle
33 const setupPasswordToggle = (toggleId, inputId) => {
34   const toggle = document.getElementById(toggleId);
35   const input = document.getElementById(inputId);
36
37   if (toggle && input) {
38     toggle.addEventListener('click', () => {
39       const type = input.type === 'password' ? 'text' : 'password';
40       input.type = type;
41       toggle.classList.toggle('fa-eye');
42       toggle.classList.toggle('fa-eye-slash');
43     });
44   }
45
46 setupPasswordToggle('toggleLoginPassword', 'loginPassword');
47 setupPasswordToggle('toggleRegisterPassword', 'registerPassword');
48 setupPasswordToggle('toggleConfirmPassword', 'confirmPassword');
49
50 // Clear error messages
51 function clearErrors() {
52   document.getElementById('loginError').classList.remove('active');
53   document.getElementById('registerError').classList.remove('active');
54 }
55
56 // Show error message
57 function showError(elementId, message) {
58   const errorElement = document.getElementById(elementId);
59   errorElement.textContent = message;
60   errorElement.classList.add('active');
61 }
```

```
62 // Show/hide loading spinner
63 function setLoading(isLoading) {
64     if (isLoading) {
65         loginForm.classList.add('hidden');
66         registerForm.classList.add('hidden');
67         loadingSpinner.classList.remove('hidden');
68     } else {
69         loadingSpinner.classList.add('hidden');
70         if (loginForm.classList.contains('hidden')) {
71             registerForm.classList.remove('hidden');
72         } else {
73             loginForm.classList.remove('hidden');
74         }
75     }
76 }
77
78 // Validate email format
79 function isValidEmail(email) {
80     const emailRegex = /^[^@\s]+@[^\s@]+\.\w+$/;
81     return emailRegex.test(email);
82 }
83
84
85 // Login Form Submit
86 loginFormElement.addEventListener('submit', async (e) => {
87     e.preventDefault();
88     clearErrors();
89
90     const email = document.getElementById('loginEmail').value.trim();
91     const password = document.getElementById('loginPassword').value;
92
93     // Validation
94     if (!email || !password) {
95         showError('loginError', 'Please fill in all fields');
96         return;
97     }
98
99     if (!isValidEmail(email)) {
100         showError('loginError', 'Please enter a valid email address');
101         return;
102     }
103
104     // Disable button and show loading
105     const loginBtn = document.getElementById('loginBtn');
106     loginBtn.disabled = true;
107     setLoading(true);
108
109     try {
110         const response = await fetch(`${API_URL}/auth/login`, {
111             method: 'POST',
112             headers: {
113                 'Content-Type': 'application/json'
114             },
115             body: JSON.stringify({ email, password })
116         });
117
118         const data = await response.json();
119
120         if (!response.ok) {
121             throw new Error(data.error || 'Login failed');
```

```

100
101     try {
102       const response = await fetch(`${API_URL}/auth/login`, {
103         method: 'POST',
104         headers: {
105           'Content-Type': 'application/json'
106         },
107         body: JSON.stringify({ email, password })
108       });
109
110       const data = await response.json();
111
112       if (!response.ok) {
113         throw new Error(data.error || 'Login failed');
114       }
115
116       // Store token and email in localStorage
117       localStorage.setItem('token', data.token);
118       localStorage.setItem('userEmail', email);
119
120       // Redirect to main page
121       window.location.href = 'index.html';
122
123     } catch (error) {
124       setLoading(false);
125       loginBtn.disabled = false;
126       showError('loginError', error.message || 'Failed to login. Please try again.');
127     }
128   );
129
130
131 // Register Form Submit
132 registerFormElement.addEventListener('submit', async (e) => {
133   e.preventDefault();
134   clearErrors();
135
136   const email = document.getElementById('registerEmail').value.trim();
137   const password = document.getElementById('registerPassword').value;
138   const confirmPassword = document.getElementById('confirmPassword').value;
139
140   // Validation
141   if (!email || !password || !confirmPassword) {
142     showError('registerError', 'Please fill in all fields');
143     return;
144   }
145
146   if (!isValidEmail(email)) {
147     showError('registerError', 'Please enter a valid email address');
148     return;
149   }
150
151   if (password.length < 6) {
152     showError('registerError', 'Password must be at least 6 characters long');
153     return;
154   }
155
156   if (password !== confirmPassword) {
157     showError('registerError', 'Passwords do not match');
158     return;
159   }
160
161
162   // Disable button and show loading
163   const registerBtn = document.getElementById('registerBtn');
164   registerBtn.disabled = true;
165   setLoading(true);
166
167
168   try {
169     const response = await fetch(`${API_URL}/auth/register`, {
170       method: 'POST',
171       headers: {
172         'Content-Type': 'application/json'
173       },
174       body: JSON.stringify({ email, password })
175     });
176
177     const data = await response.json();
178
179     if (!response.ok) {
180       throw new Error(data.error || 'Registration failed');
181     }
182
183     // Auto-login after registration
184     const loginResponse = await fetch(`${API_URL}/auth/login`, {
185       method: 'POST',
186       headers: {
187         'Content-Type': 'application/json'
188       },
189       body: JSON.stringify({ email, password })
190     });
191
192     const loginData = await loginResponse.json();
193
194     if (!loginResponse.ok) {
195       throw new Error('Registration successful! Please login.');
196     }
197   }
198
199
200
201

```

```
196 // If registration successful
197 if (!loginResponse.ok) {
198   throw new Error('Registration successful! Please login.');
199 }
200
201 // Store token and email in localStorage
202 localStorage.setItem('token', loginData.token);
203 localStorage.setItem('userEmail', email);
204
205 // Redirect to main page
206 window.location.href = 'index.html';
207
208 } catch (error) {
209   setLoading(false);
210   registerBtn.disabled = false;
211   showError('registerError', error.message || 'Failed to register. Please try again.');
212 }
213 });
214
215 // Check if user is already logged in
216 window.addEventListener('DOMContentLoaded', () => {
217   const token = localStorage.getItem('token');
218   if (token) {
219     // Redirect to main page if already logged in
220     window.location.href = 'index.html';
221   }
222 });
223
224 });
225
```

Se uso un middleware para poder manejar errores y validaciones / autenticaciones

