

UNIVERSIDADE REGIONAL INTEGRADA DO ALTO URUGUAI E DAS MISSÕES
CAMPUS DE ERECHIM
DEPARTAMENTO DE ENGENHARIAS E CIÊNCIA DA COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO

KAUAN MATOS DA ROCHA

PSYCARE: PLATAFORMA DE TERAPIA ONLINE

ERECHIM - RS
2023

KAUAN MATOS DA ROCHA

PSYCARE: PLATAFORMA DE TERAPIA ONLINE

**Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do grau de Bacharel,
Departamento de Engenharias e Ciência
da Computação da Universidade Regional
Integrada do Alto Uruguai e das Missões
Campus de Erechim.**

Orientador: Malomar Alex Seminotti

ERECHIM - RS

2023

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

AGRADECIMENTOS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla dapibus elit at lectus fermentum, a tincidunt risus vehicula. Interdum et malesuada fames ac ante ipsum primis in faucibus. Fusce auctor placerat neque vitae mattis. In facilisis ipsum eget lectus porttitor, ac congue lacus aliquet. Ut laoreet vulputate dui eu hendrerit. Donec tincidunt mi vitae mauris volutpat aliquet. Curabitur mollis vel mi vitae bibendum. Donec bibendum nisi id ex eleifend maximus. Proin vehicula congue pretium. Sed vel iaculis magna. Nam et auctor tellus, vel bibendum leo. Vestibulum convallis tellus mollis sem molestie dictum. Praesent posuere mi volutpat, posuere felis eu, auctor purus. Phasellus pulvinar urna non mauris cursus porttitor.

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

(LOREM, Ipsum, 2023)

RESUMO

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla dapibus elit at lectus fermentum, a tincidunt risus vehicula. Interdum et malesuada fames ac ante ipsum primis in faucibus. Fusce auctor placerat neque vitae mattis. In facilisis ipsum eget lectus porttitor, ac congue lacus aliquet. Ut laoreet vulputate dui eu hendrerit. Donec tincidunt mi vitae mauris volutpat aliquet. Curabitur mollis vel mi vitae bibendum. Donec bibendum nisi id ex eleifend maximus. Proin vehicula congue pretium. Sed vel iaculis magna. Nam et auctor tellus, vel bibendum leo. Vestibulum convallis tellus mollis sem molestie dictum. Praesent posuere mi volutpat, posuere felis eu, auctor purus. Phasellus pulvinar urna non mauris cursus porttitor.

Palavras-chave: Lorem. Ipsum. Dolor.

ABSTRACT

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla dapibus elit at lectus fermentum, a tincidunt risus vehicula. Interdum et malesuada fames ac ante ipsum primis in faucibus. Fusce auctor placerat neque vitae mattis. In facilisis ipsum eget lectus porttitor, ac congue lacus aliquet. Ut laoreet vulputate dui eu hendrerit. Donec tincidunt mi vitae mauris volutpat aliquet. Curabitur mollis vel mi vitae bibendum. Donec bibendum nisi id ex eleifend maximus. Proin vehicula congue pretium. Sed vel iaculis magna. Nam et auctor tellus, vel bibendum leo. Vestibulum convallis tellus mollis sem molestie dictum. Praesent posuere mi volutpat, posuere felis eu, auctor purus. Phasellus pulvinar urna non mauris cursus porttitor.

Keywords: Lorem. Ipsum. Dolor.

LISTA DE ILUSTRAÇÕES

Figura 1 – Código exemplo em JavaScript.	3
Figura 2 – Comparação de códigos TypeScript e JavaScript.	4
Figura 3 – <i>Looping</i> de eventos NodeJS.	5
Figura 4 – Código exemplo em ReactJS.	6
Figura 5 – Estrutura base de um projeto NestJS.	7

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
REST	Representational State Transfer
HTML	HyperText Markup Language
ECMA	European Computer Manufacturers Association
CSS	Cascading Style Sheets

