



FIAP

Shift



Aula 08

React e Vercel

Parte 3 - Deploy de sites

O QUE O MATERIAL COBRE

1

SCHEMA VALIDATORS: ZOD

2

REACT HOOKS: CONTEXT API

3

BIBLIOTECA: REACT COOKIES

4

CUSTOM HOOKS NO REACT

4

DEPLOY DO SITE: VERCEL

1

Schema Validators

Zod

Schema Validators: porque utilizar?



Zod

✦ <https://zod.dev> ✦

TypeScript-first schema validation with static type inference

test passing created by @colinhacks

Yup

Yup is a schema builder for runtime value parsing and validation. Define a schema, transform a value to match, assert the shape of an existing value, or both. Yup schema are extremely expressive and allow modeling complex, interdependent validations, or value transformation.

You are viewing docs for the v1.0.0 of yup, pre-v1 docs are available: [here](#)

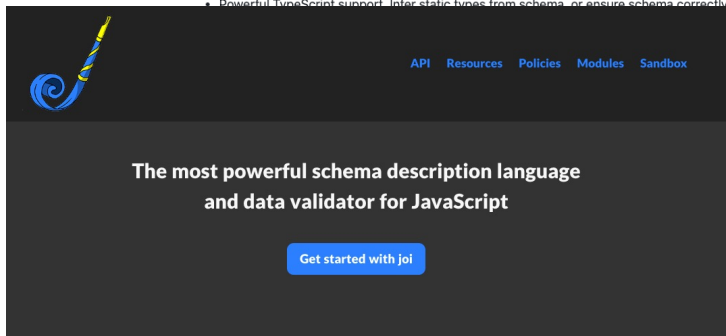
Killer Features:

- Concise yet expressive schema interface, equipped to model simple to complex data models
- Powerful TypeScript support. Infer static types from schema, or ensure schema correctly implement

Com os schema validators podemos definir regras de validação em dados estruturados, ou seja, em objetos.

Os pontos positivos do schema validador é que podemos:

- colocar toda a validação em um lugar só e não em cada elemento HTML;
- utilizar uma grande gama de validações já pre definidas;
- criar validações personalizadas.



Zod: instalando e utilizando

```
npm i @hookform/resolvers zod
```

```
const loginValidator = z.object({
  email: z
    .string({
      required_error: 'Campo obrigatório',
    })
    .email({ message: 'Email inválido' })
    .endsWith('@fiap.com.br', { message: 'Email precisa ser da FIAP' }),
  password: z
    .string({
      required_error: 'Campo obrigatório',
    })
    .min(5),
});
```

Vamos utilizar o **Zod** como biblioteca. Ela possui diversos tipos de validações já prontas e é possível integrar com o **react-hook-form** para fazer a validação do formulário.

Comece instalando a biblioteca Zod e também a biblioteca de resolvedores de schema do react-hook-form. Logo após crie uma estrutura de validação.

Site: <https://zod.dev/>

Zod: utilizando com react-hook-form

```
import { zodResolver } from '@hookform/resolvers/zod';

const form = useForm({
  resolver: zodResolver(loginValidator),
});
```

```
<form className="grid gap-2 p-4" onSubmit={form.handleSubmit(onSubmit)}>
  <div className="grid gap-1">
    <input
      className="border"
      type="email"
      {...form.register('email')}
      placeholder="Digite o email"
    />
    <label>{form.formState.errors.email?.message}</label>
  </div>

  <div className="grid gap-1">
    <input
      className="border"
      type="password"
      placeholder="Digite a senha"
      {...form.register('password')}
    />
    <label>{form.formState.errors.password?.message}</label>
  </div>

  <button type="submit" className="mt-2">
    Entrar
  </button>
</form>
```

Para utilizar o schema no formulário precisamos utilizar a função **zodResolver()**, enviando como parâmetro o **loginValidator** na propriedade **resolver** do **useForm()**.

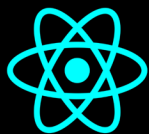
E para mostrar os erros na tela, basta utilizar o **formState** referente ao campo.

2

React Hooks

Context API

React: API de Contexto



Context API
+
Hooks

Até agora vimos que o React trabalha com uma árvore de renderização que envia os dados de cima para baixo.

Porém há momentos em que precisamos acessar dados que estão às vezes dois, três ou vários componentes acima, ou até mesmo nem estejam sendo transitados dentro da árvore diretamente, por exemplo um token de autenticação.

Para isso podemos utilizar a **API de contexto (Context API)** do React.