```
auth.js  ×  errorHandler.js
AdPr > server > middleware > auth.js > ...
You, 2 days ago | 1 author (You)
1 const jwt = require('jsonwebtoken');
2 const { config } = require('../config');
3
4 function requireAuth(req, res, next) {
5   const header = req.headers.authorization || '';
6   const token = header.startsWith('Bearer ') ? header.slice(7) : null;
7
8   if (!token) {
9     return res.status(401).json({ error: 'Missing or invalid authorization token.' });
10  }
11
12  try {
13    const payload = jwt.verify(token, config.jwtSecret);
14    req.user = payload;
15    return next();
16  } catch (error) {
17    return res.status(401).json({ error: 'Invalid or expired token.' });
18  }
19}
20
21 module.exports = { requireAuth };
22
```

```
AdPr > server > middleware > errorHandler.js > notFound
You, 2 days ago | 1 author (You)
1 function notFound(req, res, next) { You, 2 days ago * feat: initiali
2   const error = new Error(`Route not found: ${req.originalUrl}`);
3   error.status = 404;
4   next(error);
5 }
6
7 function errorHandler(err, req, res, next) {
8   const status = err.status || 500;
9   const message = err.message || 'Internal server error.';
10
11  if (status >= 500) {
12    console.error('[ERROR]', err);
13  }
14
15  res.status(status).json({ error: message });
16 }
17
18 function asyncHandler(fn) {
19  return function asyncWrapper(req, res, next) {
20    Promise.resolve(fn(req, res, next)).catch(next);
21  };
22}
23
24 module.exports = { notFound, errorHandler, asyncHandler };
25
```

Ya que el uso de inteligencia artificial fue permitido para ciertas cosas, se utilizo para poder facilitar procesos como lo pueden ser: conexion de base de datos, conectar el backend en la nube mediante una aplicacion, configurar la base de datos, y el MiddleWare

# C)Frontend

Primero es el index.html

```
AdPr > index.html > html > body > div#editModal.modal > div.modal-content > div.modal-body > div.modal-actions > button#cancelEditBtn.btn
You, yesterday | 1 author (You)
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Task Manager | Stay Organized</title>
7      <link rel="stylesheet" href="style.css">
8      <link rel="preconnect" href="https://fonts.googleapis.com">
9      <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
10     <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">
11     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
12 </head>
13 <body>
14     <div class="container">
15         <!-- Header -->
16         <header>
17             <div class="header-content">
18                 <div class="header-left">
19                     <h1>
20                         <i class="fas fa-tasks"></i>
21                         Task Manager
22                     </h1>
23                     <p class="subtitle">Stay organized and productive</p>
24                 </div>
25                 <div class="header-right">
26                     <div id="userInfo" class="user-info"></div>
27                     <button id="logoutBtn" class="btn-logout">
28                         <i class="fas fa-sign-out-alt"></i>
29                         Logout
30                     </button>
31                 </div>
32             </div>
33         </header>
34         <!-- Add Task Section -->
35         <div class="add-task-section">
36             <div class="input-group">
37
38             </div>
39             <!-- Tasks Section -->
40             <div class="tasks-section">
41                 <div class="tasks-header">
42                     <h2>
43                         <i class="fas fa-list-check"></i>
44                         My Tasks
45                     </h2>
46                     <div class="stats">
47                         <span id="totalTasks">Total: 0</span>
48                         <span id="completedTasks">Completed: 0</span>
49                     </div>
50
51                 <!-- Filters -->
52                 <div class="filters">
53                     <button class="filter-btn active" data-filter="all">
54                         <i class="fas fa-list"></i> All
55                     </button>
56                     <button class="filter-btn" data-filter="pending">
57                         <i class="fas fa-clock"></i> Pending
58                     </button>
59                     <button class="filter-btn" data-filter="completed">
60                         <i class="fas fa-check-circle"></i> Completed
61                     </button>
62
63                 <!-- Empty State -->
64                 <div id="emptyState" class="empty-state">
65                     <i class="fas fa-clipboard-list"></i>
66                     <h3>No tasks yet</h3>
67                     <p>Add your first task to get started!</p>
68                 </div>
69
70             </div>
71
72         <!-- Footer -->
73         <div class="footer">
74             <img alt="Task Manager logo" />
75             <div>
76                 <h2>Task Manager</h2>
77                 <p>Your personal productivity tool</p>
78             </div>
79             <div>
80                 <a href="#">Home</a>
81                 <a href="#">About</a>
82                 <a href="#">Contact</a>
83             </div>
84         </div>
85
86     </div>
87
88 </body>
89
90 </html>
```

```

AdPr > index.html > HTML > body > div#editModal.modal > div.modal-content > div.modal-body > div.modal-actions > button#cancelEditBtn
1   <html lang="en">
2     <body>
3       <div class="container">
4         <div class="tasks-section">
5           <!-- Empty State -->
6           <div id="emptyState" class="empty-state">
7             <i class="fas fa-clipboard-list"></i>
8             <h3>No tasks yet</h3>
9             <p>Add your first task to get started!</p>
10            </div>
11
12           <!-- Tasks List -->
13           <ul id="tasksList" class="tasks-list"></ul>
14         </div>
15
16           <!-- Footer -->
17           <footer>
18             <p>&copy; <span id="currentYear">2026</span> Task Manager. All rights reserved.</p>
19           </footer>
20         </div>
21
22           <!-- Edit Modal -->
23           <div id="editModal" class="modal">
24             <div class="modal-content">
25               <div class="modal-header">
26                 <h2>
27                   <i class="fas fa-edit"></i>
28                   Edit Task
29                 </h2>
30                 <button class="close-modal">&times;</button>
31               </div>
32               <div class="modal-body">
33                 <input type="text" id="editTaskInput" placeholder="Task name..." autocomplete="off">
34                 <div class="modal-actions">
35                   <button id="saveEditBtn" class="btn-primary">
36                     <i class="fas fa-save"></i>
37                     Save Changes
38                   </button>
39                 </div>
40               </div>
41             </div>
42           </div>
43
44           <script src="app.js"></script>
45         </body>
46       </html>
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119

```

Tiene su respectivo Style.css (El cual estara en el repo de github debido a su alto contenido)  
Y su respectivo archivo JS (El cual tambien estara en el repo de github debido a su alto contenido)

## Desoques viene el login.html:

```
AdPr > login.html ...  
You, yesterday | author (You)  
1 <!DOCTYPE html> You, yesterday + feat: implement user authentication with login ~  
2 <html lang="en">  
3 <head>  
4   <meta charset="UTF-8">  
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6   <title>Login Task Manager</title>  
7   <link rel="stylesheet" href="auth.css">  
8   <link rel="preconnect" href="https://fonts.googleapis.com">  
9   <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">  
10  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">  
11 </head>  
12 <body>  
13   <div class="auth-container">  
14     <div class="auth-card">  
15       <div class="auth-header">  
16         <div class="logo">  
17           <i class="fas fa-tasks"></i>  
18         </div>  
19         <h1>Task Manager</h1>  
20         <p>Stay organized and productive</p>  
21       </div>  
22     </div>  
23  
24   <!-- Login Form -->  
25   <div class="form-container" id="loginForm">  
26     <h2>Welcome Back!</h2>  
27     <p>Sign in to continue</p>  
28  
29     <form id="loginFormElement">  
30       <div class="input-field">  
31         <i class="fas fa-envelope"></i>  
32         <input type="email" id="loginEmail" placeholder="Email address" required autocomplete="email">  
33       </div>  
34  
35       <div class="input-field">  
36         <i class="fas fa-lock"></i>  
37         <input type="password" id="loginPassword" placeholder="Password" required autocomplete="current-password">  
38         <i class="fas fa-eye toggle-password" id="toggleLoginPassword"></i>  
39     </div>  
40  
41     <div class="error-message" id="loginError"></div>  
42  
43     <button type="submit" class="btn-primary" id="loginBtn">  
44       <span>Sign In</span>  
45       <i class="fas fa-arrow-right"></i>  
46     </button>  
47   </form>  
48  
49   <div class="form-footer">  
50     <p>Don't have an account? <a href="#" id="showRegister">Sign up</a></p>  
51   </div>  
52  
53   <!-- Register Form -->  
54   <div class="form-container hidden" id="registerForm">  
55     <h2>Create Account</h2>  
56     <p>Join us to get started</p>  
57  
58     <form id="registerFormElement">  
59       <div class="input-field">  
60         <i class="fas fa-envelope"></i>  
61         <input type="email" id="registerEmail" placeholder="Email address" required autocomplete="email">  
62       </div>  
63     </form>
```

```
AdPr > login.html ...  
You, yesterday | author (You)  
1 <html lang="en">  
2 <body>  
3   <div class="auth-container">  
4     <div class="auth-card">  
5       <div class="form-container" id="loginForm">  
6         <i class="fas fa-envelope"></i>  
7         <input type="email" id="loginEmail" placeholder="Email address" required autocomplete="email">  
8       </div>  
9  
10      <div class="input-field">  
11        <i class="fas fa-lock"></i>  
12        <input type="password" id="loginPassword" placeholder="Password" required autocomplete="current-password">  
13        <i class="fas fa-eye toggle-password" id="toggleLoginPassword"></i>  
14      </div>  
15  
16      <div class="error-message" id="loginError"></div>  
17  
18      <button type="submit" class="btn-primary" id="loginBtn">  
19        <span>Sign In</span>  
20        <i class="fas fa-arrow-right"></i>  
21      </button>  
22  
23    </div>  
24  
25    <div class="form-footer">  
26      <p>Don't have an account? <a href="#" id="showRegister">Sign up</a></p>  
27    </div>  
28  
29    <!-- Register Form -->  
30    <div class="form-container hidden" id="registerForm">  
31      <h2>Create Account</h2>  
32      <p>Join us to get started</p>  
33  
34      <form id="registerFormElement">  
35        <div class="input-field">  
36          <i class="fas fa-envelope"></i>  
37          <input type="email" id="registerEmail" placeholder="Email address" required autocomplete="email">  
38        </div>  
39  
40        <div class="input-field">  
41          <i class="fas fa-lock"></i>  
42          <input type="password" id="registerPassword" placeholder="Password" required autocomplete="new-password">  
43          <i class="fas fa-eye toggle-password" id="toggleRegisterPassword"></i>  
44        </div>  
45  
46        <div class="input-field">  
47          <i class="fas fa-lock"></i>  
48          <input type="password" id="confirmPassword" placeholder="Confirm password" required autocomplete="new-password">  
49          <i class="fas fa-eye toggle-password" id="toggleConfirmPassword"></i>  
50        </div>  
51  
52        <div class="error-message" id="registerError"></div>  
53  
54        <button type="submit" class="btn-primary" id="registerBtn">  
55          <span>Create Account</span>  
56          <i class="fas fa-user-plus"></i>  
57        </button>  
58  
59      </form>  
60  
61      <div class="form-footer">  
62        <p>Already have an account? <a href="#" id="showLogin">Sign in</a></p>  
63      </div>  
64  
65    </div>  
66  
67    <!-- Loading Spinner -->  
68    <div class="loading-spinner hidden" id="loadingSpinner">  
69      <i class="fa fa-circle-notch fa-spin" style="font-size: 2em; color: #ccc;"></i>  
70      <p>Please wait...</p>  
71    </div>  
72  
73  </div>  
74  
75  <div class="background-shapes">  
76    <div class="shape shape-1"></div>
```

