



Cibersegurança e Defesa Cibernética

SKILLZ

São Paulo

Novembro de 2023



Integrantes

SKILLZ

Débora Souza Matos – RA: 22023906

Eduardo Donini – RA: 22023219

João Fernando de Lima – RA: 22023774

Lucas André dos Santos – RA: 22022350

Mateus Nogueira Leal – RA: 22023872

São Paulo

Novembro de 2023

Introdução

Este documento tem como objetivo evidenciar o método de criptografia utilizada no desenvolvimento da Plataforma SKILLZ. Para incrementar a Codificação Base64, foi utilizada a linguagem de programação JavaScript, por meio do método “btoa ()”.

Sobre o método

O método “btoa ()” do JavaScript é uma função essencial para o desenvolvimento web, pois permite converter dados binários em texto ASCII de forma segura. Essa conversão é necessária em diversas situações, como na comunicação com APIs ou na transmissão de informações pela internet.

Vale ressaltar a importância da técnica de codificação Base64, pois se trata de uma técnica usada em situações nas quais dados binários, como imagens, precisam ser enviados como texto em protocolos web, como HTTP, portanto podem ser convertidos usando esta técnica. Essa técnica é uma maneira muito usada para garantir a segurança na transmissão de dados binários em ambientes de texto.

Porém, neste contexto é crucial entender que ele aceita apenas caracteres dentro do conjunto ASCII. Se caracteres fora desse intervalo são fornecidos, o método “btoa ()” lança erros, destacando a necessidade de considerar os limites durante a codificação.

Lógica do método

O método “btoa ()” opera seguindo uma lógica específica. Ele divide os dados binários em blocos de 3 bytes (24 bits) e converte cada bloco em 4 grupos de 6 bits. Estes grupos de 6 bits são então mapeados para caracteres ASCII usando uma tabela base64 predefinida.

Este método possui uma principal limitação que diz respeito a sua incapacidade de lidar com caracteres que estão fora do conjunto ASCII. Desta maneira, ao identificar caracteres que ferem esse conjunto o método retorna erros e exige que a entrada de dados seja validada com cuidado para evitar essas interrupções ao longo do processo de codificação.

Sua sensibilidade ao intervalo ASCII demanda atenção cuidadosa ao processar dados para evitar erros. Compreender a lógica e as limitações do “btoa ()” é essencial para utilizá-lo eficazmente, garantindo a integridade e segurança dos dados em aplicações web modernas.

Exemplo de aplicação do algoritmo:

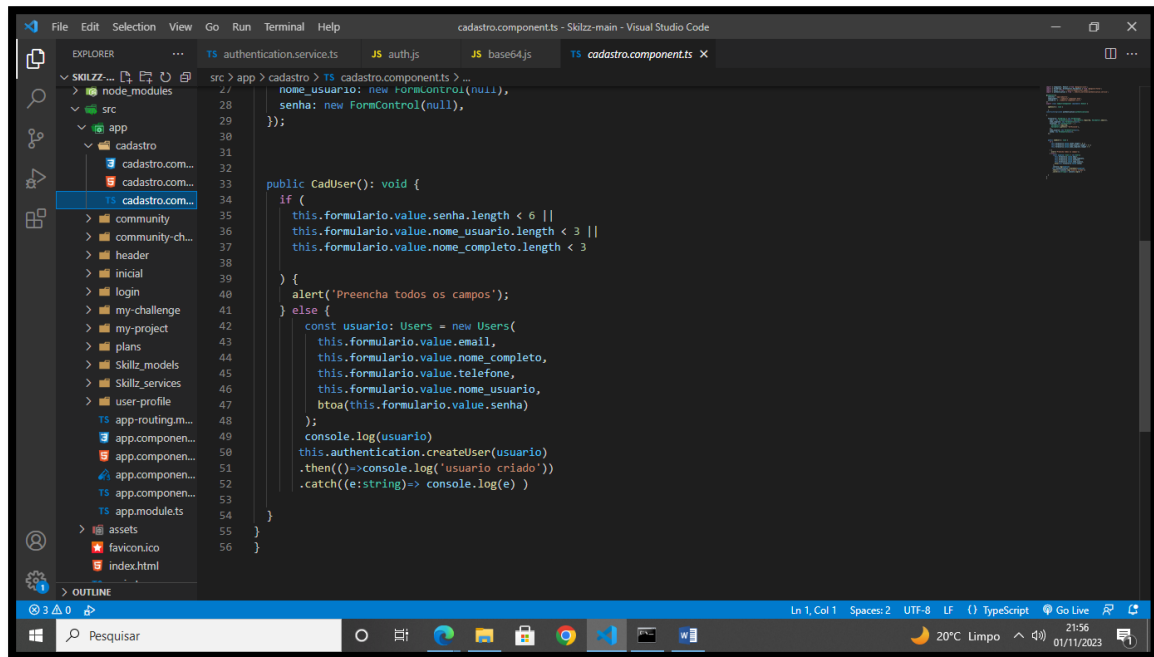
```
cyber > JS base64.js > ...
1
2 function stringToBase64(str) {
3     var encoded = '';
4     for (var i = 0; i < str.length; i += 3) {
5         var chunk = str.charCodeAt(i) << 16 | str.charCodeAt(i + 1) << 8 | str.charCodeAt(i + 2);
6         encoded += String.fromCharCode((chunk >> 18) & 0x3F, (chunk >> 12) & 0x3F, (chunk >> 6) & 0x3F, chunk & 0x3F)
7         .replace(/[\u0000-\u00FF]/g, '');
8     }
9     var padding = str.length % 3;
10    if (padding > 0) {
11        encoded = encoded.slice(0, padding - 3);
12        encoded += '=='.substring(padding);
13    }
14    return encoded;
15 }
16 var senha_original = "123456";
17 var senha_codificada = stringToBase64(senha_original);
18 console.log(senha_codificada)
19 }
```

O método “btoa ()” em JavaScript é uma ferramenta de grande valor e poder para codificação Base64, sendo a principal razão pela escolha desse algoritmo, juntamente com a sua simplicidade de aplicação e dinamismo, contribuindo positivamente para o objetivo e proposta da SKILLZ, agregando muito a parte de segurança e integridade da Plataforma.

Aplicação prática do algoritmo

Podemos ver a aplicação do algoritmo no desenvolvimento da Plataforma SKILLZ, em ambas as interfaces (Front End e Back End).