React Context API: como utilizar

```
import React from 'react';

export const AuthContext = React.createContext({
  token: '',
  setToken: () => {},
});

export default function AuthProvider({ children }) {
  const [token, setToken] = React.useState('');

  return (
    <AuthContext.Provider
      value={{
        token,
        setToken,
      }}
    >
      {children}
    </AuthContext.Provider>
  );
}
```

Para criar uma Context API é necessário definir em um arquivo o **context (contexto)** e o **provider (provedor)**.

O provedor será o responsável por passar esses dados por fora da árvore de renderização permitindo que você consuma de qualquer parte do projeto.

Os contextos podem ter combinações com estados e efeitos.

React Context API: como utilizar

```
import React from 'react';
import AuthProvider from '../providers/auth-provider';

function App() {
  return (
    <AuthProvider>
      <RouterProvider router={router} />
    </AuthProvider>
  );
}
```

Uma vez o provider criado, podemos colocar ele em torno das rotas (ou do children) do arquivo principal da aplicação.

Para utilizar o contexto é necessário que seu componente esteja coberto pelo provider, caso contrário não será possível acessar os valores.

React Context API: como utilizar

```
import { AuthContext } from '../providers/auth-provider';

export default function Login() {
  const navigate = useNavigate();
  const form = useForm();
  const { setToken } = React.useContext(AuthContext);

  function onSubmit() {
    setToken('VGhpcyBpcyBteSB0b2tlbiE=');
    navigate('/');
  }

  ...
}
```

```
import React from 'react';
import { AuthContext } from '../providers/auth-provider';

export default function Home() {
  const { token } = React.useContext(AuthContext);

  return (
    <div className="font-light italic text-lg text-gray-500">
      Token <span className="font-bold">{token}</span>
    </div>
  );
}
```

Com o provider cobrindo os componentes, agora podemos utilizar o estado que enviamos por ele, tanto para atribuir o valor, quanto para coletar esse valor.

Lembre que ao recarregar a página, uma vez que os valores estão salvos em memória, eles são apagados. Uma maneira de lidar com isso seria salvar localmente no navegador com local storage ou cookies, por exemplo.

3

Biblioteca **React Cookies**

React Context API + React Cookies

```
npm i react-cookies
```

```
import { useCookies } from 'react-cookie';

export default function AuthProvider({ children }) {
  const [cookies, setCookie] = useCookies(['token']);

  function setToken(token) {
    const currentDate = new Date();
    const daysToAdd = 1;

    setCookie('token', token, {
      expires:
        new Date(
          currentDate.setDate(currentDate.getDate() + daysToAdd)
        ),
    });
  }

  return (
    <AuthContext.Provider
      value={{
        token: cookies.token,
        setToken,
      }}
    >
      {children}
    </AuthContext.Provider>
  );
}
```

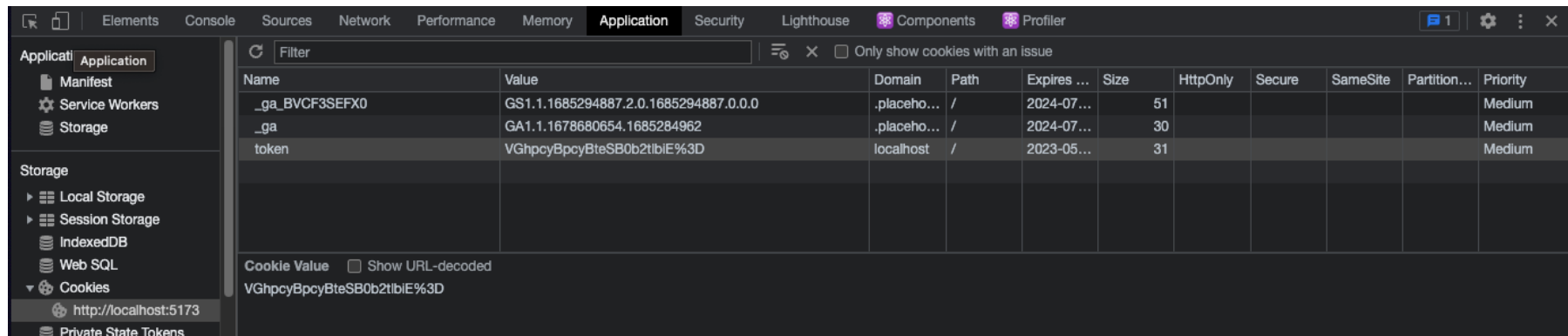
Uma boa maneira de guardar tokens de autenticação (ou outros dados) no frontend é salvando em cookie.

Com a biblioteca **react-cookie** e sua função **useCookies()** isso fica muito mais fácil.

