STARSEEKER: PROJETO DE DESENVOLVIMENTO DE UM APLICATIVO DE EVENTOS ASTRONÔMICOS E LANÇAMENTOS ESPACIAIS

Gabriel Périco[1]
João Paulo Gregolon Paludo[1]
Franciele Petry[2]
Roberson Junior Fernandes Alves[3]

Resumo

Este artigo tem como principal objetivo demonstrar as atividades realizadas, bem como os resultados alcançados com o processo de desenvolvimento de um aplicativo desktop de eventos astronômicos e lançamentos espaciais. Durante o andamento do projeto vários aspectos foram avaliados a fim de produzir um software de qualidade e que desempenhasse funções realmente utilizáveis por quem tem interesse nas áreas da astronomia e da exploração espacial. A construção desse aplicativo fez uso de ferramentas como a linguagem de programação TypeScript, framework para desenvolvimento de aplicações desktop Electron, framework React Next.js, armazenamento de dados em localStorage, Notion, Figma, Kanban, entre outras.

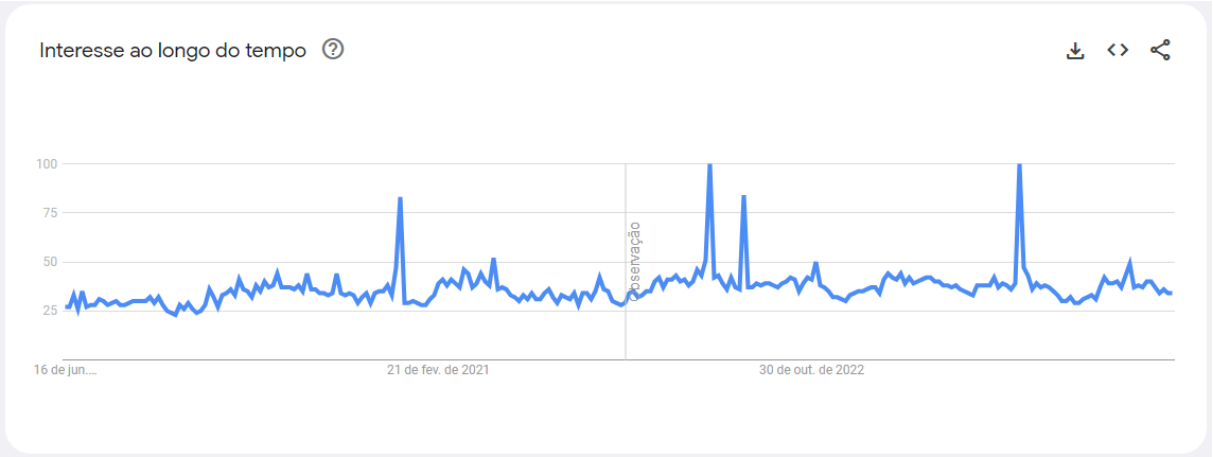
Palavras-chaves: Eventos astronômicos; Aplicação desktop; Next.js; Electron.

[1] Discentes do Curso de Ciência da Computação
Unoesc - Campus de São Miguel do Oeste
Rua Oiapoc, 211. São Miguel do Oeste-SC
gabrielperico2014@gmail.com; joaopgpaludo@gmail.com.
[2] Mestre em Informática
Docente do Curso de Ciência da Computação
Unoesc - Campus de São Miguel do Oeste
Rua Oiapoc, 211. São Miguel do Oeste-SC.
franciele.petry@unoesc.edu.br
[3] Mestre em Computação Aplicada
Docente do Curso de Ciência da Computação
Unoesc - Campus de São Miguel do Oeste
Rua Oiapoc, 211. São Miguel do Oeste-SC.
roberson.alves@unoesc.edu.br

1 INTRODUÇÃO

O interesse pela astronomia e pelas atividades espaciais sempre foi significativo, como mostra a imagem 1. O advento das redes sociais e a facilidade de acesso à informação permitiram que entusiastas e curiosos acompanhassem de perto descobertas científicas, eventos astronômicos e lançamentos espaciais. Contudo, a falta de uma plataforma centralizada que agregue essas informações de forma organizada e acessível ainda representa um desafio para muitos interessados. Até mesmo o desenvolvimento científico pode ser prejudicado nessa área pela falta de maneiras de incentivar e facilitar a busca pelo tema, como apontado em artigo da Extra, em 2014.