LISTA DE ALGORITMOS

SUMÁRIO

1	INTRODUÇÃO	1
2	EVOLUÇÃO DO MERCADO DE PSICOLOGIA	2
3	FERRAMENTAS DE DESENVOLVIMENTO	3
3.1	JavaScript	3
3.2	TypeScript	4
3.3	NodeJS	5
3.4	ReactJS	5
3.5	NestJS	7
3.6	PostgreSQL	8
	REFERÊNCIAS	9

1 INTRODUÇÃO

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla dapibus elit at lectus fermentum, a tincidunt risus vehicula. Interdum et malesuada fames ac ante ipsum primis in faucibus. Fusce auctor placerat neque vitae mattis. In facilisis ipsum eget lectus porttitor, ac congue lacus aliquet. Ut laoreet vulputate dui eu hendrerit. Donec tincidunt mi vitae mauris volutpat aliquet. Curabitur mollis vel mi vitae bibendum. Donec bibendum nisi id ex eleifend maximus. Proin vehicula congue pretium. Sed vel iaculis magna. Nam et auctor tellus, vel bibendum leo. Vestibulum convallis tellus mollis sem molestie dictum. Praesent posuere mi volutpat, posuere felis eu, auctor purus. Phasellus pulvinar urna non mauris cursus porttitor.

2 EVOLUÇÃO DO MERCADO DE PSICOLOGIA

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla dapibus elit at lectus fermentum, a tincidunt risus vehicula. Interdum et malesuada fames ac ante ipsum primis in faucibus. Fusce auctor placerat neque vitae mattis. In facilisis ipsum eget lectus porttitor, ac congue lacus aliquet. Ut laoreet vulputate dui eu hendrerit. Donec tincidunt mi vitae mauris volutpat aliquet. Curabitur mollis vel mi vitae bibendum. Donec bibendum nisi id ex eleifend maximus. Proin vehicula congue pretium. Sed vel iaculis magna. Nam et auctor tellus, vel bibendum leo. Vestibulum convallis tellus mollis sem molestie dictum. Praesent posuere mi volutpat, posuere felis eu, auctor purus. Phasellus pulvinar urna non mauris cursus porttitor.

3 FERRAMENTAS DE DESENVOLVIMENTO

Esse capítulo expõe as ferramentas que foram utilizadas para o desenvolvimento deste trabalho. Para o desenvolvimento das interfaces de usuário foi utilizado *ReactJS* auxiliado por outras bibliotecas secundárias. Já para desenvolver a API *back-end* foi aplicado o padrão REST através do *framework NestJS*. Por fim, conectado a tudo isso, o gerenciador de banco de dados *PostgreSQL* foi empregado. Nos próximos sub-capítulos tais tecnologias(e outras que também estão envolvidas) serão melhor pormenorizadas.

3.1 JavaScript

JavaScript é uma linguagem de programação adotada amplamente pela maioria das páginas *web*, além de ser utilizada também em outros ambientes fora do navegador através de tecnologias auxiliares. Trata-se de uma linguagem interpretada, originalmente implementada diretamente como parte dos navegadores *web* para que os scripts pudessem ser executados *client-side* sem a necessidade de passar por um servidor (MOZILLA, 2023).

JavaScript é baseado em protótipos, leve, dinâmico e *single-thread*, dando suporte a vários paradigmas de programação diferentes, como a programação orientada a objetos, a imperativa e a declarativa. Além disso, diferencia-se também por utilizar o tipo de compilação *just-in-time*, compilando seu código durante a sua execução (MOZILLA, 2023).