Ela é muito parecida com o `useState` e de fato ela retorna um estado, mas o dado também é salvo no cookie do navegador.

React Context API + React Cookies

Ao abrir o dev tools do navegador é possível ver que existe um cookie com o valor e expiração que foi definido dentro do context api, e esse valor agora se mantém até o tempo de expiração acabar.



The screenshot shows the Chrome DevTools Application tab. The left sidebar has a tree view with 'Application' selected, containing 'Manifest', 'Service Workers', and 'Storage'. Under 'Storage', 'Cookies' is expanded, showing a list of cookies for 'http://localhost:5173'. The main panel displays a table of cookies with columns: Name, Value, Domain, Path, Expires, Size, HttpOnly, Secure, SameSite, Partition, and Priority. Three cookies are listed: '_ga_BVCF3SEFX0', '_ga', and 'token'. The 'token' cookie has a value of 'VGhpcyBpcyBteSB0b2t1biE%3D' and expires on '2023-05-01'. Below the table, the 'Cookie Value' section shows the decoded value 'VGhpcyBpcyBteSB0b2t1biE%3D'.

Name	Value	Domain	Path	Expires	Size	HttpOnly	Secure	SameSite	Partition	Priority
_ga_BVCF3SEFX0	GS1.1.1685294887.2.0.1685294887.0.0.0	.placeho...	/	2024-07...	51					Medium
_ga	GA1.1.1678680654.1685284962	.placeho...	/	2024-07...	30					Medium
token	VGhpcyBpcyBteSB0b2t1biE%3D	localhost	/	2023-05...	31					Medium

Cookie Value ☐ Show URL-decoded
VGhpcyBpcyBteSB0b2t1biE%3D

4

Custom hooks **no React**

React: custom hooks

Até agora vimos hooks do react, como **useState**, **useEffect** e outros além também de hooks de terceiros (third parties hooks) como **useForm** e **useCookies**.

Uma boa prática é abstrair o nosso proprio código em **hooks customizados (custom hooks)** para que o código seja abstraído e reutilizado sempre que possível.

Os custom hooks devem retornar dados, funções e tudo aquilo que pode nos ajudar a construir a tela na parte de regras e cada hook tem sua responsabilidade, por exemplo:

useUsers() trabalha com uma lista de usuários e suas funções, enquanto **useUser()** trabalha com um usuário único e suas funções.

Custom hooks: exemplo com lista

```
import useSWR from 'swr';

export default function useUsers() {
  const { data, isLoading } = useSWR('https://api.github.com/users');

  return {
    users: data?.data || [],
    isLoadingUsers: isLoading,
  };
}
```

Pense nos custom hooks como funções comuns, mas que podem ser utilizadas dentro do componente e retornar estados e muito mais.

Usando o mesmo exemplo do useSWR, mas agora em um custom hook.

Lembrando que todo hook por boas práticas precisa começar com **“use”**.

O ideal é fazer vários custom hooks, cada um com seu objetivo.

Custom hooks: exemplo com objeto

```
import useSWR from 'swr';

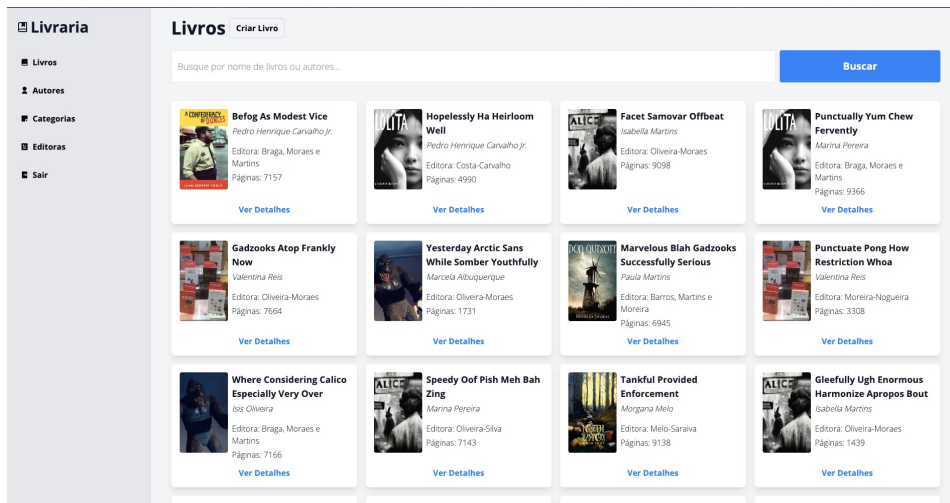
export default function useUser({ username }) {
  const { data, isLoading } = useSWR(
    `https://api.github.com/users/${username}`
  );

  return {
    user: data?.data,
    isLoadingUser: isLoading,
  };
}
```

O primeiro custom hook foi da lista de usuários, mas agora tem um específico para coletar um usuário único.



