

Roteiro da aula 14-02

Instale o NodeJS

Instale o VSCode

Configurando backend

Crie uma pasta chamada habits, e abra esta pasta no VSCode

Crie uma pasta chamada backend

No VSCode, Vá no Menu Terminal e clique em Novo Terminal

Digite os comandos abaixo:

1. `npm init -y` // cria o projeto Node
2. `npm install fastify` // instala dependência fastify – servidor web
3. `npm install typescript -D` // instala a dependência do typescript no projeto
4. `npx tsc --init` // cria o arquivo de configuração tsconfig.json
5. `npm install tsx -D` // instala a dependência tsx, que transforma o arquivo TS em JS (já que Node não reconhece TypeScript)
6. `npx tsx src/server.ts` // executa o arquivo
7. no package.json, alterar test para dev: `tsx watch src/server.ts`
8. No arquivo server.ts, digitar:

```
import Fastify from 'fastify'
```

```
const app = Fastify()
```

```
app.get('/hello', {
```

```
  return 'Hello World'
```

```
})
```

```
app.listen({
```

```
  port: 3333,
```

```
})
```

```
.then( () => {
```

```
  console.log('Http Server running')
```

```
})
```

9. Instalar extensão do prisma no VSCode

10. npm install prisma -D
11. npm install @prisma/client
12. npx prisma init --datasource-provider SQLite
13. no arquivo prisma/schema.prisma, digite

```
model Habit {  
  id      String @id @default(uuid())  
  title   String  
  created_at DateTime  
  @@map("habits")  
}
```

14. npx prisma migrate dev
15. npx prisma studio
16. criar um exemplo de hábito
17. no arquivo server.ts, adicione

```
import { PrismaClient } from '@prisma/client'  
  
import cors from '@fastify/cors'  
  
const prisma = new PrismaClient()  
app.register(cors)  
  
app.get('/hello2', async () => {  
  const habits = await prisma.habit.findMany({  
    where: {  
      title: {  
        startsWith: 'beber'  
      }  
    }  
  })  
  
  return habits  
})  
  
18. teste as APIs
```

Configurando frontend

Em um novo terminal, na pasta habits, execute os comandos abaixo:

1. `npm create vite@latest`
2. defina o nome do projeto como frontend
3. Escolha React
4. Escolha TypeScript
5. Na pasta frontend ...
6. `npm install`
7. `npm run dev`
8. veja aplicação rodando...
9. deixe o arquivo `src/App.tsx` como

```
function App() {
```

```
  return (
```

```
    <h1> Hello World </h1>
```

```
  )
```

```
}
```

```
export default App
```

10. perceba as alterações

11. criar um arquivo em `src/componentes/Habit.tsx`

```
interface HabitProps {
```

```
  completed: number
```

```
}
```

```
export function Habit(props: HabitProps){
```

```
  return (
```

```
    <div >
```

```
      Teste {props.completed}
```

```
    </div>
```

```
  )
```

```
}
```

12. alterar o arquivo `src/App.tsx` para

```
import { Habit } from "../components/Habit";
```

```
function App() {
```

```
  return (
```

```
    <>
```

```

    <Habit completed={2}/>
    <Habit completed={1}/>
    <Habit completed={0}/>
  </>
)
}
export default App

```

13. npm install -D tailwindcss postcss autoprefixer
14. npx tailwindcss init -p
15. instalar extensão tailwind e postcss no VSCode
16. criar arquivo src/styles/global.css

@tailwind base;

@tailwind utilities;

@tailwind components;

17. adicione a linha no arquivo src/App.tsx

```
import './styles/global.css'
```

18. altere o arquivo tailwind.config.cjs

content: [

'./src/**/*.tsx',

'./index.html'

],

19. altere o arquivo src/components/Habit.tsx

```
<div className="bg-zinc-900 w-20 h-10 text-white rounded m-2 flex items-center justify-center">
```

```
  Teste {props.completed}
```

```
</div>
```

20. npm run dev