

# SQL INJECTION NO CÓDIGO

The screenshot shows the Visual Studio Code interface with an Angular project. The Explorer on the left shows the file structure, including the `login` component. The main editor displays the `login.component.ts` file, which contains a `loginModel` and a `login` method. The `login` method has a vulnerability where it checks if the email contains any of the words in `listaPalavras` using `includes`, which is susceptible to SQL injection.

```
src > app > views > login > TS login.component.ts > ...
22 usuarioLogado = ""
23
24 onSubmit() {
25   console.log(this.loginModel)
26
27   let erroEncontrado = 0;
28
29   const listaPalavras: string[] = ["select ", "from ", "drop ", "or ", "having ", "group ", "insert ", "exec ", "\"", "'", "--", "#",
30
31   listaPalavras.forEach(palavra => {
32     console.log("palavra atual:", palavra)
33
34     if(this.loginModel.email.toLowerCase().includes(palavra)) {
35       console.log("Palavra encontrada:", palavra)
36       this.mensagem = "Dados inválidos: " + palavra;
37
38       erroEncontrado = 1;
39     }
40
41   })
```

The bottom panel shows the output of the Angular CLI, indicating that the application bundle generation is complete and the application is running on `localhost:4200`.

Initial Chunk Files	Names	Raw Size
vendor.js	vendor	2.34 MB
polyfills.js	polyfills	299.95 kB
styles.css, styles.js	styles	173.22 kB
main.js	main	71.45 kB
runtime.js	runtime	6.51 kB
Initial Total		2.87 MB

Build at: 2022-04-25T15:00:30.374Z - Hash: 25ab2d96111a7f99 - Time: 11388ms

\*\* Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ \*\*

✓ Compiled successfully.

# SQL INJECTION NO SITE

The screenshot shows a web browser window with the address bar at `localhost:4200/login`. The page displays the 'GAME MANIA' logo and a login form with the following fields and buttons:

- Email:
- Senha:
- Fazer Login

The DevTools console is open, showing the following log entries:

- 1 Issue: 1
- palavra atual: -- login.component.ts:32
- palavra atual: # login.component.ts:32
- palavra atual: \* login.component.ts:32
- palavra atual: ; login.component.ts:32
- Failed to load resource: the server responded with a status of 400 (Bad Request) :3000/login:1
- Failed to load resource: the server responded with a status of 400 (Bad Request) :3000/login:1
- ERROR > HttpErrorResponse core.mjs:6485
- User login.component.ts:25
- palavra atual: select login.component.ts:32
- palavra atual: from login.component.ts:32
- palavra atual: drop login.component.ts:32
- palavra atual: or login.component.ts:32
- palavra atual: having login.component.ts:32
- palavra atual: group login.component.ts:32
- palavra atual: insert login.component.ts:32
- palavra atual: exec login.component.ts:32
- palavra atual: " login.component.ts:32
- palavra atual: ' login.component.ts:32
- palavra atual: -- login.component.ts:32
- palavra atual: # login.component.ts:32
- palavra atual: \* login.component.ts:32
- palavra atual: ; login.component.ts:32
- Failed to load resource: the server responded with a status of 400 (Bad Request) :3000/login:1
- Failed to load resource: the server responded with a status of 400 (Bad Request) :3000/login:1
- ERROR > HttpErrorResponse core.mjs:6485

The Windows taskbar at the bottom shows the system clock at 12:08 on 25/04/2022, with a temperature of 26°C and the location 'Parc ensolarado'.

# POST NO POSTMAN

The screenshot displays the Postman application interface. At the top, the title bar shows 'Postman' with standard window controls. The menu bar includes 'File', 'Edit', 'View', and 'Help'. The main toolbar features 'Home', 'Workspaces', 'Reports', 'Explore', a search bar, and buttons for 'Sign In' and 'Create Account'. A yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'.