```
AdPr > login.html ...  
You, yesterday | author (You)  
1 <!DOCTYPE html> You, yesterday + feat: implement user authentication with login ~  
2 <html lang="en">  
3 <head>  
4   <meta charset="UTF-8">  
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6   <title>Login Task Manager</title>  
7   <link rel="stylesheet" href="auth.css">  
8   <link rel="preconnect" href="https://fonts.googleapis.com">  
9   <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">  
10  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">  
11 </head>  
12 <body>  
13   <div class="auth-container">  
14     <div class="auth-card">  
15       <div class="auth-header">  
16         <div class="logo">  
17           <i class="fas fa-tasks"></i>  
18         </div>  
19         <h1>Task Manager</h1>  
20         <p>Stay organized and productive</p>  
21       </div>  
22     </div>  
23  
24   <!-- Login Form -->  
25   <div class="form-container" id="loginForm">  
26     <h2>Welcome Back!</h2>  
27     <p>Sign in to continue</p>  
28  
29     <form id="loginFormElement">  
30       <div class="input-field">  
31         <i class="fas fa-envelope"></i>  
32         <input type="email" id="loginEmail" placeholder="Email address" required autocomplete="email">  
33       </div>  
34  
35       <div class="input-field">  
36         <i class="fas fa-lock"></i>  
37         <input type="password" id="loginPassword" placeholder="Password" required autocomplete="current-password">  
38         <i class="fas fa-eye toggle-password" id="toggleLoginPassword"></i>  
39     </div>  
40  
41     <div class="error-message" id="loginError"></div>  
42  
43     <button type="submit" class="btn-primary" id="loginBtn">  
44       <span>Sign In</span>  
45       <i class="fas fa-arrow-right"></i>  
