

Cashurrasco API

Visão Geral:

A API "Cashurrasco" é uma ferramenta desenvolvida como parte de um projeto acadêmico que visa simplificar a organização de eventos de churrasco. Com o objetivo de tornar a experiência de planejamento e realização de churrascos mais fácil e eficiente, esta API oferece uma série de recursos e funcionalidades essenciais.

Recursos Principais:

1. **Cálculo de Ingredientes:** A API permite calcular a quantidade necessária de carne, carvão, temperos, acompanhamentos e bebidas com base no número de convidados e nas preferências do organizador do churrasco.
2. **Estimativa de Custos:** Os usuários podem obter uma estimativa dos custos totais do churrasco, incluindo ingredientes, utensílios e outros gastos relacionados ao evento.
3. **Agendamento de Eventos:** Os usuários podem agendar a data e a hora do churrasco.
4. **Personalização:** Os organizadores podem personalizar o evento de acordo com suas preferências e necessidades específicas.
5. **Histórico de eventos:** O usuário pode obter registros de todos os eventos realizados.
6. **Conta do usuário:** O usuário pode se cadastrar, buscar dados da conta, alterar dados da conta e excluir a conta.

A API "Cashurrasco" foi projetada para ser fácil de usar e flexível, atendendo tanto a pequenos encontros informais quanto a grandes eventos. Com esta ferramenta, a organização de churrascos se torna uma tarefa mais organizada e eficaz, garantindo que os anfitriões e convidados desfrutem ao máximo de seus momentos de lazer e socialização.

Endpoints:

<https://cashurrasco-api.vercel.app/usuarios>

- **Método suportados:** GET, POST, PUT e DELETE
- **Parâmetros de consulta:** para usar os métodos GET, PUT, e DELETE é necessário passar o id do usuário para executar as respectivas operações de “Ler dados do usuário”, “Alterar dados do usuário” e “Excluir usuário”. exemplo de URL: <https://cashurrasco-api.vercel.app/usuarios/id>

- **Respostas da API:**

- GET:

```
// Exemplo de Resposta de sucesso:

// Status code: 200.
{
  msg: "usuário retornado com sucesso!",
  status: "success",
  usuario: { // Todos os dados do usuário }
}

// Exemplo das respostas de erro:

// Status code: 404.
{
  msg: "O Usuário não encontrado.",
  status: "error",
}
```

- POST

```
// Exemplo de Resposta de sucesso:

// Status code: 201.
{
  msg: "Usuário cadastrado com sucesso!",
  status: "success",
  dados: { // Dados do usuário que acabou de se cadastrar }
}
```

```
// Exemplo das respostas de erro:

// Status code: 400.
{
  msg: "Erro ao cadastrar usuário.",
  status: "error"
}
```

◦ PUT

```
// Exemplo de Resposta de sucesso:

// Status code: 200.
{
  msg: "Usuário editado com sucesso!",
  status: "success"
}

// Exemplo das respostas de erro:

// Status code: 400.
{
  msg: "Não foi possível editar o usuário.",
  status: "error"
}:

```

◦ DELETE

```
// Exemplo de Resposta de sucesso:

// Status code: 200.
{
  msg: "Usuário deletado com sucesso!",
  status: "success"
}

// Exemplo das respostas de erro:

// Status code: 400.
{
  msg: "Não foi possível deletar o usuário.",
  status: "error"
}:

```

- **Exemplos de código:**

- GET

```
async function buscarDadosUsuario(idUsuario) {

  try {

    const request = await fetch(`${url}/usuarios/${idUsuario}`, {
      method: "GET",
      headers: {
        "Content-Type": "application/json"
      },
    });

    const response = await request.json();

    console.log(response);

  } catch (error) {
    console.error(error);
  }

}
```

- POST

```
// Fazer cadastro
formCadastro.addEventListener('submit', async evento => {
  evento.preventDefault();

  // Preencher o objeto para cadastrar
  const usuario = {
    nome: document.querySelector('#nome').value,
    email: document.querySelector('#email').value,
    celular: document.querySelector('#celular').value,
    senha: document.querySelector('#senha').value
  }

  console.log(usuario);

  await cadastrarUsuario(usuario);

});
```

```
const url = "https://cashurrasco-api.vercel.app";

async function cadastrarUsuario(obj) {

  try {

    const request = await fetch(`${url}/usuarios`, {
      method: "POST",
      headers: {
        "Content-Type": "application/json"
      },
      body: JSON.stringify(obj)
    });

    const response = await request.json();

    console.log(response);

  } catch (error) {
    console.error(error);
  }

}
```