Imagem 1: Interesse sobre astronomia baseado em buscas no Google no Brasil nos últimos 5 anos



Fonte: Google Trends (2024).

Visando preencher essa lacuna, surge o StarSeeker, um aplicativo inovador dedicado a fornecer uma experiência completa e intuitiva para aqueles que desejam se manter atualizados sobre eventos astronômicos e lançamentos espaciais. Contando com funcionalidades abrangentes que vão desde acompanhamento de datas de eventos como eclipses e chuvas de meteoros, até informações detalhadas sobre missões espaciais e lançamentos de foguetes, o StarSeeker busca servir como um guia para os interessados pelo espaço.

O desenvolvimento do StarSeeker é pautado pela necessidade de facilitar o acesso à informação e promover a educação científica, incentivando o engajamento do público com a astronomia e a exploração espacial. O aplicativo se propõe a ser

uma ferramenta útil tanto para amadores quanto para profissionais, trazendo funcionalidades de notícias ligadas às áreas mencionadas, acompanhamento das fases da lua pela posição geográfica do usuário e data entre outras ferramentas.

A missão do projeto é mudar a forma como as pessoas interagem com o céu noturno e com as missões espaciais, tornando as experiências mais acessíveis, centralizadas e inspiradoras.

2 DESENVOLVIMENTO

A parte do desenvolvimento apresenta informações sobre o interesse do público na astronomia e em eventos astronômicos e a relevância dessas áreas, assim como trata das principais tecnologias utilizadas no desenvolvimento do software, e também os resultados obtidos.

2.1 Referencial teórico

Como apontado por Peixoto e Kleinke (2016), grande parte do interesse de alunos do ensino médio na parte da astronomia se direciona a temas ligados à ficção científica, por serem temáticas mais presentes em conteúdos mais expostos ao grande público, como filmes, séries de televisão etc.

Outro fator que pode ser visto como empecilho para o desenvolvimento científico da astronomia e até mesmo da busca por conteúdos relacionados é exemplificado por Machado et al. (2018), que buscaram despertar o interesse pela astronomia em escolas, e é a ausência de contato com a astronomia e a falta de acesso à informação e equipamentos astronômicos, onde muitos dos envolvidos não tinham conhecimento algum sobre a área, mas demonstraram grande interesse.

Além disso, Arruda, Zapparoli e Passos (2019) trazem que páginas em redes sociais, como o Facebook, podem dar suporte à comunidade de interessados por astronomia, tanto no aspecto da aprendizagem científica, quanto no que diz respeito a apreciação da astrofotografia e interesses pessoais em eventos astronômicos.

Dessa forma, o StarSeeker poderia desempenhar um importante papel, ao servir como centralizador de notícias e fonte de acesso à informação sobre astronomia, podendo aproximar as pessoas de eventos reais ligados à área, permitindo que a ciência e a exploração espacial que realmente acontecem ocupem um lugar junto à ficção científica no que diz respeito ao conhecimento do público em

geral, além de possibilitar que as pessoas tenham contato com a área, podendo despertar nelas o interesse pelo assunto.