46     </button>  
47  
48   </div>  
49  
50   <!-- Register Form -->  
51   <div class="form-container hidden" id="registerForm">  
52     <h2>Create Account</h2>  
53     <p>Join us to get started</p>  
54  
55     <form id="registerFormElement">  
56       <div class="input-field">  
57         <i class="fas fa-envelope"></i>  
58         <input type="email" id="registerEmail" placeholder="Email address" required autocomplete="email">  
59       </div>  
60  
61       <div class="input-field">  
62         <i class="fas fa-lock"></i>  
63         <input type="password" id="registerPassword" placeholder="Password" required autocomplete="new-password">  
64         <i class="fas fa-eye toggle-password" id="toggleRegisterPassword"></i>  
65       </div>  
66  
67       <div class="input-field">  
68         <i class="fas fa-lock"></i>  
69         <input type="password" id="confirmPassword" placeholder="Confirm password" required autocomplete="new-password">  
70         <i class="fas fa-eye toggle-password" id="toggleConfirmPassword"></i>  
71       </div>  
72  
73       <div class="error-message" id="registerError"></div>  
74  
75       <button type="submit" class="btn-primary" id="registerBtn">  
76         <span>Create Account</span>  
77         <i class="fas fa-user-plus"></i>  
78       </button>  
79  
80     </form>  
81  
82     <div class="form-footer">  
83       <p>Already have an account? <a href="#" id="showLogin">Sign in</a></p>  
84     </div>  
85  
86   </div>  
87  
88   <!-- Loading Spinner -->  
89   <div class="loading-spinner hidden" id="loadingSpinner">  
90     <i class="fa fa-circle-notch fa-spin" style="font-size: 2em; color: #ccc;"></i>  
91     <p>Please wait...</p>  
92   </div>  
93  
94 </div>
```