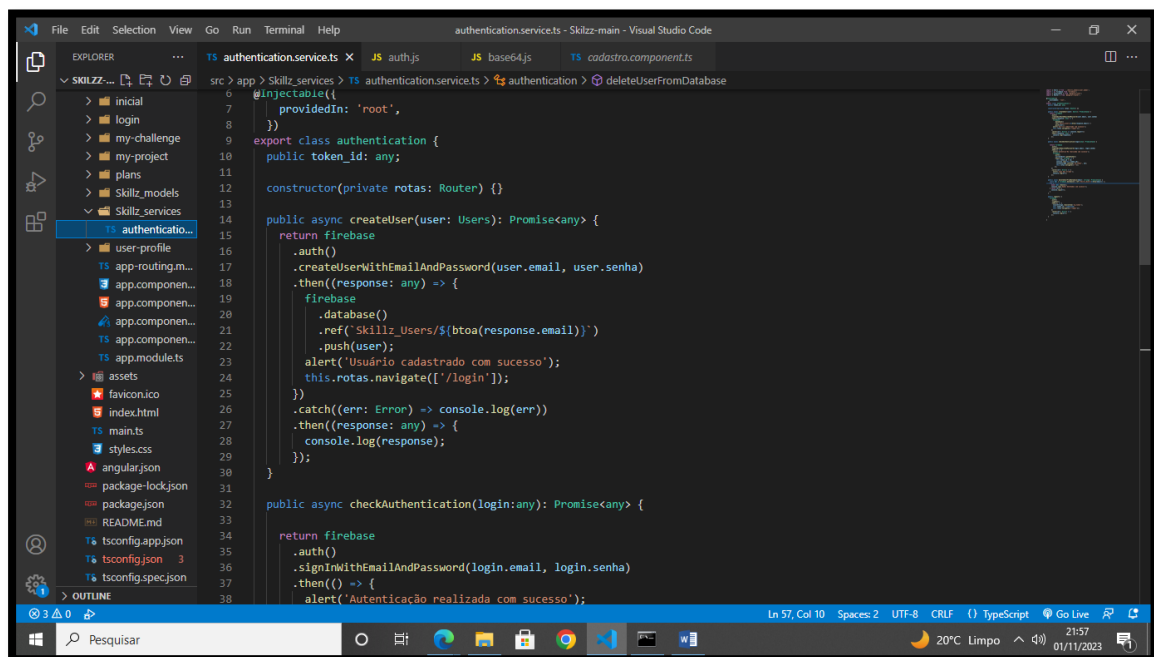
Função aplicada no código do:



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying a file tree for a project named 'SKILLZ-main'. The file 'cadastro.component.ts' is selected. The main editor displays the code for this component. The code includes a class 'CadastroComponent' with a 'cadastro' method. This method checks if the form is valid (email, password, and name fields are filled) and if the password is at least 6 characters long. If valid, it calls 'this.authentication.createUser' with the user details. If not valid, it shows an alert 'Preencha todos os campos'. The code also includes a 'public CadUser(): void' method that initializes the form controls.

```
src > app > cadastro > TS cadastro.component.ts > ...
27 nome_usuario: new FormControl(null),
28 senha: new FormControl(null),
29 });
30
31
32
33
34 public CadUser(): void {
35   if (
36     this.formulario.value.senha.length < 6 ||
37     this.formulario.value.nome_usuario.length < 3 ||
38     this.formulario.value.nome_completo.length < 3
39   ) {
40     alert('Preencha todos os campos');
41   } else {
42     const usuario: Users = new Users(
43       this.formulario.value.email,
44       this.formulario.value.nome_completo,
45       this.formulario.value.telefone,
46       this.formulario.value.nome_usuario,
47       btoa(this.formulario.value.senha)
48     );
49     console.log(usuario);
50     this.authentication.createUser(usuario)
51       .then(() => console.log('usuario criado'))
52       .catch((e:string) => console.log(e));
53   }
54 }
55
56 }
```

Figura 1 FRON-END



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying a file tree for a project named 'SKILLZ-main'. The file 'authentication.service.ts' is selected. The main editor displays the code for this service. The code includes a class 'AuthenticationService' with a 'createUser' method. This method calls 'this.firebase.auth().createUserWithEmailAndPassword' with the user email and password. It then calls 'this.firebase.database().ref().push()' to save the user details to the database. The code also includes a 'checkAuthentication' method that calls 'this.firebase.auth().signInWithEmailAndPassword' with the login email and password. The code also includes a 'constructor' that initializes the 'token_id' property.