◦ PUT

```
// Função assíncrona para realizar a requisição
async function alterarUsuario(obj) {

  try {

    const request = await fetch(`${url}/usuarios/${id}`, {
      method: "PUT",
      headers: {
        "Content-Type": "application/json"
      },
      // Mandando o objeto com as alterações
      body: JSON.stringify(obj)
    });

    const response = await request.json();

    console.log(response);

  } catch (error) {
    console.error(error);
  }

}
```

```
}
```

- DELETE

```
async function apagarUsuario(id) {  
  
  try {  
  
    const request = await fetch(`${url}/usuarios/${id}`, {  
      method: "DELETE",  
      headers: {  
        "Content-Type": "application/json"  
      },  
    });  
  
    const response = await request.json();  
  
    console.log(response);  
  
  } catch (error) {  
    console.error(error);  
  }  
  
}
```

<https://cashurrasco-api.vercel.app/usuarios/login>

- **Métodos suportados:** POST.
- **Parâmetros de consulta:** Nenhum.
- **Resposta da API:**
 - POST

```
// Exemplo de Resposta de sucesso:  
  
// Status code: 200.  
{  
  msg: "Logado!",  
  status: "success",  
  dados: { // Todos os dados do usuário que acabou de logar }  
}
```

```
// Exemplo das respostas de erro:

// Status code: 404.
{
  msg: "Email ou senha inválidos...",
  status: "error"
}
```

- Exemplo de código:

```
formLogin.addEventListener('submit', async evento => {
  evento.preventDefault();

  // Preencher o objeto para login
  const usuario = {
    email: document.querySelector('#loginEmail').value,
    senha: document.querySelector('#loginSenha').value
  }

  await login(usuario);
});

// URL da API
const url = "https://cashurrasco-api.vercel.app";

// Função assíncrona para realizar a requisição
async function login(obj) {

  try {

    const request = await fetch(`${url}/usuarios/login`, {
      method: "POST",
      headers: {
        "Content-Type": "application/json"
      },
      body: JSON.stringify(obj)
    });

    const response = await request.json();

    console.log(response);

  } catch (error) {
    console.error(error);
  }
}
```

```
}
```

<https://cashurrasco-api.vercel.app/Evento>

- **Métodos suportados:** GET, POST, PUT e DELETE.
- **Parâmetros de consulta:** para usar os métodos GET, PUT, e DELETE é necessário passar o id do evento para executar as respectivas operações de “Ler dados do evento”, “Editar evento” e “Excluir evento”. exemplo de URL:
<https://cashurrasco-api.vercel.app/evento/id>

- **Respostas da API:**

- GET

```
// Exemplo de Resposta de sucesso:

// Status code: 200.
{
  msg: "Evento retornado com sucesso!",
  status: "success",
  dados: { // Todos os dados do evento }
}

// Exemplo das respostas de erro:

// Status code: 404.
{
  msg: "Nenhum evento encontrado.",
  status: "error"
}
```

- POST

```
// Exemplo de Resposta de sucesso:

// Status code: 201.
{
  msg: "Evento cadastrado com sucesso!",
  status: "success",
}
```



```
    dados: { // Todos os dados do usuário que acabou de logar }
  }

// Exemplo das respostas de erro:

// Status code: 400.
{
  msg: "Não foi possível cadastrar o evento.",
  status: "error"
}
```

■ PUT

```
// Exemplo de Resposta de sucesso:

// Status code: 200.
{
  msg: "Evento editado com sucesso!",
  status: "success",
  dados: { // Todos os dados editados do evento }
}

// Exemplo das respostas de erro:

// Status code: 400.
{
  msg: "Não foi possível editar o evento.",
  status: "error"
}
```

■ DELETE

```
// Exemplo de Resposta de sucesso:

// Status code: 200.
{
  msg: "Evento deletado com sucesso!",
  status: "success"
}

// Exemplo das respostas de erro:

// Status code: 400.
{
```

```
msg: "Não foi possível deletar o evento.",
status: "error"
}
```

- **Exemplos de código:**

- OBS: Estrutura de objeto esperada ao salvar ou editar eventos, se não seguir essa estrutura pode dar errado. estrutura abaixo:

```
const obj = {
  // Obrigatório colocar o id do usuário que criou o evento
  idOrganizador: id,
  nomeEvento: "SEILA",
  qtdHomens: 0,
  qtdMulheres: 0,
  qtdCrianças: 0,
  endereco: "NÃO SEI!",
  custoLocal: 0,

  carnes: {

    bovino: {

      contraFile: {
        selecionado: true,
      },

      maminha: {
        selecionado: true,
      },

      alcatra: {
        selecionado: false
      }

    },

    suino: {

      costela: {
        selecionado: true,
      },

      fileSuino: {
        selcionado: false
      },