The screenshot shows a code editor interface with two tabs: 'login.html' and 'auth.css'. The 'login.html' tab is active, displaying the following HTML code:

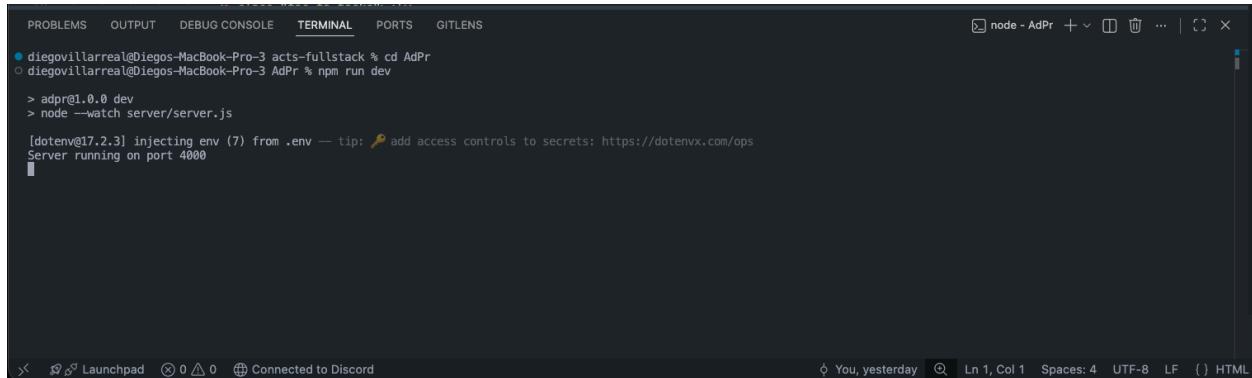
```
login.html
AdPr > login.html <-- auth.css
AdPr > login.html <!--
<html lang="en">
<head>
</head>
<body>
    <div class="auth-container">
        <div class="auth-card">
            <!-- Loading spinner -->
            <div class="loading-spinner hidden" id="loadingSpinner">
                <i class="fas fa-circle-notch fa-spin" style="font-size: 2em;"></i>
                <p>Please wait...</p>
            </div>
        </div>
        <div class="background-shapes">
            <div class="shape shape-1"></div>
            <div class="shape shape-2"></div>
            <div class="shape shape-3"></div>
        </div>
    </div>
    <script src="auth.js"></script>
</body>
</html>
```

Este HTML tiene su respectivo Style.css (el cual estara en el repo debido a su alto contenido)  
Tambien cuenta con su [Auth.js](#) (el cual ya se menciono anteriormente y se encuentra en el repo debdo a su alto contenido)

# D)Despliegue Preliminar

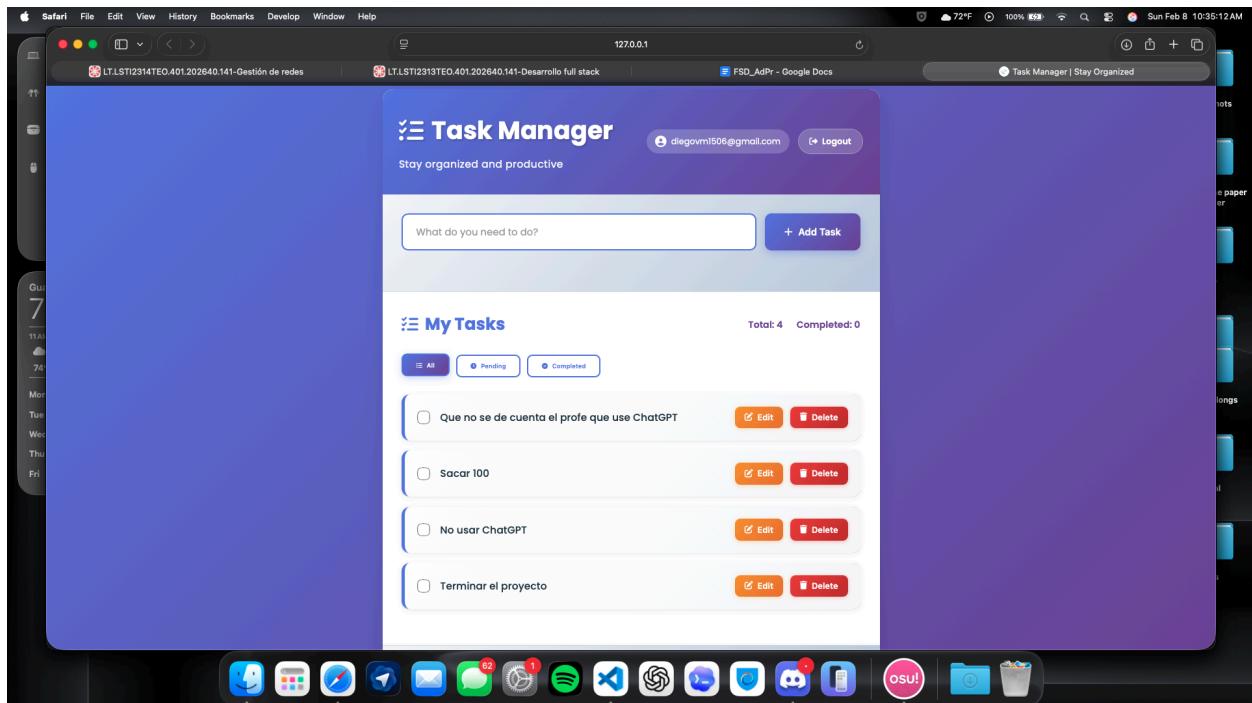
Ejecucion:

Para poder activar el server se necesita poner en la terminal “npm run dev”

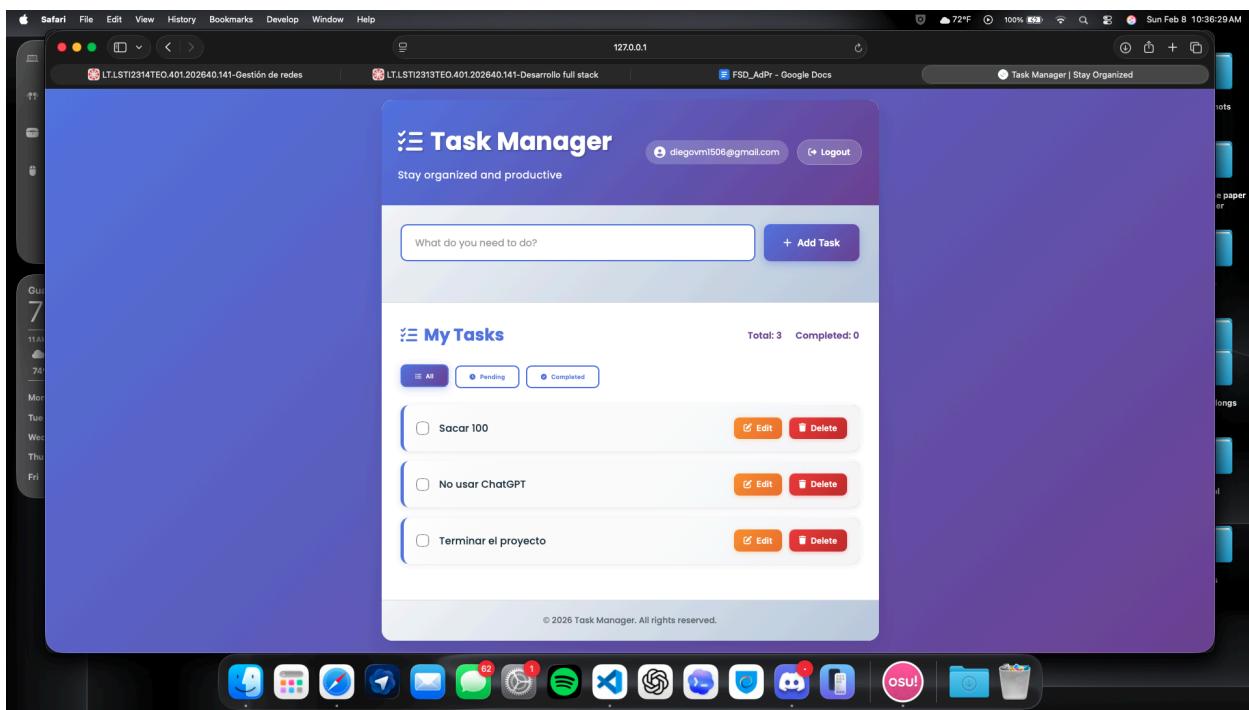
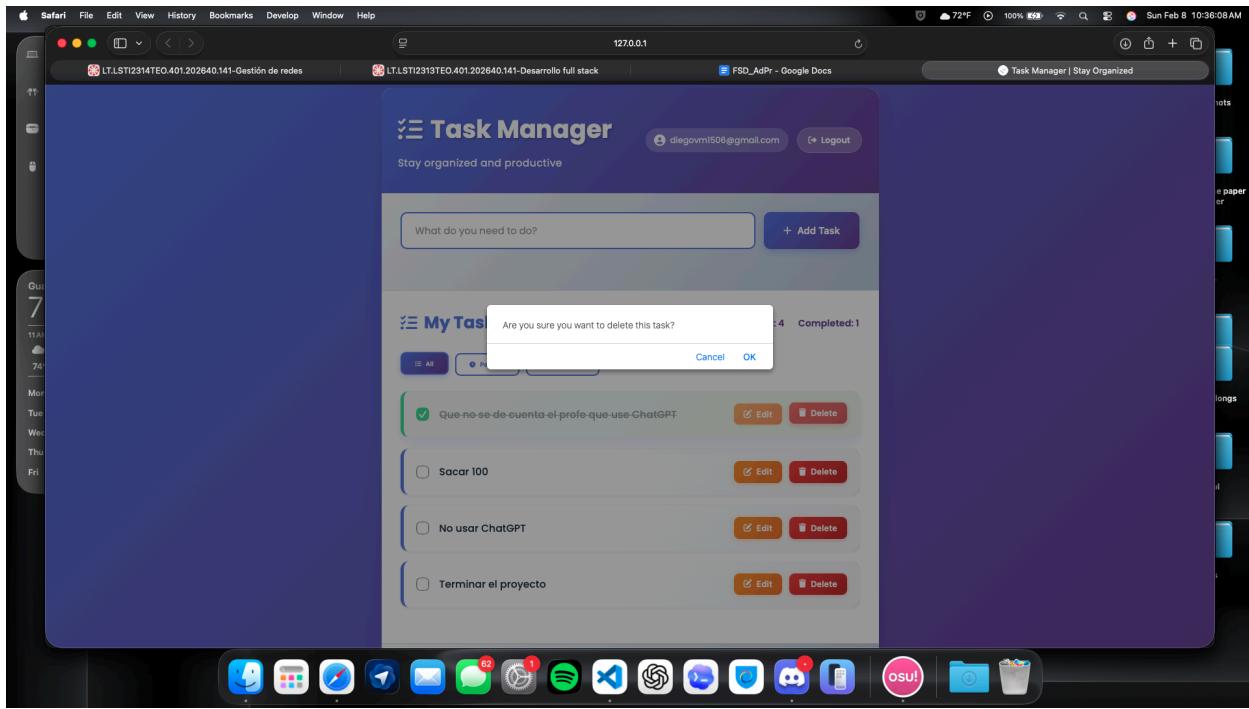


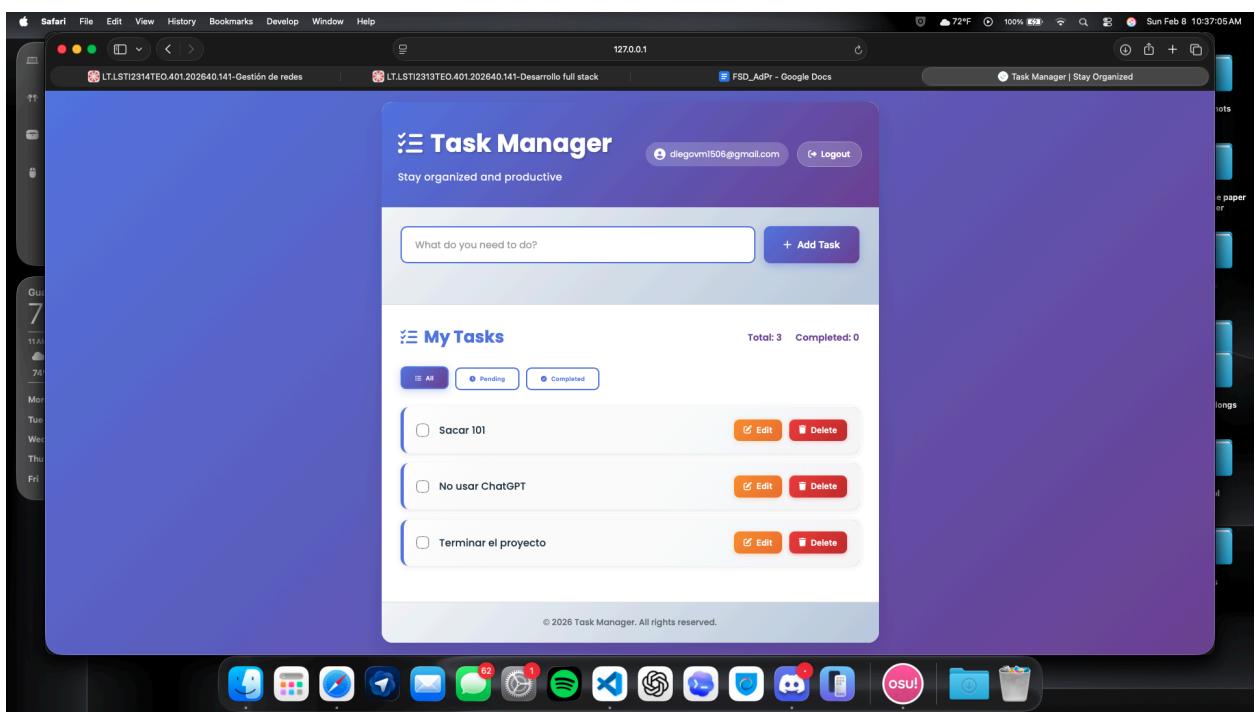
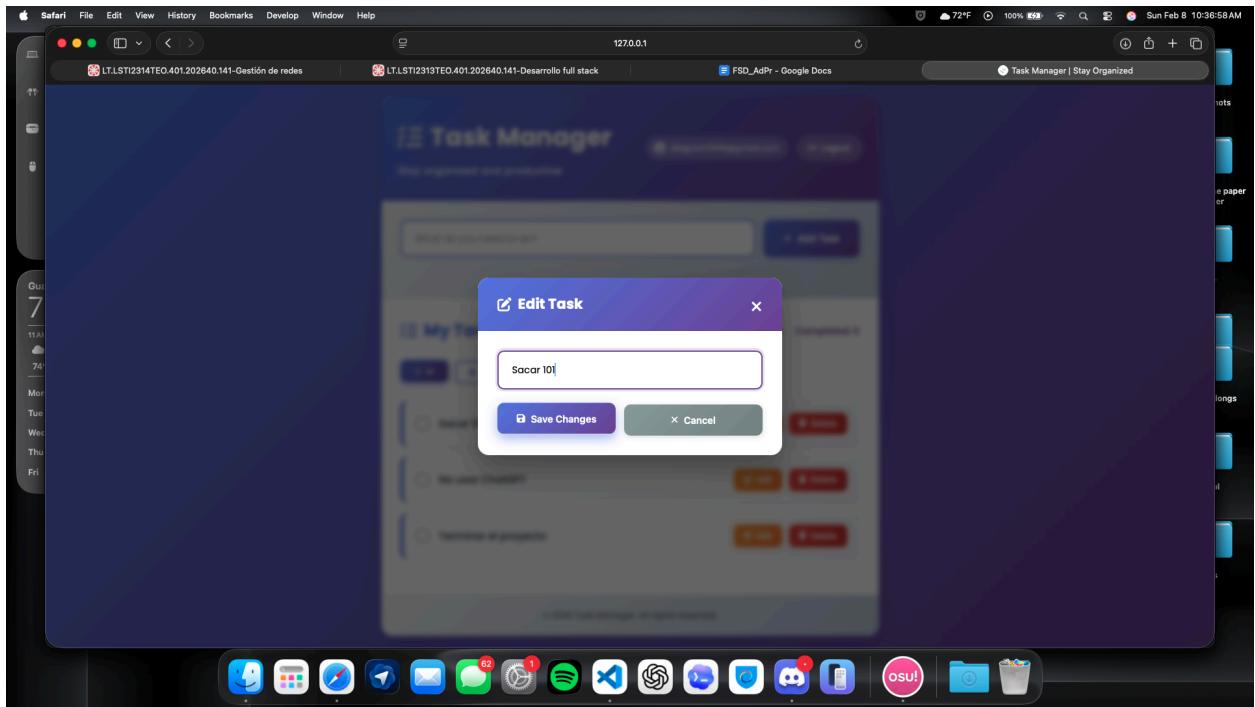
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
diegovillarreal@Diegos-MacBook-Pro-3:~/acts-fullstack % cd AdPr
diegovillarreal@Diegos-MacBook-Pro-3:~/AdPr % npm run dev
> adpr@1.0.0 dev
> node --watch server/server.js
[dotenv@17.2.3] injecting env (7) from .env -- tip: 🎁 add access controls to secrets: https://dotenvx.com/ops
Server running on port 4000
```

