**UNIVERSIDADE DO OESTE DE SANTA CATARINA – UNOESC**

ADRIEL MATHEUS GRAUPP

GUSTAVO BERWIAN

**APLICATIVO FLUTTER**

**SERÁ QUE CHOVE?**

São Miguel do Oeste

2023

ADRIEL MATHEUS GRAUPP

GUSTAVO BERWIAN

**APLICATIVO FLUTTER**

**SERÁ QUE CHOVE?**

Trabalho apresentado ao curso de Ciência da Computação como requisito para obtenção de nota para o componente curricular de Engenharia de Software II.

Orientadora: Franciele Carla Petry,

São Miguel do Oeste

2023

**SUMÁRIO:**

Introdução3

Será Que Chove?4

Diagramas4

Metodologia Ágil6

Storytelling/Pitch......................................................................................................................7

Testes de Software....................................................................................................................8

**Conclusão...............................................................................................................................14**

**Referências............................................................................................................................15**

**INTRODUÇÃO**

O presente trabalho é sobre o desenvolvimento do Será Que Chove, trata-se de um aplicativo para obter informações do clima, como por exemplo: temperatura, previsão de chuva, clima atual e para as próximas horas, sendo possível acessar a temperatura de qualquer cidade brasileira. O sistema tem como objetivo sanar suas dúvidas ao sair de casa, será que preciso levar casaco ou guarda chuva?

O texto apresenta informações de suma importância para o desenvolvimento de um aplicativo. Em especifico nesse trabalho iremos abordar o conteúdo mais relacionado a parte de engenharia de software, mostraremos diagramas que possam auxiliar no desenvolvimento, como por exemplo: diagrama de classes e diagrama de casos de uso, além do mais apresentaremos tipos de testes que foram implementados em nosso App, métodos ágeis utilizados para organizar o desenvolvimento do projeto e por fim como foi a criação do Storyetelling e do Pitch. Tais conhecimentos poderão ser aplicados para o desenvolvimento de qualquer sistema. Apresentaremos ao leitor principais objetivos e funções presentes no sistema, o artigo traz imagens, diagramas e exemplos que podem auxiliar o leitor no entendimento do conteúdo, mostraremos na prática a criação e planejamento de projeto.

**SERÁ QUE CHOVE?**

A previsão de chuva é uma área de grande interesse para diversos setores, como agricultura, turismo e planejamento urbano. No entanto, as previsões meteorológicas nem sempre são precisas, o que pode levar a inconveniências e prejuízos. O aplicativo Será Que Chove? busca solucionar esse problema ao oferecer aos usuários uma previsão mais precisa e confiável, contribuindo para uma melhor tomada de decisões.

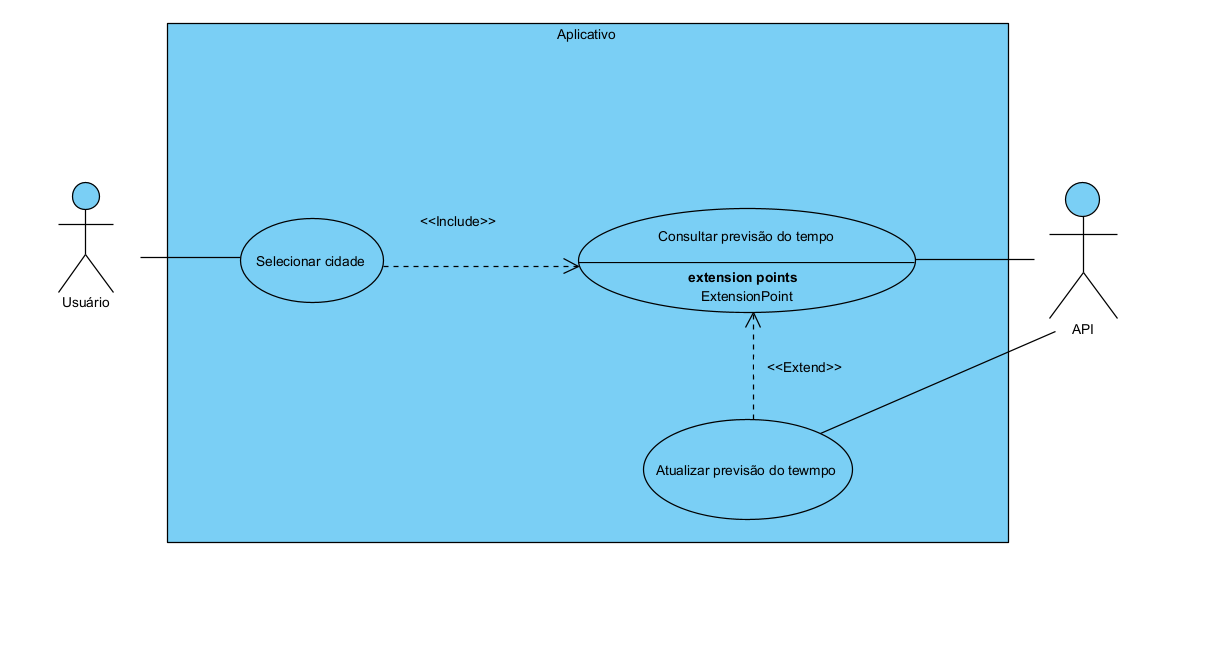
Para o desenvolvimento do aplicativo, foram coletados dados meteorológicos de fontes confiáveis, no nosso caso a API AccuWeather. Esses dados foram utilizados para treinar o modelo de previsão, permitindo que o aplicativo ofereça informações atualizadas e precisas.