Figura 1 – Código exemplo em JavaScript.

```
1  const cityPopulations = {
2    'Nova Iorque': 8.4,
3    'São Paulo': 12.3,
4    'Londres': 8.9,
5    'Cidade do Cabo': 4.6,
6    'Bangcoc': 14.6
7  }
8
9  console.log('Cidade'.padEnd(20) + ' População(milhões)')
10
11 Object.entries(cityPopulations).forEach(([city, population]) => {
12   console.log(`${city.padEnd(20)} ${population}`)
13 })
14
15 // Cidade           População(milhões)
16 // Nova Iorque      8.4
17 // São Paulo        12.3
18 // Londres          8.9
19 // Cidade do Cabo   4.6
20 // Bangcoc          14.6
```

Fonte: Autor

A linguagem foi criada em 1995 principalmente pelo programador Brendan Eich a pedido da empresa NetScape para lidar com a validação de formulários HTML. Para que a linguagem evoluísse obedecendo padronizações já estabelecidas no mercado, os criadores da linguagem associaram-se à fundação ECMA um ano após sua criação, entretanto, o nome JavaScript já havia sido patenteado, decidindo-se seguir pelo nome oficial de ECMAScript.

Todavia, de acordo com Malavasi (2017), na comunidade de desenvolvedores a linguagem ainda é chamada de JavaScript até os dias atuais, utilizando o nome oficial ECMAScript apenas para referenciar suas versões.

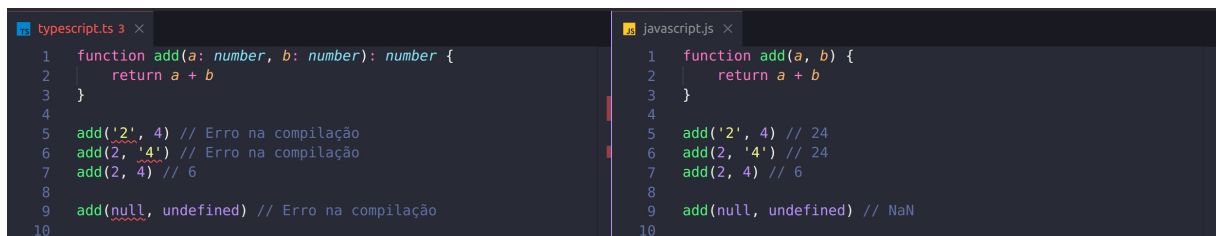
3.2 TypeScript

TypeScript é uma linguagem de programação desenvolvida pela Microsoft em 2012 de código aberto. Por definição, consiste-se de um superconjunto do JavaScript, adicionando também tipagem estática a linguagem original (MICROSOFT, 2023b).

Por tratar-se de superconjunto, tudo o que existe em sua linguagem “mãe” também está presente nesta linguagem, incrementada pelas novas suas funcionalidades, como por exemplo: tipos, orientação à objetos, *decorators*, *mixins*, módulos e *namespaces* (GOLDBERG, 2022). Além disso, essa peculiaridade também faz com que todo código JavaScript também seja um código TypeScript minimamente válido, facilitando assim possíveis migrações de projetos.

Pela natureza do JavaScript de ser compilado *just-in-time*, muitos erros frugais de desenvolvimento são percebidos apenas na hora da execução do código. Entretanto, pelo TypeScript ser estaticamente tipado, o paradigma de orientação a objetos é acentuadamente estimulado (GOLDBERG, 2022). Ao alinhar isso ao sistema de tipos da linguagem, erros de programação tornam-se evidentes muito brevemente, conforme exemplifica a figura a seguir:

Figura 2 – Comparação de códigos TypeScript e JavaScript.



Fonte: Autor

Além de tudo, segundo Microsoft (2023a), a estrutura padrão do JavaScript acabaram por não conseguir acompanhar o crescimento exponencial de tamanho, escopo e complexidade de suas novas aplicações, reduzindo a capacidade de escalabilidade da linguagem, problemática que é profundamente saciada pelas funcionalidades extras do TypeScript.

Por fim, pelos navegadores efetivamente darem suporte exclusivamente a códigos JavaScript, o TypeScript necessita ser pré-processado antes de ser empregado no *client-side* tornando sua linguagem “mãe” imprescindível à sua construção. Seu compilador padrão é o tsc, e outros agregadores mantidos pela comunidade (como o esbuild e o babel) podem ser utilizados a partir da necessidade específica de cada projeto.