2.2 Materiais e métodos

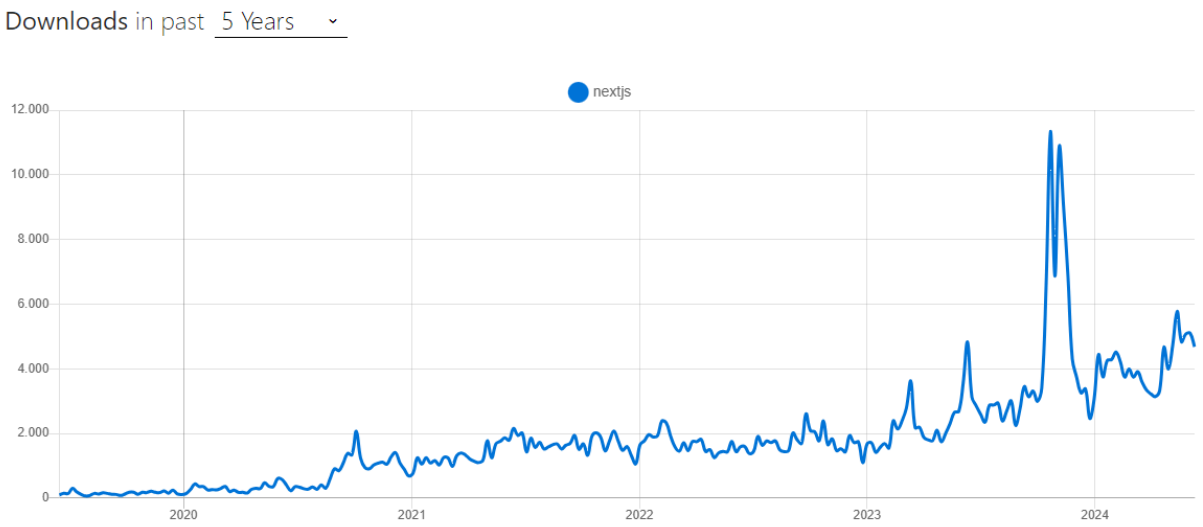
O presente tópico discorre sobre o levantamento de requisitos para o desenvolvimento das atividades propostas e da aplicação em si.

Para a construção do app, foi utilizado o editor de código-fonte Visual Studio Code, onde a linguagem de programação escolhida foi o TypeScript, levando em consideração que é uma linguagem fortemente tipada implementada sobre o JavaScript, sendo muito versátil, capaz de rodar em qualquer ambiente que execute JavaScript, como é o caso do Electron, e confiável, permitindo identificar erros ainda no editor, como especificado por Kinsta (2023) e também pela Microsoft na própria documentação da linguagem.

Visto que a ideia era de desenvolver uma aplicação desktop, foi utilizado o framework Electron, um framework de código aberto que assume grande parte das responsabilidades sobre as partes mais complexas de desenvolver uma aplicação desktop permitindo que os desenvolvedores foquem na sua aplicação, como descrito na documentação do framework. Além disso, a base em Chromium traz a possibilidade de utilizar o Electron em conjunto com React, Next.js, TypeScript e Sass, recursos usados no projeto, também favoreceram o uso do framework, bem como algumas vantagens apontadas por Borges, como o fato de permitir desenvolver aplicações multi-plataforma, e a não necessidade de conhecer novas linguagens além do que se usa no desenvolvimento web para criar a aplicação.

O Next.js foi escolhido por ser um framework de popularidade ascendente, e que teve um grande crescimento no final de 2023, conforme mostra o gráfico de histórico de downloads do Next.js do npm trends (imagem 2).

Imagem 2: Downloads do Next.js nos últimos 5 anos.