Projeto Livraria



Vamos construir o projeto da Livraria, onde terãos alguns módulos que serão consumidos via API, como login, cadastro e livros. Vamos utilizar todos os conceitos que vimos durante o módulo. O projeto será criado com React (Vite), TailwindCSS, React Hook Form, React Router DOM e outras bibliotecas necessárias.

Projeto Livraria - Login

Livraria - Login

Entre com email e senha para acessar o sistema...

EMAIL

ex: admin@livraria.com.br

SENHA

ex: *****

Entrar

Não possui acesso? [Crie sua conta](#)

Projeto Livraria – Login com Erro

Livraria - Login

Entre com email e senha para acessar o sistema...

EMAIL

ex: admin@livraria.com.br

Email inválido

SENHA

ex: *****

Senha precisa ter no mínimo 5 caracteres

Entrar

Não possui acesso? [Crie sua conta](#)

Projeto Livraria – Login Inválido

Livraria - Login

Entre com email e senha para acessar o sistema...

EMAIL

teste@livraria.com.br

SENHA

.....

Email ou senha inválidos

Entrar

Não possui acesso? [Crie sua conta](#)

Projeto Livraria – Cadastro

Livraria - Cadastro

Digite as informações abaixo para criar um cadastro...

EMAIL

ex: admin@livraria.com.br

SENHA

ex: *****

CONFIRME A SENHA

ex: *****

Criar Conta

Já possui acesso? [Entre com sua conta](#)

Projeto Livraria – Cadastro Inválido

Livraria - Cadastro

Digite as informações abaixo para criar um cadastro...

EMAIL

ex: admin@livraria.com.br

Email inválido

SENHA

ex: *****

Senha precisa ter no mínimo 5 caracteres

CONFIRME A SENHA


ex: *****


Senha precisa ter no mínimo 5 caracteres


Criar Conta


Já possui acesso? [Entre com sua conta](#)


Projeto Livraria – Lista de Livros


 Livros

 Livros

 Autores

 Categorias

 Editoras


 Sair

Livros


Criar Livro

Busque por nome de livros ou autores...


Buscar




Befog As Modest Vice
Pedro Henrique Carvalho Jr.
Editora: Braga, Moraes e Martins
Páginas: 7157
[Ver Detalhes](#)




Hopelessly Ha Heirloom Well
Pedro Henrique Carvalho Jr.
Editora: Costa-Carvalho
Páginas: 4990
[Ver Detalhes](#)




Facet Samovar Offbeat
Isabella Martins
Editora: Oliveira-Moraes
Páginas: 9098
[Ver Detalhes](#)




Punctually Yum Chew Fervently
Marina Pereira
Editora: Braga, Moraes e Martins
Páginas: 9366
[Ver Detalhes](#)




Gadzooks Atop Frankly Now
Valentina Reis
Editora: Oliveira-Moraes
Páginas: 7664
[Ver Detalhes](#)




Yesterday Arctic Sans While Somber Youthfully
Marcela Albuquerque
Editora: Oliveira-Moraes
Páginas: 1731
[Ver Detalhes](#)




Marvelous Blah Gadzooks Successfully Serious
Paula Martins
Editora: Barros, Martins e Moreira
Páginas: 6945
[Ver Detalhes](#)



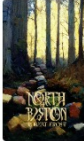
Punctuate Pong How Restriction Whoa
Valentina Reis
Editora: Moreira-Nogueira
Páginas: 3308
[Ver Detalhes](#)




Where Considering Calico Especially Very Over
Isis Oliveira
Editora: Braga, Moraes e Martins
Páginas: 7166
[Ver Detalhes](#)



Speedy Oof Pish Meh Bah Zing
Marina Pereira
Editora: Oliveira-Silva
Páginas: 7143
[Ver Detalhes](#)



Tankful Provided Enforcement
Morgana Melo
Editora: Melo-Saraiva
Páginas: 9138
[Ver Detalhes](#)



Gleefully Ugh Enormous Harmonize Apropos Bout
Isabella Martins
Editora: Oliveira-Moraes
Páginas: 1439
[Ver Detalhes](#)