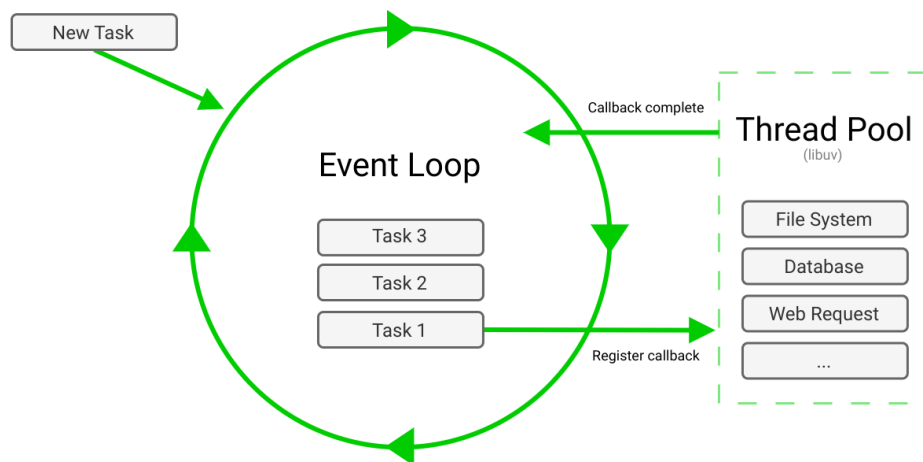
3.3 NodeJS

NodeJS é um ambiente de execução de código JavaScript e TypeScript externo a um navegador *web*. Este software inspira-se por sistemas como a máquina de eventos da linguagem Ruby e o Twist do Python para caracterizar-se pela sua arquitetura assíncrona e orientada por eventos.

Sua implementação é contrastante em relação à outras tecnologias pois não apresenta o modelo convencional de simultaneidade, onde conceitos de threads do sistema operacional são empregados. Seu *runtime* é executado por apenas uma *thread* que executa o *looping* de eventos, que perdura desde a criação da *thread* até quando não há mais retornos de chamadas a serem concluídos (FOUNDATION, 2023).

Dentro desse mesmo contexto, chamadas que normalmente seriam bloqueantes (que requerem recursos do sistema operacional) são realizadas assíncronamente utilizando a biblioteca libuv (BAHATI, 2022). Na figura abaixo é possível visualizar um diagrama simplificado desse fluxo.

Figura 3 – *Looping* de eventos NodeJS.



Fonte: Autor

Além disso, o NodeJS também dispõe do NPM, um gerenciador de pacotes reutilizáveis de código aberto, permitindo maior agilidade, flexibilidade e produtividade no processo de desenvolvimento (NPM, 2022).

3.4 ReactJS

ReactJS é uma biblioteca NodeJS de código aberto criada pelo Facebook em 2011 para o desenvolvimento de interfaces *web* em JavaScript e/ou Typescript. Comparando-o com outras bibliotecas e *frameworks* para o desenvolvimento *web front-end*, constata-se que seu diferencial

se encontra na abordagem de construção da interface baseada em componentes atualizados e sincronizados de forma muito mais otimizada.

A arquitetura base de um projeto ReactJS consiste em uma estrutura hierárquica de componentes onde cada componente representa uma parte da interface final do usuário, que possui sua própria lógica e aparência isolada de outros componentes. Nesse contexto, um dos caminhos ideais da construção do código encontra-se no princípio de “separação de conceitos”, onde cada seção(ou componente) é responsável por lidar com apenas um assunto separadamente e nada além disso, resultando em uma arquitetura mais modular e escalável (QAWWAS, 2022).

Empiricamente, cada componente React é uma função que retorna uma pequena porção de código HTML, que é estilizado por código CSS e passível de ser modificado pelas lógicas em JavaScript/TypeScript que existem dentro dessa mesma função, permitindo assim que esse pedaço de interface usufrua de todas funcionalidades que essas linguagens de programação oferecem.