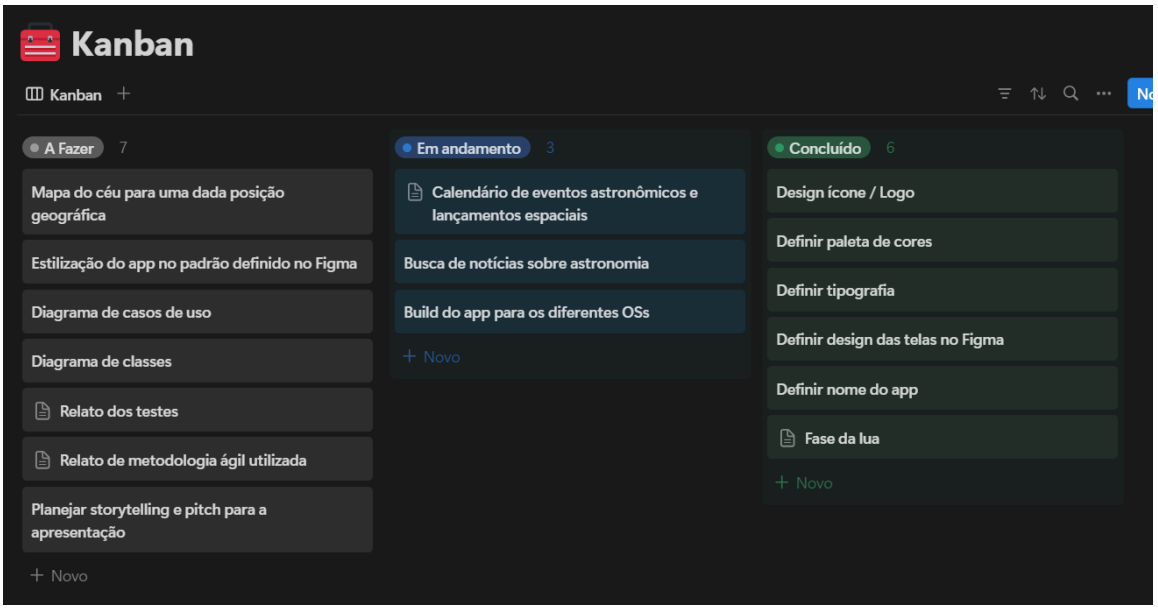


Fonte: npm trends (2024).

Além disso, um recurso do framework Next.js que favoreceu sua utilização foi o SSR, ou “Server Side Rendering”, que, como explica Nascimento (2022), renderiza a página a ser visualizada pelo usuário no servidor, eximindo o navegador, ou no nosso caso, o Electron, de renderizar a interface visual da aplicação, tendo apenas que exibir a página já renderizada pelo Next.js.

No âmbito de metodologias ágeis, a ferramenta Notion foi utilizada para delegar as atividades que seriam desempenhadas por cada integrante da equipe, e também para ordenar a realização das mesmas. A separação foi feita no modelo Kanban, onde as tarefas foram divididas nas categorias “A fazer”, “Em andamento” e “Concluído”, onde as tarefas avançavam para as colunas seguintes conforme eram executadas, conforme a imagem em sequência demonstra.

Imagem 3: Tarefas divididas no Notion.

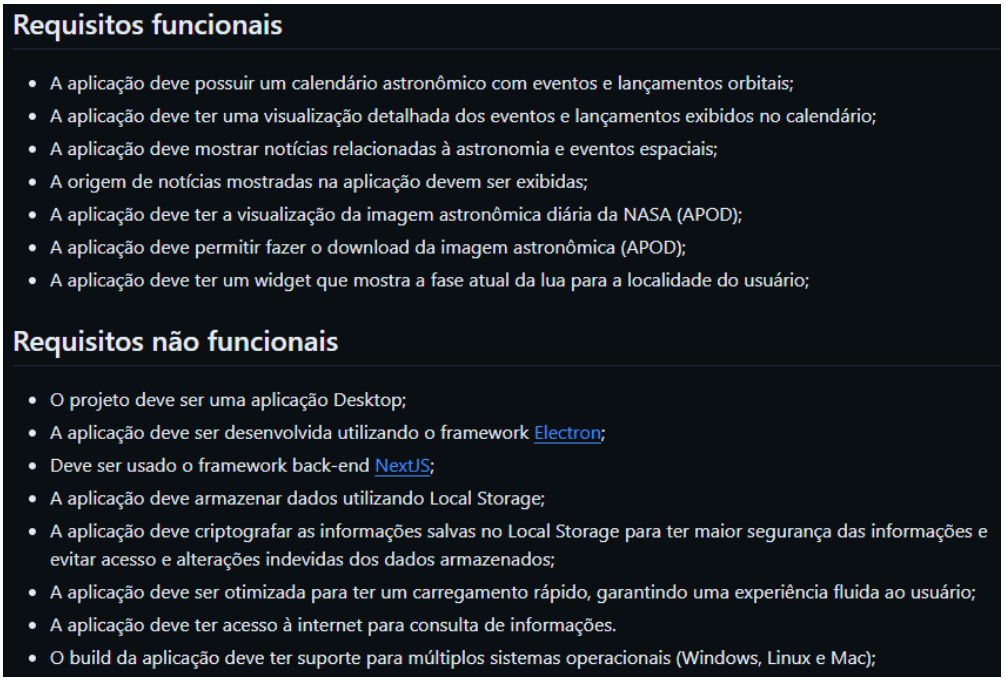


Fonte: Os autores (2024).