Ya con eso, se puede inicializar en el entorno local:



Ahí se pueden observar varias tareas que puse, las cuales se pueden editar, eliminar, o agregar, tambien obsérvese que se puede cerrar sesión





The screenshot shows a Mac OS X desktop environment with a dark-themed Visual Studio Code (VS Code) window open. The terminal tab in VS Code displays the output of a Node.js script named `viewDatabase.js`. The script logs the contents of a database, showing three items with IDs 2, 3, and 4, each with a title, description, owner, and creation timestamp. It also provides database statistics: total users (2), total tasks (3), and average tasks per user (1.50). The VS Code interface includes an Explorer sidebar on the right showing the project structure for "ACTS-FULLSTACK", which includes files like `config.js`, `createDb.js`, `db.js`, `schema.sql`, `server.js`, `setup.js`, `setupRenderDb.js`, and `viewDatabase.js`. The status bar at the bottom indicates the current date and time as "Sun Feb 8 10:37:47 AM".

```
AdPr > server > viewDatabase.js ...
You 22 hours ago | author (You)
1 // View Database Contents      You, 22 hours ago * feat: update API URL configuration for local an...
2 // Run this with: node server/viewDatabase.js
3
4 const { query } = require('./db');
5
6 async function viewDatabase() {
7   console.log('===== DATABASE CONTENTS =====');
8   console.log(`\n${'-'}. ${'-'}`);
```

diegovillarreal@Diegos-MacBook-Pro-3 AdPr % node server/viewDatabase.js

| ID | Title                | Description | Owner                        | Created                        |
|----|----------------------|-------------|------------------------------|--------------------------------|
| 2  | Terminar el proyecto | (none)      | diegovm150@gmail.com (ID: 2) | Created: 2/7/2026, 12:25:48 PM |
| 3  | No usar ChatGPT      | (none)      | diegovm150@gmail.com (ID: 2) | Created: 2/7/2026, 12:25:56 PM |
| 4  | Sacar foto           | (none)      | diegovm150@gmail.com (ID: 2) | Created: 2/7/2026, 12:26:02 PM |

Total Items: 3

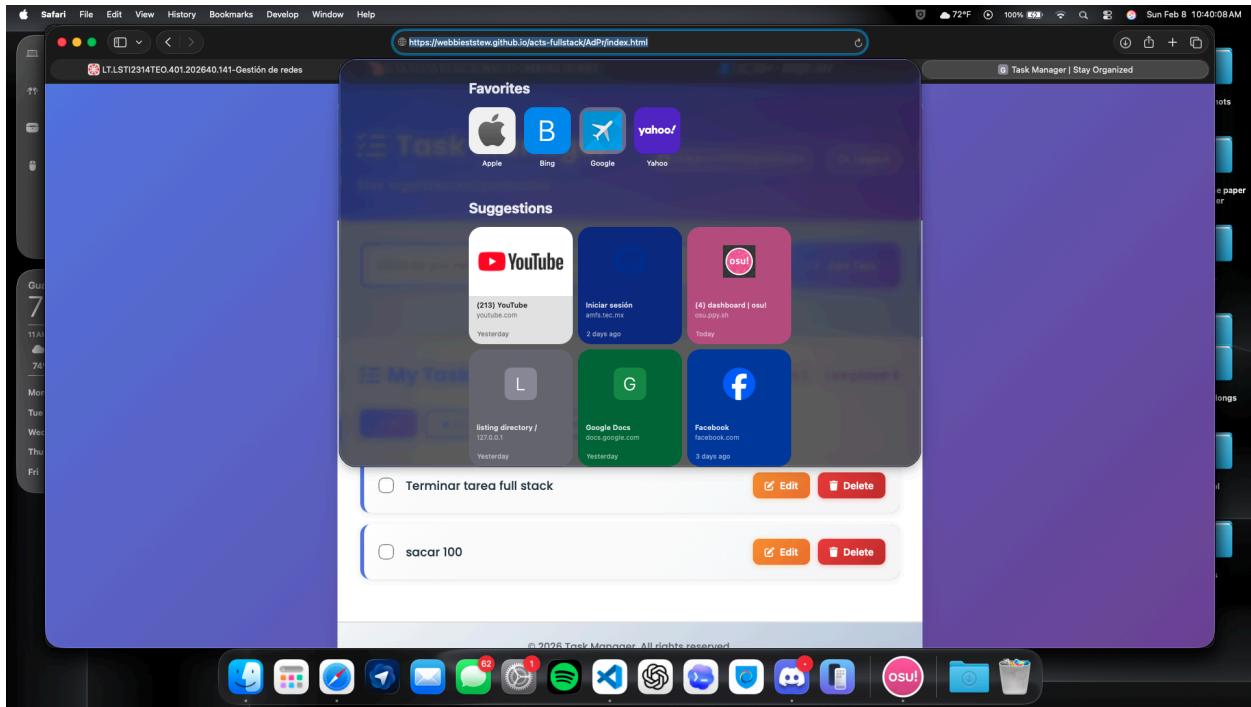
DATABASE STATISTICS:

Total Users: 2  
Total Tasks: 3  
Average Tasks per User: 1.50

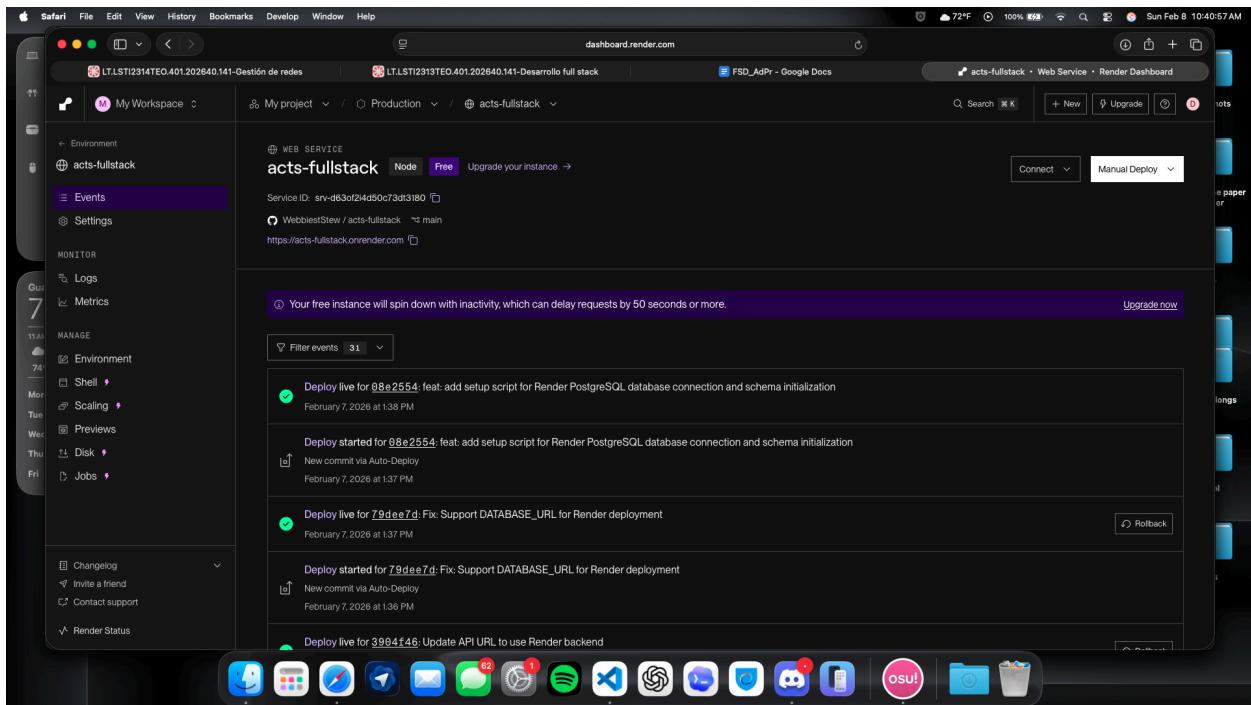
```
diegovillarreal@Diegos-MacBook-Pro-3 AdPr %
└─ Launchpad ┌─ Connected to Discord
```

Tambien se observa en la DB que cambiaron las tasks

Tambien en la nube se puede visualizar la pagina con el link proporcionado al principio del documento:



Se hizo una base de datos, y el backend fue subido a la plataforma Render:



The screenshot shows the Render Dashboard interface. On the left, a sidebar lists 'My Workspace' (selected), 'Environment' (selected), 'Logs', 'Metrics', 'Shell', 'Scaling', 'Previews', 'Disk', 'Jobs', 'Changelog', 'Invite a friend', 'Contact support', and 'Render Status'. The main content area is titled 'Environment' and contains sections for 'Environment Variables' and 'Secret Files'. Under 'Environment Variables', there are two entries: 'DATABASE\_URL' with a value of '.....' and 'JWT\_SECRET' with a value of '.....'. A 'Create environment group' button is located at the top right of this section. Below it is a 'Secret Files' section with the note 'Store plaintext files containing secret data (such as a .env file or a private key)'. At the bottom of the dashboard are various application icons.

The screenshot shows the Render Dashboard interface. On the left, a sidebar lists 'My Workspace' (selected), 'taskmanager-db' (selected), 'Info', 'Logs', 'Metrics' (selected), 'Apps', and 'Recovery'. The main content area is titled 'Metrics' and displays 'Table Sizes' and 'Index Sizes' tables. The 'Table Sizes' table shows two rows:

| DATABASE            | SCHEMA | TABLE | SIZE  |
|---------------------|--------|-------|-------|
| taskmanager_db_ck0w | public | users | 16 KB |
| taskmanager_db_ck0w | public | items | 16 KB |

The 'Index Sizes' table shows three rows:

| DATABASE            | SCHEMA | TABLE | INDEX           | SIZE  |
|---------------------|--------|-------|-----------------|-------|
| taskmanager_db_ck0w | public | users | users_pkey      | 16 KB |
| taskmanager_db_ck0w | public | users | users_email_key | 16 KB |
| taskmanager_db_ck0w | public | items | items_pkey      | 16 KB |

At the bottom of the dashboard are various application icons.

Por seguridad de los demás usuarios, no se puede ver ni los correos, ni las contraseñas que se ponen

# Conclusion

Hacer una pagina, puede parecer sencillo, pero el trabajo que conlleva hacerlo, puede llegar a tornarse tedioso, en este caso, en la parte donde mas hubo dificultad fue al conectar la base de datos, pero se logro.