**DIAGRAMAS**

Diagramas na engenharia de software são ferramentas visuais importantes para comunicar, analisar, projetar e documentar sistemas complexos. Eles facilitam a compreensão, simplificam a complexidade, auxiliam na tomada de decisões e melhoram a colaboração entre as equipes. Além disso, os diagramas ajudam no gerenciamento de projetos e estimativas de tempo. Em resumo, os diagramas desempenham um papel fundamental no desenvolvimento de software de qualidade. Neste trabalho utilizamos dois tipos de diagramas, diagrama de casos de uso e diagrama de classes.

**Casos de Uso:**

Na Linguagem de modelagem unificada (UML), o diagrama de caso de uso resume os detalhes dos usuários do seu sistema (também conhecidos como atores) e as interações deles com o sistema. Na criação de um, use-se um conjunto de símbolos e conectores especializados. O diagrama de caso de uso tem como principais finalidade ajudar sua equipe a representar e discutir cenários e interações entre usuário e sistema. Logo abaixo você poderá conferir a interação entre usuário e sistema no projeto:

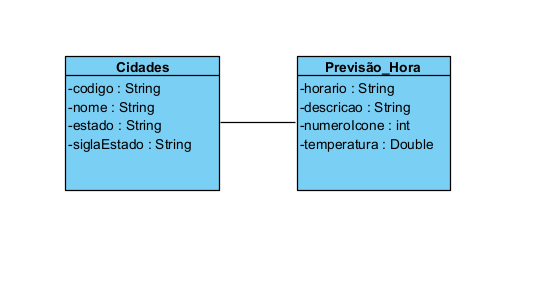


**Diagrama de Classes**

Diagramas de classes servem para mapear de forma clara a estrutura de um determinado sistema ao modelar suas classes, seus atributos, operações e relações entre objetos.

Os diagramas de classes são um tipo de diagrama da estrutura porque descrevem o que deve estar presente no sistema a ser modelado. Podemos afirmar de maneira mais simples que seria um conjunto de objetos com as mesmas características, assim saberemos identificar objetos e agrupá-los, de forma a encontrar suas respectivas classes.

Na imagem logo abaixo pode ser identificada todas as classes, métodos e atributos que foram necessários para o desenvolvimento do projeto.