2.2.1 Requisitos

O passo inicial para o desenvolvimento da aplicação foi o levantamento daquilo que o software deveria ter como recursos e funções e como ele deveria executar isso, isto é, a definição dos requisitos do projeto, que foi feita embasando-se em dados e informações disponíveis na internet, e resultou nos requisitos funcionais e não funcionais iniciais, demonstrados na imagem 4.

Imagem 4: Requisitos do projeto.



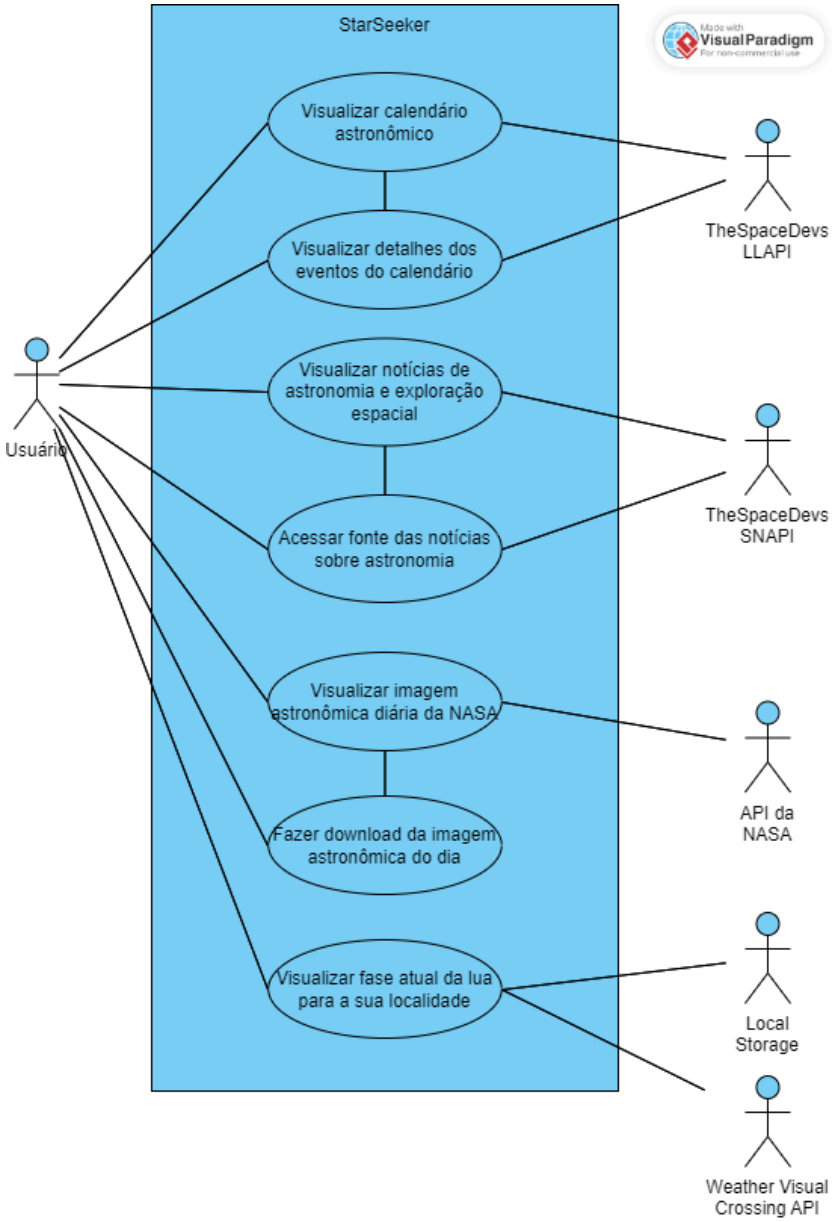
Fonte: Os autores (2024).