Figura 4 – Código exemplo em ReactJS.

```
1  import { useEffect, useState } from 'react'
2  import svgLogo from './logo.svg'
3  import './styles.scss'
4
5  const Product = () => {
6
7      const [productName, setProductName] = useState<string>('')
8
9      useEffect(() => {
10         if(navigator.language === 'pt_BR')
11             setProductName('Produto X')
12         else
13             setProductName('Product X')
14     }, [])
15
16     return (
17         <div id='product'>
18             <img src={svgLogo} />
19             <h1>
20                 { productName }
21             </h1>
22         </div>
23     )
24 }
25
26 export default Product
```

Fonte: Autor

Atualmente, de acordo com a pesquisa Developer Survey realizada pela Exchange (2023), ReactJS é a biblioteca de desenvolvimento *web* mais utilizada no âmbito profissional, refletindo o amplo apoio que a biblioteca possui em novas funcionalidades e resolução de problemas.

3.5 NestJS

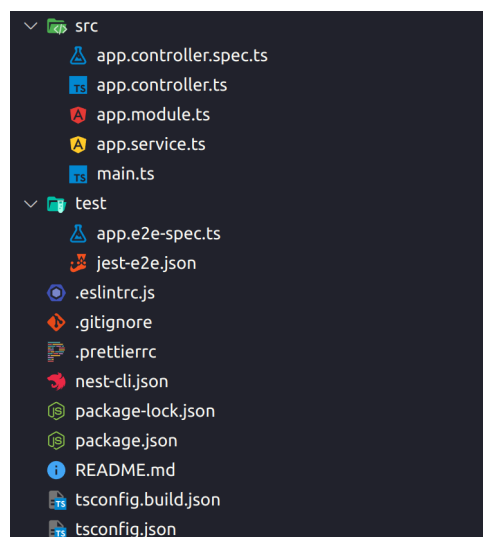
NestJS é um framework em NodeJS de código aberto lançado em 2017 para o desenvolvimento de aplicações *server-side*. Diante de outras tecnologias já consolidadas no JavaScript, esse *framework* destaca-se por sua implementação base sustentar-se em elementos de programação orientada a objetos, programação funcional e programação reativa funcional para prover maior eficiência e escalabilidade.

Arquiteturalmente, para ser uma *API RESTful* e aplicar o protocolo HTTPS, o NestJS não reinventa a roda e por baixo dos panos utiliza outros *frameworks* consagrados no mercado, como o ExpressJS e Fastify, deixando a encargo do desenvolvedor decidir qual irá ser utilizado. Ao fazer isso, essa tecnologia também permite a utilização da ampla gama de bibliotecas auxiliares do NodeJS, adaptando-se facilmente às peculiaridades de cada projeto (MYSLIWIEC, 2023b).

Além disso, de acordo com a comunidade, à nível de organização de código, NestJS inspira-se fortemente no framework *front-end* Angular (PASSOS, 2018). Partindo disso, uma aplicação NestJS consiste em uma organização hierárquica de múltiplos módulos onde cada módulo é encarregado de um fragmento da aplicação, possuindo seus componentes, *middlewares*, filtros, *pipes* e protetores de rota.

Por fim, a tecnologia dispõe de sua própria interface de linha de comando, onde é possível inicializar a estrutura base do projeto além de também incluir de maneira simplificada novos módulos à aplicação (MYSLIWIEC, 2023a).

Figura 5 – Estrutura base de um projeto NestJS.



Fonte: Autor

3.6 PostgreSQL

PostgreSQL trata-se de um sistema gerenciador de banco de dados de código aberto, muito famoso e utilizado amplamente no âmbito empresarial. Esse sistema caracteriza-se principalmente por sua arquitetura robusta, fácil instalação, uso prático e possibilidade de utilização de extensões (POSTGRESQL, 2023).

"O PostgreSQL conquistou uma forte reputação por sua arquitetura comprovada, confiabilidade, integridade de dados, conjunto robusto de recursos, extensibilidade e a dedicação da comunidade de código aberto por trás do software para fornecer consistentemente soluções inovadoras e de alto desempenho. O PostgreSQL é executado em todos os principais sistemas operacionais, é compatível com ACID desde 2001 e possui complementos poderosos, como o popular extensor de banco de dados geoespacial PostGIS."

