

# Sites **Dinâmicos**

Desafio - **Nível Básico**



E-mail: [contato@gama.academy](mailto:contato@gama.academy)

Website: [gama.academy](http://gama.academy)

# Sites Dinâmicos

**1ª Edição**

**Autoria:** Hendy Almeida, Vinicius Oliveira

**Editora:** Lauren da Silveira Francke

**Consultor de Acessibilidade:** Eduardo Zangrossi Lino

**Consultora de Design:** Juliana Moraes

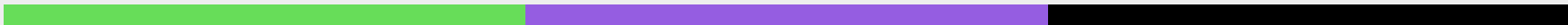
**Equipe Multidisciplinar:**

Natália Gomes

Copyright © Gama Academy 2022

Proibida a reprodução e a distribuição sem autorização prévia.

# Sumário



Clique nos links para ir ao capítulo correspondente:

- [Objetivos de aprendizagem](#)
- [Instruções gerais](#)
- [Entregáveis](#)
- [Material Complementar](#)



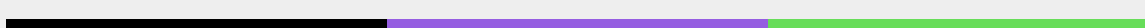


# Objetivos de aprendizagem

Coloque em prática seus conhecimentos de Javascript e HTML criando um site com conteúdo dinâmico e consumo de API.

**Esperamos que ao final do desafio você consiga:**

- Consumir dados de API com Javascript;
- Navegar pelo DOM com Javascript;
- Exibir conteúdos dinâmicos nas páginas do site;
- Enviar dados para API;





# Instruções gerais

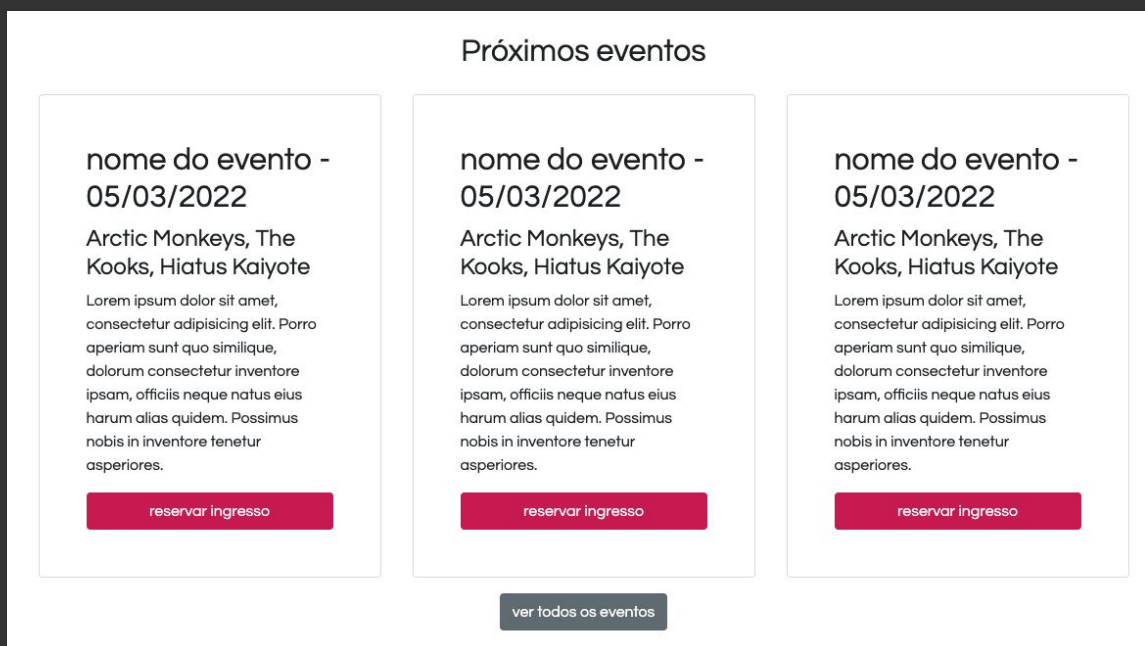
Olá!

Uma casa de show chamada Sound Garden será inaugurada e precisa de um site que exiba os eventos e permita que os clientes reservem ingressos através da Landing Page.