```

```
        lombo: {
            selecionado: false
        }
    },
    frango: {
        sobrecoxa: {
            selecionado: true
        },
        coracao: {
            selecionado: true
        },
        asa: {
            selecionado: true
        }
    },
    linguicas: {
        toscana: {
            selecionado: false
        },
        cuiabana: {
            selecionado: false
        },
        linguicaFrango: {
            selecionado: true
        }
    }
},
bebidas: {
    agua: {
        selecionado: true
    },
    refri: {
        selecionado: true
    },
    cerveja: {
        selecionado: true
    }
}
```

```
    },  
  
    suco: {  
        selecionado: true  
    }  
},  
  
suprimentos: {  
  
    copoDesc: {  
        selecionado: true,  
    },  
  
    talheres: {  
        selecionado: true  
    },  
  
    prato: {  
        selecionado: true  
    },  
  
    carvao: {  
        selecionado: true  
    },  
  
    guardanapos: {  
        selecionado: true  
    },  
  
    palitos: {  
        selecionado: true  
    }  
},  
  
acompanhamentos: {  
  
    arroz: {  
        selecionado: false  
    },  
  
    farofa: {  
        selecionado: true  
    },  
  
    pao: {  
        selecionado: false  
    },  
  
    paoAlho: {  
        selecionado: true  
    }  
}
```

```

    },
    vinagrete: {
      selecionado: true
    },
    queijoCoalho: {
      selecionado: false
    }
  }
}

```

■ GET

```

async function buscarEvento()
  const request = await fetch(`${url}/evento/${id}`, {
    method: "GET",
    headers: {
      "Content-Type": "application/json"
    },
  });

  const response = await request.json();

  console.log(response);
}

```

■ POST

```

async function criarEvento(obj) {

  const request = await fetch(`${url}/evento`, {
    method: "POST",
    headers: {
      "Content-Type": "application/json"
    },
    body: JSON.stringify(obj)
  });

  const response = await request.json();

  console.log(response);
}

```

```
}
```

■ PUT

```
async function editarEvento(obj) {  
  
    const request = await fetch(`${url}/evento/${id}`, {  
        method: "PUT",  
        headers: {  
            "Content-Type": "application/json"  
        },  
        body: JSON.stringify(obj)  
    });  
  
    const response = await request.json();  
  
    console.log(response);  
  
}
```

■ DELETE

```
async function apagarEvento(obj) {  
  
    const request = await fetch(`${url}/evento/${id}`, {  
        method: "DELETE",  
        headers: {  
            "Content-Type": "application/json"  
        },  
    });  
  
    const response = await request.json();  
  
    console.log(response);  
  
}
```

- Ejemplos de código:
 - GET

```

async function buscarEvento(idEvento) {

    const request = await fetch(`${url}/evento/${idEvento}`, {
        method: "GET",
        headers: {
            "Content-Type": "application/json"
        },
    });

    const response = await request.json();

    console.log(response);

}

```

◦ POST

```

async function criarEvento() {

    const request = await fetch(`${url}/evento/`, {
        method: "POST",
        headers: {
            "Content-Type": "application/json"
        },
        body: JSON.stringify(obj)
    });

    const response = await request.json();

    console.log(response);

}

```

◦ PUT

```

async function editarEvento(idEvento) {

    const request = await fetch(`${url}/evento/${idEvento}`, {
        method: "PUT",
        headers: {
            "Content-Type": "application/json"
        },
        body: JSON.stringify(obj)
    });

}

```

```
const response = await request.json();

console.log(response);

}
```

◦ DELETE

```
async function editarEvento(idEvento) {

    const request = await fetch(`${url}/evento/${idEvento}`, {
        method: "DELETE",
        headers: {
            "Content-Type": "application/json"
        },
    });

    const response = await request.json();

    console.log(response);

}
```

<https://cashurrasco-api.vercel.app/Evento/todos/idCriador>

- **Métodos suportados:** GET.
- **Parâmetros de consulta:** Além de passar o endpoint “todos”, é obrigatório colocar o ID do usuário que criou os eventos para buscar todos os churrascos que ele criou.
- **Resposta da API:**

```
// Exemplo de Resposta de sucesso:

// Status code: 200.
{
  msg: "Eventos retornados com sucesso!",
  status: "success",
  eventos: [ // Todos os eventos aqui ]
}
```



```
// Exemplo das respostas de erro:

// Status code: 404.
{
  msg: "Nenhum evento encontrado.",
  status: "error"
}
```

- o **Exemplo de código:**

```
async function todosEventos(id) {

  const request = await fetch(`${url}/evento/todos/${id}`, {
    method: "GET",
    headers: {
      "Content-Type": "application/json"
    },
  });

  const response = await request.json();

  console.log(response);

}
```