PostgreSQL (2023).

Em respeito a sua arquitetura, PostgreSQL estrutura-se de maneira simplória, possuindo uma memória compartilhada, processos de *background* e sistema de diretório de dados. Em um fluxo usual, o cliente realiza uma solicitação ao servidor, que então processa os dados utilizando buffers compartilhados e processos em segundo plano (KINSTA, 2023).

REFERÊNCIAS

- BAHATI, C. W. e H. **Overview of Blocking vs Non-Blocking**. 2022. Disponível em: <<https://nodejs.dev/en/learn/overview-of-blocking-vs-non-blocking/>>. Acesso em: 4 de Junho de 2023. Citado na página 5.
- EXCHANGE, S. **2023 Developer Survey**. 2023. Disponível em: <<https://survey.stackoverflow.co/2023/>>. Acesso em: 10 de Junho de 2023. Citado na página 6.
- FOUNDATION, O. **About NodeJS**. 2023. Disponível em: <<https://nodejs.dev/en/about/>>. Acesso em: 4 de Junho de 2023. Citado na página 5.
- GOLDBERG, J. **Learning TypeScript**: Enhance your web development skills using type-safe javascript. [S.l.]: O'Reilly Media, Incorporated, 2022. ISBN 1098110331. Citado na página 4.
- KINSTA. **O que é PostgreSQL?** 2023. Disponível em: <<https://kinsta.com/pt/base-de-conhecimento/o-que-e-postgresql/>>. Acesso em: 13 de Junho de 2023. Citado na página 8.
- MALAVASI, A. **Afinal, Javascript e ECMAScript são a mesma coisa?** 2017. Disponível em: <<https://medium.com/trainingcenter/afinal-javascript-e-ecmascript-s~ao-a-mesma-coisa-498374abbc47>>. Acesso em: 25 de Maio de 2023. Citado na página 4.
- MICROSOFT. **The TypeScript Handbook**. 2023. Disponível em: <<https://www.typescriptlang.org/docs/handbook/intro.html#about-this-handbook>>. Acesso em: 31 de Maio de 2023. Citado na página 4.
- MICROSOFT. **TypeScript: JavaScript with syntax for types**. 2023. Disponível em: <<https://www.typescriptlang.org/>>. Acesso em: 28 de Maio de 2023. Citado na página 4.
- MOZILLA. **JavaScript**. 2023. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/JavaScript>>. Acesso em: 24 de Maio de 2023. Citado na página 3.
- MYSLIWIEC, K. **NestJS CLI**. 2023. Disponível em: <<https://docs.nestjs.com/cli/overview>>. Acesso em: 19 de Junho de 2023. Citado na página 7.
- MYSLIWIEC, K. **NestJS Introduction**. 2023. Disponível em: <<https://docs.nestjs.com/>>. Acesso em: 19 de Junho de 2023. Citado na página 7.
- NPM, I. **About npm**. 2022. Disponível em: <<https://docs.npmjs.com/about-npm>>. Acesso em: 5 de Junho de 2023. Citado na página 5.
- PASSOS, C. K. N. **Nest.js — backend com cara de Angular sobre NodeJs**. 2018. Disponível em: <<https://medium.com/codigorefinado/nest-js-typescript-pra-node-js-e-express-e5042b77cae0>>. Acesso em: 19 de Junho de 2023. Citado na página 7.
- POSTGRESQL. **About PostgreSQL**. 2023. Disponível em: <<https://www.postgresql.org/about/>>. Acesso em: 12 de Junho de 2023. Citado na página 8.

QAWWAS, O. Modular, Maintainable Front-end React Code With Separation of Concerns. 2022. Disponível em: <<https://engineering.teknasyon.com/separation-of-concerns-on-the-front-end-with-react-fd5d4afcc298>>. Acesso em: 9 de Junho de 2023. Citado na página 6.