2.2.2 Diagramas

As informações obtidas nas primeiras pesquisas foram utilizadas na criação dos diagramas responsáveis pela descrição do funcionamento do sistema objetivando a orientação do processo de desenvolvimento e a redução de ambiguidades que poderiam surgir durante o progresso do projeto, transmitindo a ideia do funcionamento da aplicação de forma clara e objetiva.

O primeiro diagrama apresentado abaixo é o diagrama de casos de uso da aplicação, que resume os usuários do sistema (também chamados de atores) e as interações que eles podem ter com o sistema.

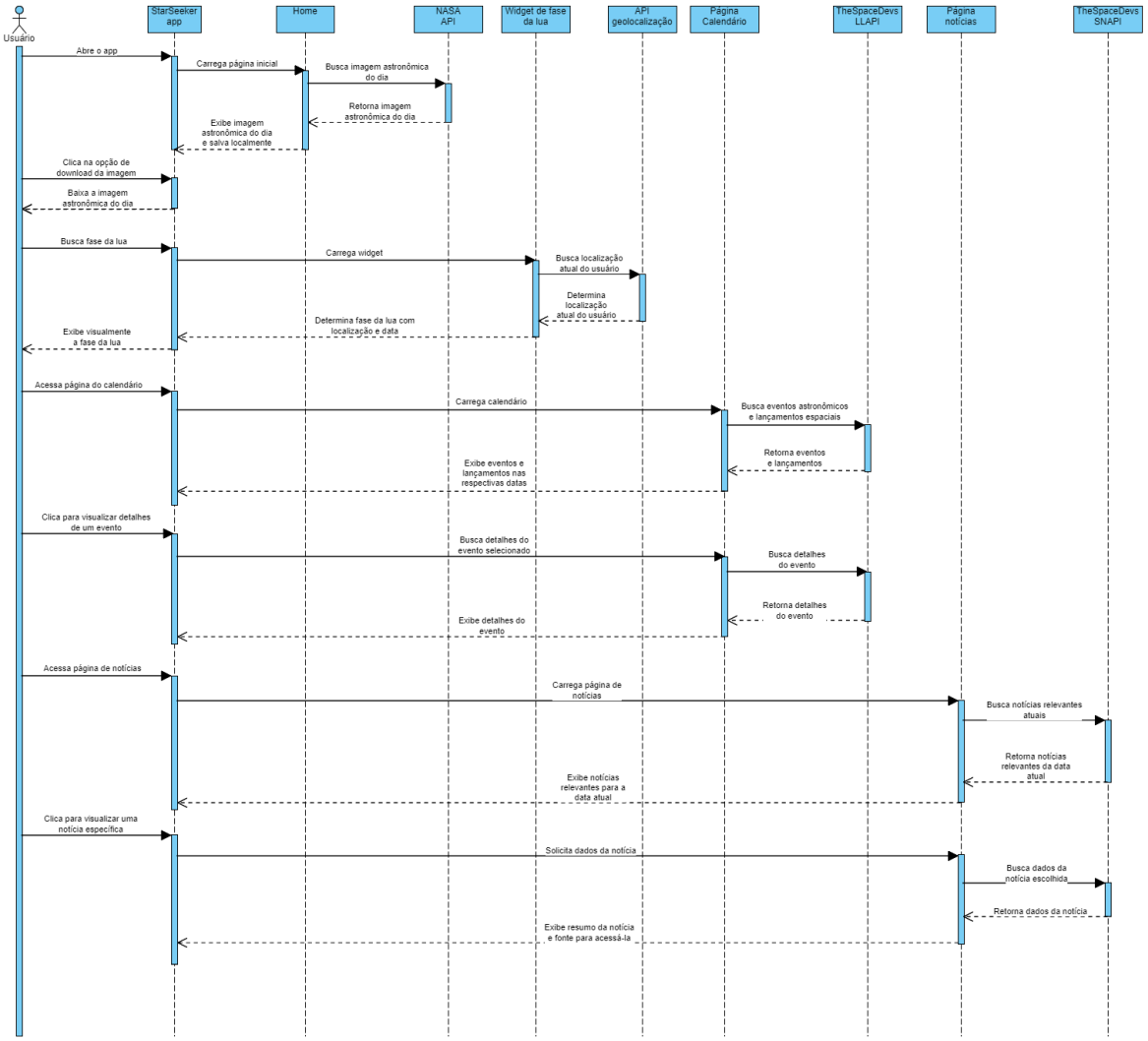
Imagem 5: Diagrama de casos de uso.