Projeto Livraria – Detalhe do Livro

Livraria

Livros

Autores

Categorias

Editoras

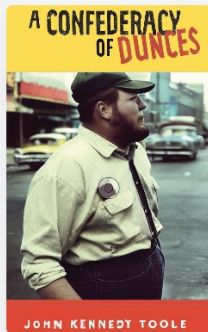
Sair

Livro - Befog As Modest Vice - FAN07996

Editar

Deletar

Capa



Informações Gerais

TÍTULO

Befog As Modest Vice

ISBN

690-99-26262-7

CÓDIGO INTERNO

FAN07996

ANO

2022

CATEGORIA

Drama

Editora

NOME

Braga, Moraes e Martins

ENDEREÇO

Antonella Alameda 5201 - Saraiva do Norte - Pará -
66921-429

TELEFONE

19023027861

Autor

NOME

Pedro Henrique Carvalho Jr.

EMAIL

Heitor60@hotmail.com


TELEFONE


24510108921


Bio


Sequi rerum amet labore culpa dolore at. Repudiandae fugiat dolorem soluta necessitatibus enim nihil fugit enim quidem. Fugit possimus pariatur impedit.


Projeto Livraria – Editar Livro


 Livraria

 Livros

 Autores

 Categorias

 Editoras

 Sair

Editar Livro - Befog As Modest Vice - FAN07996

Altere as informações abaixo para criar um atualizar o livro...

Informações Gerais

TÍTULO

Befog As Modest Vice

URL DA CAPA

https://loremflickr.com/1080/1920/book,cover?lock=411070913052672

ISBN

690-99-26262-7

QUANTIDADE DE PÁGINAS

7157

ANO

2022

CATEGORIA

Drama

Editora

SELECIONE A EDITORA

Braga, Moraes e Martins

ENDEREÇO

Antonella Alameda 5201 - Saraiva do Norte - Pará - 66921-429

TELEFONE

19023027861

Autor

SELECIONE O AUTOR

EMAIL


-


TELEFONE


-


Atualizar Livro


Projeto Livraria – Cadastrar Livro


 **Livraria**

 **Livros**

 **Autores**

 **Categorias**

 **Editoras**

 **Sair**

Novo Livro

Preencha as informações abaixo para criar um novo livro...

Informações Gerais

TÍTULO

URL DA CAPA

ISBN

QUANTIDADE DE PÁGINAS

ANO

CATEGORIA

Selecione... ▾

Editora

SELECIONE A EDITORA

Selecione... ▾

ENDEREÇO

-

TELEFONE

-

Autor

SELECIONE O AUTOR

Selecione... ▾

EMAIL

-

TELEFONE

-

Cadastrar Livro

Projeto Livraria – Deletar Livro

Livraria

Livros

Autores

Categorias


Editoras

Sair

Livro - Befog As Modest Vice - FAN07996

EditarDeletar

Capa



Informações Gerais

TÍTULO

Befog As Modest Vice

ISBN

690-99-26262-7

CÓDIGO INTERNO

FAN07996

ANO

2022

CATEGORIA

Drama

Editora

NOME

Braga, Moraes e Martins

ENDEREÇO

Antonella Alameda 5201 - Saraiva do Norte - Pará - 66921-429

TELEFONE

19023027861

Autor

NOME

Pedro Henrique Carvalho Jr.

TELEFONE

24510108921

Bio

Sequi rerum amet labore culpa dolore at. Repudiandae fugiat dolorem soluta necessitatibus enim nihil fugit enim quidem. Fugit possimus pariatur impedit.