The interface is divided into several sections. On the left, a sidebar contains icons for Overview, Environment, Collections, Recent, and History. The main workspace shows a POST request to 'http://localhost:3000/users'. The request body is a JSON object:

```
1 {
2   "email": "renata.95.silva@email.com",
3   "password": "batata123",
4   "nome": "Renata",
5   "id": 6
6 }
7
```

Below the request, the 'Body' tab is selected, showing the response in 'Pretty' format:

```
1 {
2   "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFnbnCI6InJlbmF0YS45NS5zaWx2YUJlbWFnbnCI6IjE2NTA5MDM4ODUsImV4cCI6MTY1MDkwNzQ4NSwic3ViIjoiaWoiIj9.
3   zTCVxdVl7pXQqU00tDc_GmTaDGQ25_qPt1kNzIm9DK",
4   "user": {
5     "email": "renata.95.silva@email.com",
6     "nome": "Renata",
7     "id": 6
8   }
9 }
```

The status bar at the bottom shows 'Status: 201 Created', 'Time: 204 ms', and 'Size: 707 B'. The system tray at the very bottom displays the date '25/04/2022' and time '13:25'.

# PUT NO POSTMAN

The screenshot displays the Postman application interface. At the top, there's a menu bar with 'File', 'Edit', 'View', and 'Help'. Below it, a navigation bar includes 'Home', 'Workspaces', 'Reports', and 'Explore', along with a search bar and buttons for 'Sign In' and 'Create Account'. A yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main interface shows a PUT request to 'http://localhost:3000/users/6'. The 'Body' tab is selected, showing a JSON payload: 

```
{  "email": "renata.95.silva@email.com",  "password": "batata123",  "nome": "Renata",  "id": 6}
```

. The 'Send' button is visible. Below the request, the 'Body' tab of the response is shown, displaying the same JSON payload. The status bar at the bottom indicates 'Status: 200 OK', 'Time: 228 ms', and 'Size: 547 B'. The system tray at the very bottom shows the date '25/04/2022' and time '13:26'.

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

Overview PUT http://localhost:3000/users/6 No Environment

http://localhost:3000/users/6 Save

PUT http://localhost:3000/users/6 Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "email": "renata.95.silva@email.com",
3   "password": "batata123",
4   "nome": "Renata",
5   "id": 6
6 }
```

Body Cookies Headers (13) Test Results

Status: 200 OK Time: 228 ms Size: 547 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "email": "renata.95.silva@email.com",
3   "password": "$2a$10$AXG0mPqRIu9pFyI4VHZXj.SBFi06zUCBUTYF5RgtwMnXu73Pp.Gxi",
4   "nome": "Renata",
5   "id": 6
6 }
```

Find and Replace Console

Runner Trash

26°C Parc ensolarado

13:26 25/04/2022

# DELETE NO POSTMAN

The screenshot displays the Postman application interface. At the top, the menu bar includes File, Edit, View, and Help. Below it, the main navigation bar shows Home, Workspaces, Reports, and Explore, along with a search bar and buttons for Sign In and Create Account. A yellow banner indicates "Working locally in Scratch Pad. Switch to a Workspace".

The main workspace shows a DELETE request to `http://localhost:3000/users/6`. The request is configured with the following details:

- Method:** DELETE
- URL:** `http://localhost:3000/users/6`
- Body:**

```
{  "email": "renata.95.silva@email.com",  "password": "batata123",  "nome": "Renata",  "id": 6}
```
- Headers:** 8 headers are listed.
- Params:** none
- Authorization:** none
- Pre-request Script:** none
- Tests:** none
- Settings:** none

The response is displayed in the bottom panel, showing a status of 200 OK, a time of 51 ms, and a size of 395 B. The response body is formatted as JSON.

The bottom status bar shows the temperature (26°C), location (Parc ensolarado), and the system clock (13:30, 25/04/2022).

# LOGIN NO SITE