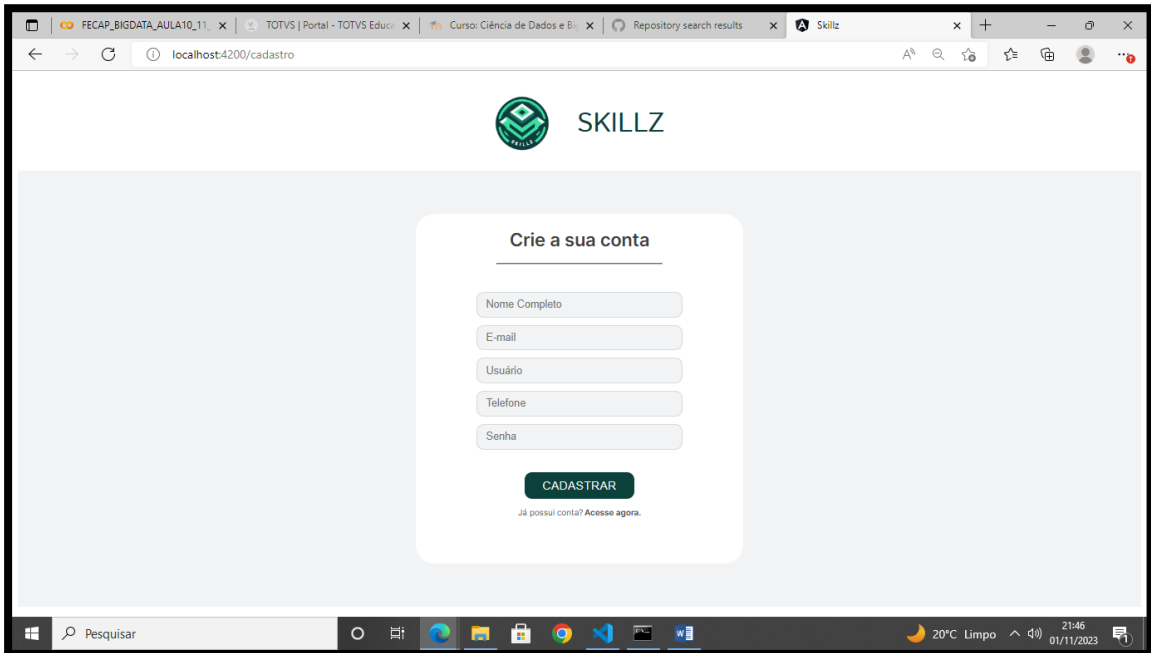
```
src > app > Skillz_services > TS authentication.service.ts > TS authentication > deleteUserFromDatabase
6 @Injectable({
7   providedIn: 'root',
8 })
9 export class authentication {
10   public token_id: any;
11
12   constructor(private rotas: Router) {}
13
14   public async createUser(user: Users): Promise<any> {
15     return firebase
16       .auth()
17       .createUserWithEmailAndPassword(user.email, user.senha)
18       .then((response: any) => {
19         firebase
20           .database()
21           .ref('Skillz_Users/${btoa(response.email)}')
22           .push(user);
23         alert('Usuário cadastrado com sucesso');
24         this.rotas.navigate(['/login']);
25       })
26       .catch((err: Error) => console.log(err))
27       .then((response: any) => {
28         console.log(response);
29       });
30   }
31
32   public async checkAuthentication(login:any): Promise<any> {
33     return firebase
34       .auth()
35       .signInWithEmailAndPassword(login.email, login.senha)
36       .then(() => {
37         alert('Autenticação realizada com sucesso');
38       });
39   }
40 }
```