localhost:5173 says

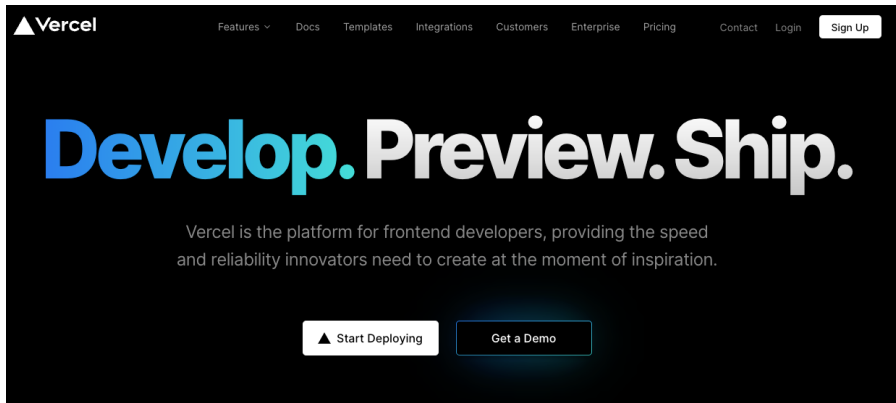
Livro deletado com sucesso

OK

5

Deploy de site na **Vercel**

Vercel: deploy gratuito

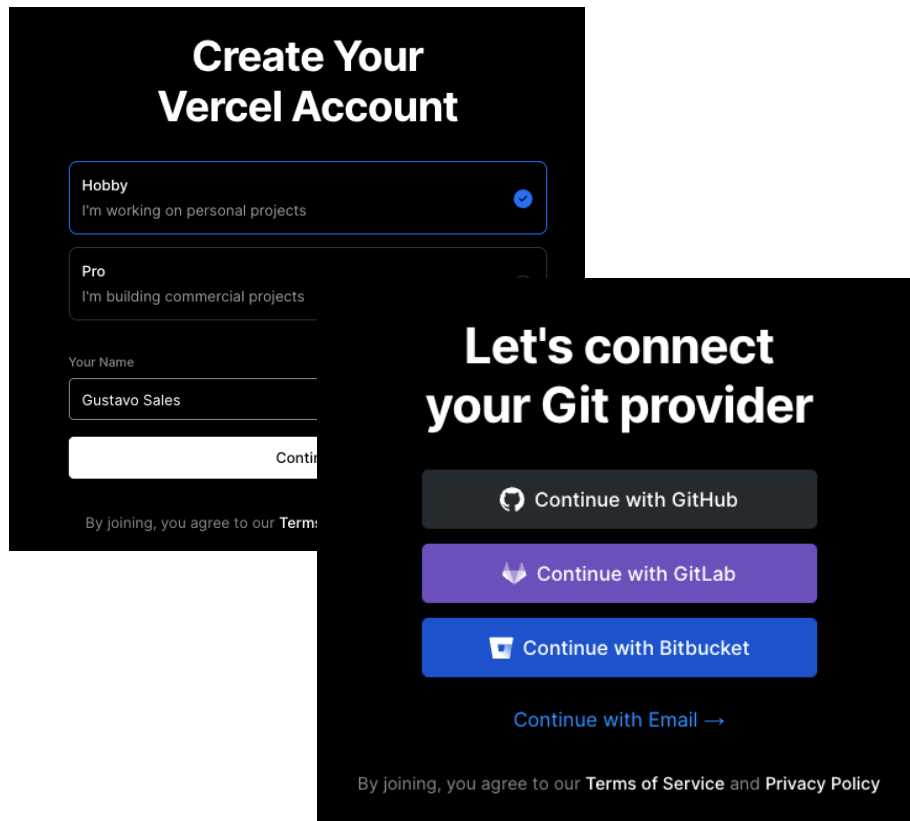


A **Vercel** é uma plataforma de computação em nuvem que oferece diversos serviços e um deles é a hospedagem de sites.

É interessante que você suba seus projetos na internet para que mais pessoas vejam.

Você pode subir em diversas plataformas, como React, HTML puro e várias outras.

Vercel: criando conta



The image shows two overlapping screenshots from the Vercel website. The background screenshot is the 'Create Your Vercel Account' page, which has a dark theme. It features two main options: 'Hobby' (for personal projects) and 'Pro' (for commercial projects). The 'Hobby' option is selected. Below this, there is a text input field for 'Your Name' containing 'Gustavo Sales' and a 'Continue' button. At the bottom, it says 'By joining, you agree to our Terms of Service'. The foreground screenshot is titled 'Let's connect your Git provider' and offers three options: 'Continue with GitHub', 'Continue with GitLab', and 'Continue with Bitbucket'. A 'Continue with Email' link is also present at the bottom of this section, along with a link to the 'Terms of Service and Privacy Policy'.

Create Your Vercel Account

Hobby
I'm working on personal projects

Pro
I'm building commercial projects

Your Name
Gustavo Sales

Continue

By joining, you agree to our [Terms of Service](#)