Fonte: Os autores (2024).

Além deste, foi desenvolvido também um diagrama de sequência, ilustrando a ordem com que as mensagens são transmitidas entre os objetos participantes das possíveis interações do usuário com a aplicação, desde a interação do usuário com a interface, passando pelos componentes que compõem a aplicação desktop até as APIs consultadas.

Imagem 6: Diagrama de sequência.



Fonte: Os autores (2024).

2.2.3 Testes

Como definido pela IEEE(1981), a garantia de qualidade consiste em um padrão de ações necessárias para assegurar de maneira confiante que um produto ou item esteja em conformidade com os requisitos técnicos estabelecidos. Para tal,

segundo Bastos et al., existem três dimensões a considerar a fim de garantir a qualidade: confiança, funcionalidade e performance. A confiança reflete o quanto o sistema é resistente à falhas, a funcionalidade remete ao funcionamento do sistema estar de acordo com o esperado, baseando-se nos requisitos definidos, e a performance trata do tempo de resposta do sistema, avaliando se é aceitável, mesmo quando colocado em situações de alto volume de processamento.

Os testes de software têm papel fundamental na garantia da qualidade de software, como cita Muller(2020), visando reduzir o risco de falhas que podem ocorrer durante o funcionamento do software, detectar e corrigir defeitos, além de assegurar que o software atende com o especificado. Entre os tipos de testes de software que podem ser empregados, pode-se citar: testes de caixa branca, testes de caixa preta, teste de regressão, testes de integração, testes de carga, teste de usabilidade etc.

No desenvolvimento do app StarSeeker, foram desenvolvidos testes unitários, representando os testes de caixa branca, onde os testes são desenvolvidos tendo acesso ao código fonte e baseando-se nele, para assegurar o correto funcionamento dos componentes fazendo uso da ferramenta de testes Jest, um poderoso framework de testes JavaScript focado em simplicidade que possui integração com o framework utilizado para desenvolvimento da aplicação, o Next.js. Os testes de unidade desenvolvidos para componentes como o widget de fase da lua, o calendário astronômico, entre outros, têm o objetivo de garantir que cada um desses componentes realize as ações que são de sua responsabilidade e que seu código interno realiza o que é proposto para cada componente de maneira satisfatória.

Além dos testes de caixa branca desenvolvidos, foram feitos também testes de release, onde foram gerados executáveis da aplicação desktop para cada sistema operacional proposto (Windows, Linux e Mac OS) e testados nos respectivos ambientes, com exceção do ambiente Mac OS que não pôde ser testado.

Em complemento aos testes citados, foram feitos também testes de usabilidade com pessoas simulando usuários reais e interagindo diretamente com a aplicação, a fim de verificar se a interface do app é intuitiva e de fácil utilização e se ele de fato faz de maneira eficaz e eficiente aquilo que se propõe.

3 CONCLUSÃO

O presente artigo expôs as etapas de criação do sistema desktop de eventos astronômicos e lançamentos espaciais StarSeeker, considerando o interesse na área em questão e a falta de uma forma de centralização de informações sobre o assunto. Investimentos para desenvolver um sistema como o StarSeeker são válidos, pois permitem a criação de uma ferramenta que possibilita levar os usuários ao encontro da informação sobre a temática de forma fácil e prática.

A conclusão dos objetivos inicialmente propostos foi possível, uma vez que o desenvolvimento se deu de forma eficaz, contando com a organização dos envolvidos com o uso de métodos ágeis e controle de progresso das tarefas, possibilitando a implementação das funcionalidades desejadas.