O site está separado em duas partes:

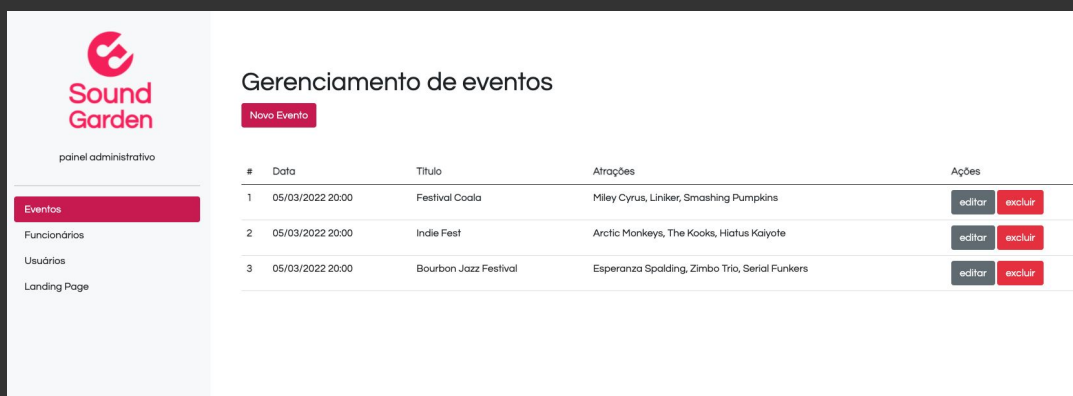
## 1- Landing Page:

Permite visualizar eventos e reservar ingressos.



## 2- Paineil Administrativo:

Permite criar, editar e excluir eventos.





# Instruções gerais

---

Disponibilizamos [o código HTML e CSS](#) do painel de controle para que você foque no desenvolvimento Javascript.

A API está pronta para receber esses dados e seu desafio será fazer a integração do front-end com o back-end.

Na [documentação da API](#) você podem consultar:

- endpoints com métodos HTTP
- parâmetros
- retorno de cada endpoint



# Instruções gerais

## Painel de controle [\(link\)](#)

1. **Criar Evento:** receba os dados que o usuário inserir através do formulário da página **criar-evento.html** e envie essa informação para a API utilizando o método POST.
2. **Editar Evento:** para cada evento listado, existe um botão editar que deve direcionar para **editar-evento.html?id=0**, com o id do evento selecionado. Na página de edição, o formulário deve aparecer preenchido com os dados do evento, permitindo a edição das informações.
3. **Excluir Evento:** para cada evento listado, existe um botão editar que deve direcionar para **excluir-evento.html?id=0**, com o id do evento selecionado. Na página de edição, o formulário deve aparecer preenchido com os dados do evento, porém com os campos desabilitados. Ao clicar no botão "excluir para sempre", deve fazer uma requisição na API para excluir o evento do banco de dados.
4. **Ver reservas do evento:** Listar as reservas de ingressos do evento selecionado.



# Instruções gerais

## Landing Page [\(link\)](#)

1. Eventos principais: na [página inicial](#) deve exibir 3 eventos cadastrados.
2. **Listar eventos:** na página **eventos.html** deve listar todos os eventos cadastrados. Como no exemplo desse [link](#).
3. **Reserva de ingresso:** ao clicar em "reservar ingresso", deve abrir um **modal** com formulário (nome e email), para que o usuário possa preencher os dados e reservar o ingresso.

## FUNCIONALIDADE OPCIONAL

1. Criar um banner rotativo para a primeira seção da [landing page](#). Esse banner deverá exibir alguns eventos de destaque para os usuários.





# Entregáveis

A entrega do desafio é feita através de um formulário na plataforma Gama. Antes de enviar, verifique se o projeto atende aos critérios de avaliação.

## **Critérios de Avaliação**

- O site deve ser disponibilizado no Github Pages, com todas as imagens renderizando corretamente.
- Você deve sempre fazer tratamento de erros quando usar as promises usando as estruturas de catch.
- Funcionalidades:
  - Criar Evento;
  - Editar Evento;
  - Excluir Evento;
  - Listar eventos;
  - Criar reservas nos eventos;
  - Listar reservas;

## **Quais os entregáveis desse desafio?**

- Código disponibilizado no Github;
- Páginas publicadas no Github Pages;



# Material complementar

**Caso você queira aprofundar seu conhecimento, abaixo seguem algumas dicas de materiais complementares:**

- [Modelo de Objeto de Documento \(DOM\)](#)
- [API Fetch](#)
- [Métodos HTTP](#)