Figura 2 BACK-END

Etapa de inclusão da criptografia

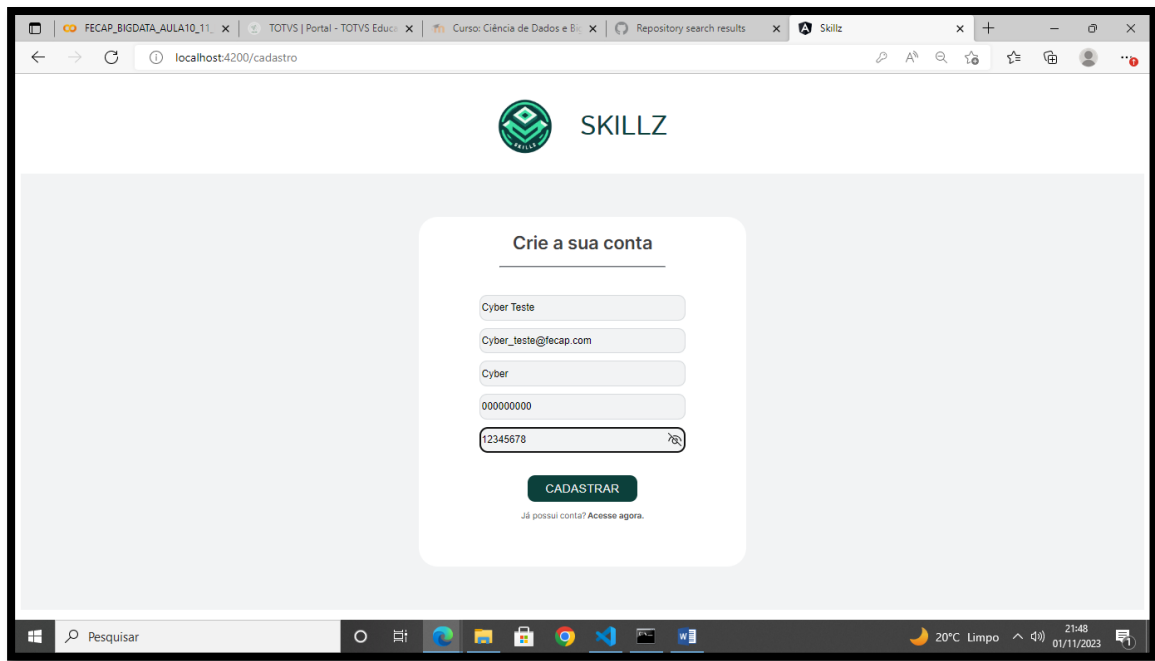
A criptografia foi incluída ao longo de todo o processo de cadastro dos usuários na plataforma da SKILLZ, passando pela inclusão dos dados necessários para realizar o cadastro completo e se estendendo até a etapa de salvamento desses usuários no banco de dados do Firebase (nota-se que já no bando de dados a senha do usuário é armazenada de maneira criptografada).

As etapas a seguir evidenciam o processo de criptografia:

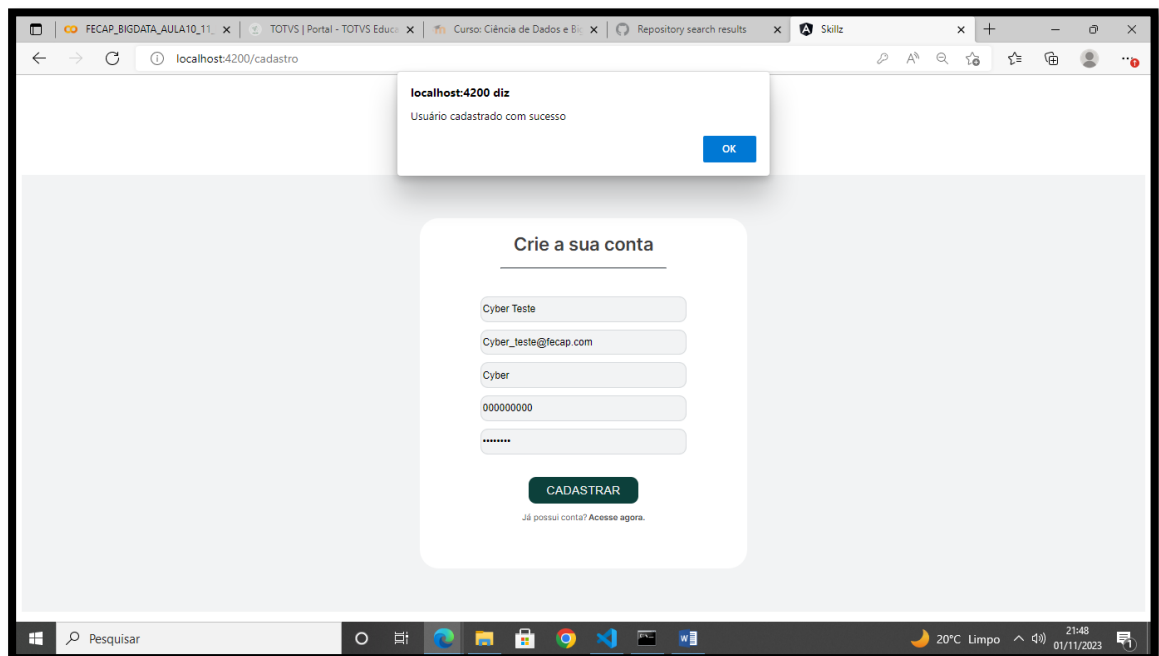


The screenshot shows a web browser window with the URL `localhost:4200/cadastro`. The page features the SKILLZ logo at the top center. Below the logo is a registration form titled "Crie a sua conta". The form contains five input fields: "Nome Completo", "E-mail", "Usuário", "Telefone", and "Senha". Below these fields is a green button labeled "CADASTRAR". At the bottom of the form, there is a link that says "Já possui conta? Acesse agora." The browser's taskbar at the bottom shows the Windows logo, a search bar, and several application icons. The system tray on the right indicates the temperature is 20°C, the location is Limpo, and the date and time are 01/11/2023 at 21:46.

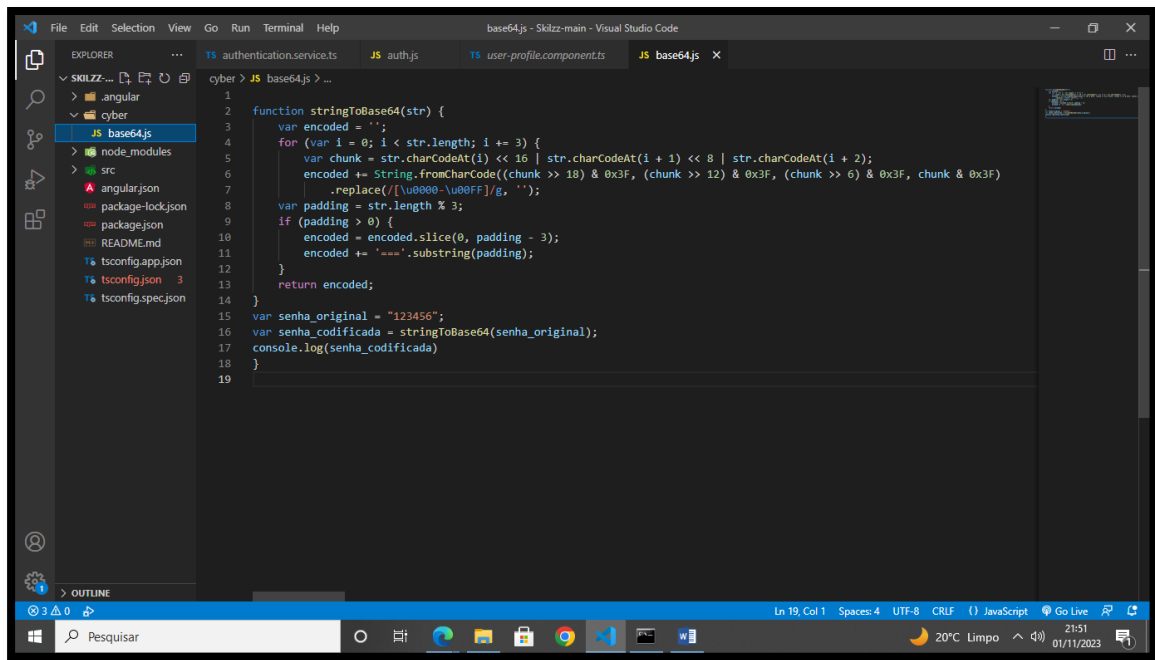
Etapa: 1 Cadastro de Usuários na Plataforma