Havia ainda ideias de outras funcionalidades relacionadas que poderiam ser acrescentadas ao app, além de melhorias em alguns quesitos, como armazenamento de informações de usuário permitindo uma customização ainda maior do sistema.

STARSEEKER: PROJECT TO DEVELOP AN ASTRONOMICAL EVENTS AND SPACE LAUNCHES APPLICATION

Abstract

This article's main objective is to demonstrate the activities carried out, as well as the results obtained in the process of development of an astronomical events and space launches application. During the progress of the project, several aspects were evaluated in order to produce a software of quality and that features actually usable functions for those who are interested in astronomy and space exploration. The construction of this application used tools such as the TypeScript programming language, desktop application development framework Electron, React framework Next.js, localStorage for data storing, Notion, Figma, Kanban and others.

Keywords: Astronomical events; Desktop application; Next.js; Electron.

REFERÊNCIAS

ARRUDA, S. DE M.; ZAPPAROLI, F. V. D.; PASSOS, M. M. **Aprendizagem de Astronomia em grupos do Facebook.** Caderno Brasileiro de Ensino de Física, v. 36, n. 2, p. 383–413, ago. 2019.

BASTOS, Anderson; et. al. **Base de conhecimento em teste de software.** São Paulo: Martins Fontes, 2007.

BORGES, J. **Criação de APP Desktop usando Electron JS.** Disponível em: <https://www.dio.me/articles/criacao-de-app-desktop-usando-electron-js>. Acesso em: 22 jun. 2024.

Electron | Build cross-platform desktop apps with JavaScript, HTML, and CSS. Disponível em: <https://www.electronjs.org/>. Acesso em: 22 jun. 2024.

Estudantes destacam dificuldades da astronomia no Brasil. Disponível em: <https://extra.globo.com/noticias/educacao/vida-de-calouro/estudantes-destacam-dificuldades-da-astronomia-no-brasil-14901498.html>. Acesso em: 21 jun. 2024.

GOOGLE TRENDS. **Google Trends.** Disponível em: <https://trends.google.com/trends/explore?cat=435&date=today%205-y&geo=BR&hl=pt-BR>. Acesso em: 21 jun. 2024.

IEEE - Institute of Electrical and Electronics Engineers. **730-1981** - IEEE Standard for Software Quality Assurance Plans. Nova Iorque, NY, 1981. 12 p.

Jest. **Delightful JavaScript Testing**. Disponível em: <https://jestjs.io/pt-BR/>. Acesso em: 03 jul. 2024.

MACHADO, M. M. et al. **Astronomia na Escola: Despertando o Interesse pela Ciência na Fronteira Oeste do Rio Grande do Sul.** Revista Extensão em Foco, v. 16, p. 55–73, 2018.

MICROSOFT. **TypeScript - JavaScript that scales.** Disponível em: <https://www.typescriptlang.org/>. Acesso em: 21 jun. 2024.

NASCIMENTO F. **NextJS: por que usar? | Alura Cursos Online.** 18 out. 2022. Disponível em: <https://www.alura.com.br/artigos/next-js-vantagens>. Acesso em: 22 jun. 2024.

nextjs | npm trends. Disponível em: <https://npmtrends.com/nextjs>. Acesso em: 22 jun. 2024.

O que é Teste de Software? Por que é necessário? | CWI Software. Disponível em: <https://cwi.com.br/blog/o-que-e-teste-de-software-por-que-e-necessario/>. Acesso em: 03 jul. 2024.

O que é TypeScript? Um Guia Abrangente. Disponível em: <https://kinsta.com/pt/base-de-conhecimento/o-que-e-typescript/>. Acesso em: 21 jun. 2024.

PEIXOTO, D. E; KLEINKE, M. U. **EXPECTATIVAS DE ESTUDANTES SOBRE A ASTRONOMIA NO ENSINO MÉDIO.** Revista Latino-Americana de Educação em Astronomia - RELEA, v. 22, p. 21–34, 2016.