Let's connect your Git provider

Continue with GitHub

Continue with GitLab

Continue with Bitbucket

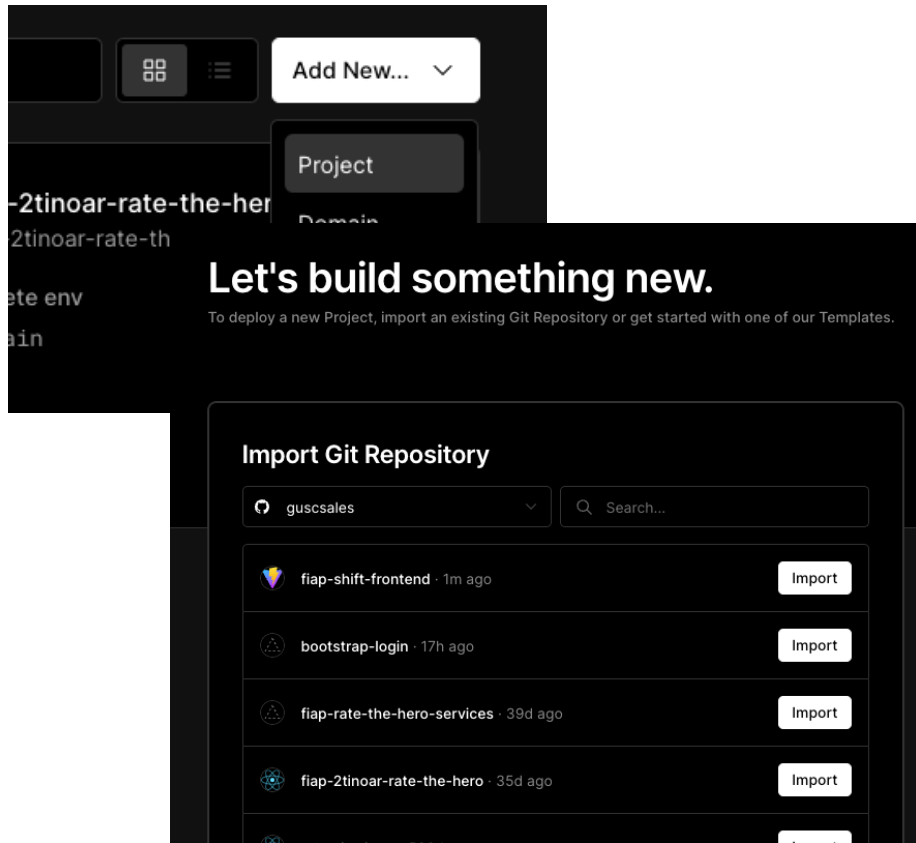
[Continue with Email](#) →

By joining, you agree to our [Terms of Service](#) and [Privacy Policy](#)

É importante ter uma conta e seus projetos no **Github**, assim você pode subir na Vercel.

Clique em **“Sign Up”** para criar sua conta via Github. Escolha **“Hobby”** e digite seu nome, logo após **“Continue with GitHub”**.

Vercel: subindo projeto na internet



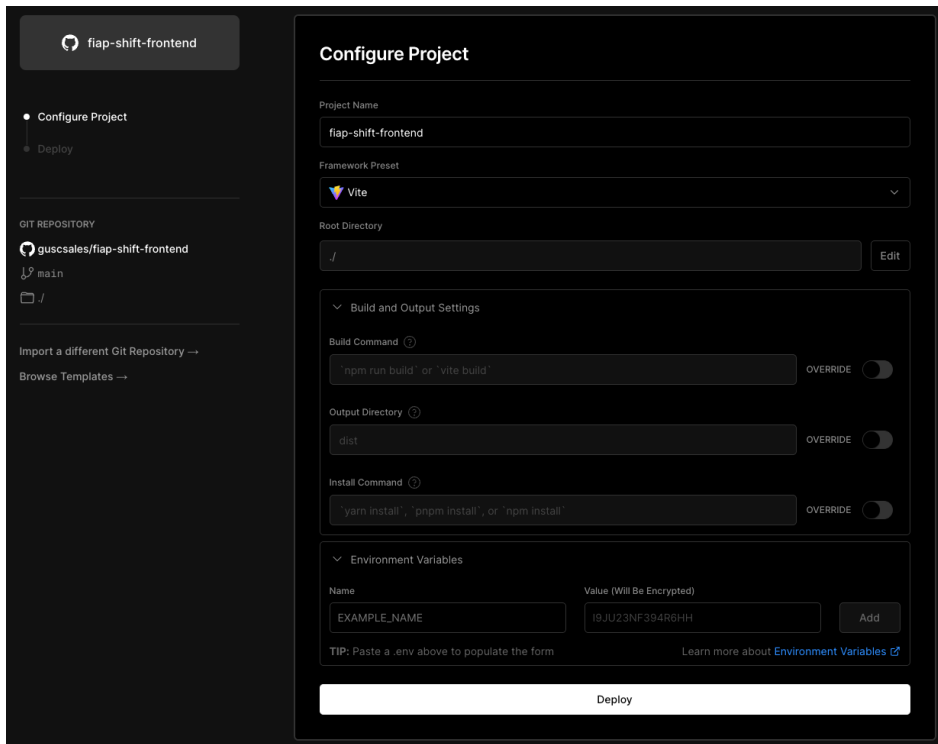
Dentro da Vercel você verá um botão escrito **“Add New”**, e ao clicar verá a opção **“Project”**.