Etapa: 2 Cadastro de Usuário na Plataforma

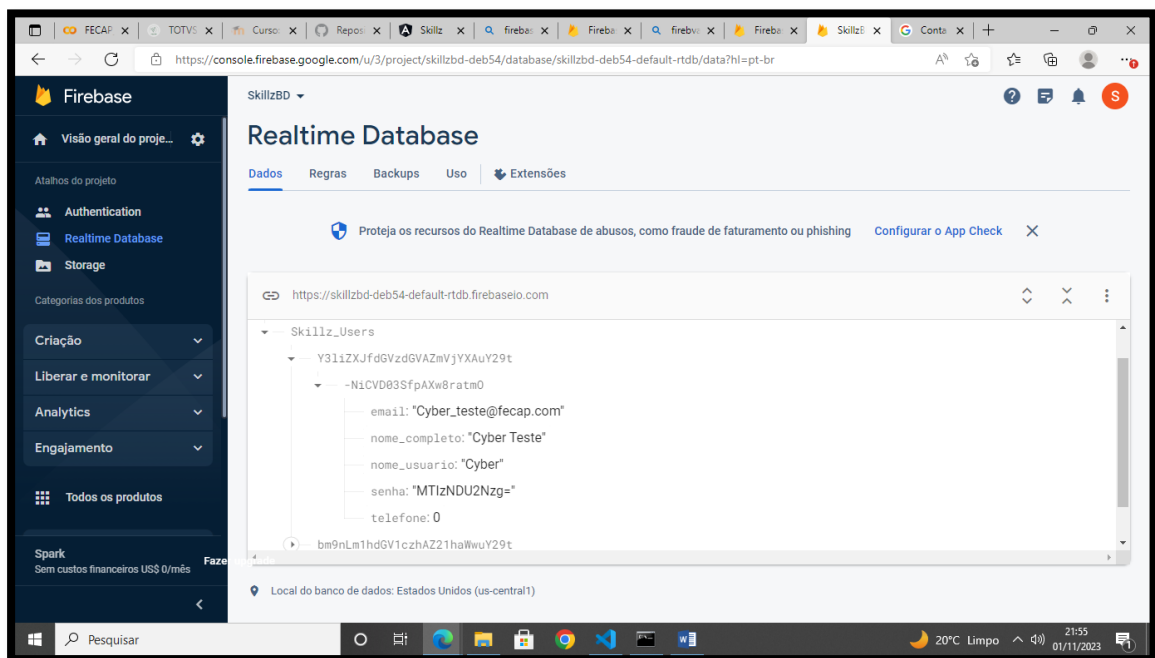


Etapa: 3 Cadastro de Usuário na Plataforma



```
1 function stringToBase64(str) {
2   var encoded = '';
3   for (var i = 0; i < str.length; i += 3) {
4     var chunk = str.charCodeAt(i) << 16 | str.charCodeAt(i + 1) << 8 | str.charCodeAt(i + 2);
5     encoded += String.fromCharCode((chunk >> 18) & 0x3F, (chunk >> 12) & 0x3F, (chunk >> 6) & 0x3F, chunk & 0x3F)
6     .replace(/[\u0000-\u00FF]/g, '');
7   }
8   var padding = str.length % 3;
9   if (padding > 0) {
10    encoded = encoded.slice(0, padding - 3);
11    encoded += '=='.substr(0, padding);
12  }
13  return encoded;
14 }
15 var senha_original = "123456";
16 var senha_codificada = stringToBase64(senha_original);
17 console.log(senha_codificada);
18 }
19
```

Etapa: 4 Aplicação do Algoritmo de Criptografia



Etapa: 5 Armazenamento dos Dados do Usuário