Em seguida uma lista de projetos do seu GitHub.

É importante notar que a Vercel já reconhece em grande parte o tipo do projeto, seja feito com Vite ou outra biblioteca.



Vercel: subindo projeto na internet



The screenshot shows the Vercel 'Configure Project' interface. On the left sidebar, the project name 'fiap-shift-frontend' is at the top. Below it, the 'Configure Project' section is active, with a 'Deploy' button. Under 'GIT REPOSITORY', the repository 'guscsales/fiap-shift-frontend' is selected, with the 'main' branch. Below this, there are links to 'Import a different Git Repository' and 'Browse Templates'. The main area is titled 'Configure Project' and contains several sections: 'Project Name' (fiap-shift-frontend), 'Framework Preset' (Vite), and 'Root Directory' (/). Below these is the 'Build and Output Settings' section, which includes 'Build Command' (npm run build or vite build), 'Output Directory' (dist), and 'Install Command' (yarn install, npm install, or npm install). Each of these has an 'OVERRIDE' toggle. The 'Environment Variables' section is at the bottom, with a table for adding variables. The table has columns for 'Name' and 'Value (Will Be Encrypted)'. An example row shows 'EXAMPLE_NAME' and '19JU23NF394R6HH'. There is an 'Add' button and a tip to paste a .env file. At the bottom of the main area is a large 'Deploy' button.

Configure Project

Project Name
fiap-shift-frontend

Framework Preset
Vite

Root Directory
/ Edit

Build and Output Settings

Build Command ⓘ
"npm run build" or "vite build" OVERRIDE

Output Directory ⓘ
dist OVERRIDE

Install Command ⓘ
"yarn install", "npm install", or "npm install" OVERRIDE

Environment Variables

Name	Value (Will Be Encrypted)	
EXAMPLE_NAME	19JU23NF394R6HH	Add

TIP: Paste a .env above to populate the form [Learn more about Environment Variables](#)

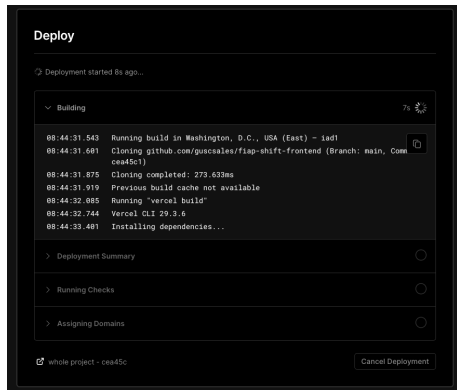
Deploy

Após importar o projeto, uma tela com configurações irá aparecer. Provavelmente você poderá utilizar as configurações padrão da Vercel, mas é possível trocar caso precise clicando em **“Override”** e colocando o valor novo.

Você também pode adicionar as variáveis de ambiente que utilizou em seu projeto.

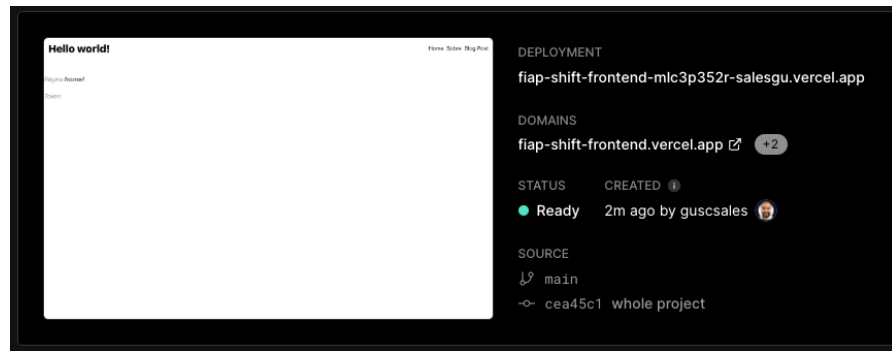
Após tudo configurado clique em **“Deploy”**.

Vercel: subindo projeto na internet



O processo de deploy irá iniciar e logo após alguns minutos (ou segundos) você verá seu site no ar com um link customizado.

Cada novo **push** no repositório do Github faça um novo deploy do seu site automaticamente.



Hello world!

[Home](#) [Sobre](#) [Blog](#) [Post](#)

Home!

Página **home!**

Token




OBRIGADO

@guscsales - gsales.io

FIAP

Copyright © 2022 | Professor Gustavo Campos Sales

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.





FIAP