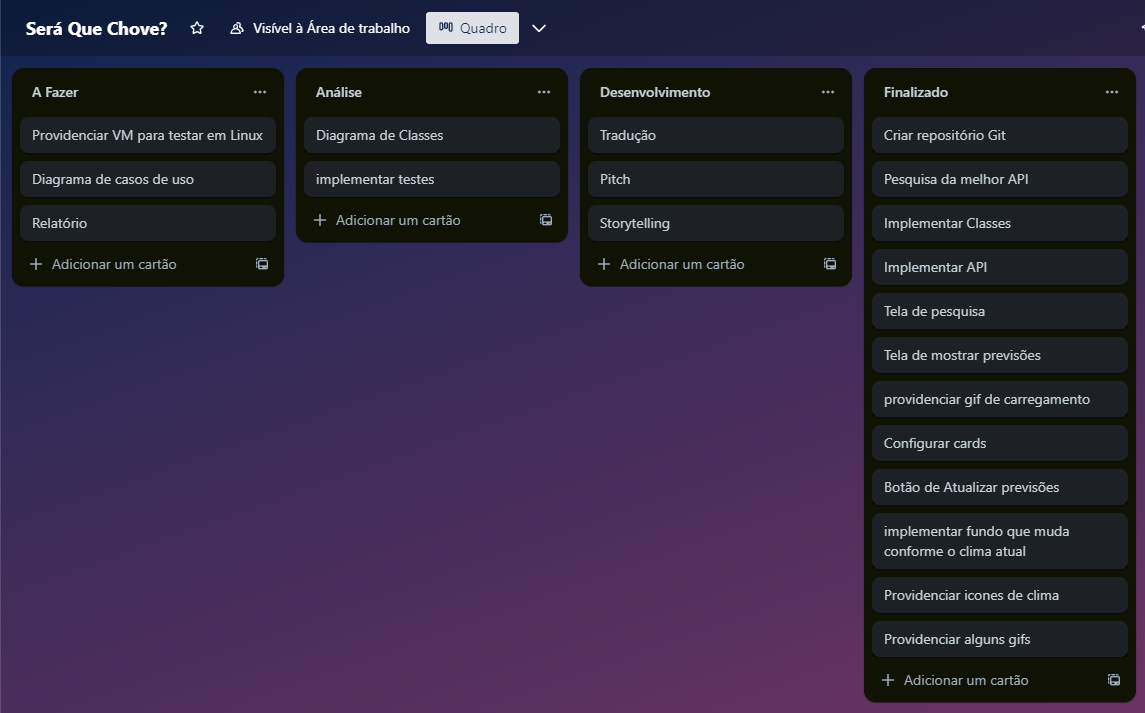
**METODOLOGIA AGIL**

Metodologias ágeis na engenharia de software são abordagens flexíveis e colaborativas que priorizam a entrega incremental de valor ao cliente. Elas promovem a colaboração, a adaptabilidade, a melhoria contínua e permitem uma resposta rápida a mudanças. A importância das metodologias ágeis reside na satisfação do cliente, na capacidade de lidar com mudanças, na entrega mais rápida de valor e na melhoria constante do produto. Existem várias metodologias ágeis amplamente utilizadas, como Scrum, Extreme Programming (XP), Kanban e Lean Software Development.

No caso de nosso trabalho utilizamos a metodologia ágil do Kanban utilizando a plataforma online chamada de Trello.

**KANBAN**

Kanban é uma metodologia ágil de gestão visual que visa otimizar o fluxo de trabalho no desenvolvimento de software. Através de um quadro visual, as tarefas são representadas por cartões que avançam em colunas que representam os estágios do processo. O Kanban proporciona transparência, limita o trabalho em andamento, promove o fluxo contínuo e melhoria contínua. Os benefícios incluem visualização clara do trabalho, melhoria do fluxo, flexibilidade e foco na qualidade. Segue a próxima imagem um print tirado do Trello:



**STORYTELLING / PITCH**

Storytelling é uma técnica de comunicação que envolve a criação e narração de uma história para transmitir uma mensagem ou compartilhar informações. Em vez de simplesmente fornecer fatos ou dados, o storytelling utiliza elementos narrativos, como personagens, enredos e emoções, para engajar o público e transmitir uma mensagem de forma memorável e impactante. É uma maneira eficaz de comunicar ideias, transmitir valores e criar conexões emocionais com o público-alvo.

Pitch, por sua vez, é uma apresentação concisa e persuasiva usada para vender uma ideia, produto ou serviço. É um discurso curto, geralmente de alguns minutos, projetado para captar a atenção do público e despertar seu interesse. O objetivo de um pitch é fornecer uma visão geral convincente do que está sendo oferecido e motivar o público a se envolver, seja investindo, apoiando ou adquirindo o que está sendo apresentado.

A combinação de storytelling e pitch é uma estratégia poderosa para capturar a atenção do público, criar conexões emocionais e persuadir os ouvintes a se envolver ou apoiar o que está sendo apresentado.

**Storytelling do Será Que Chove?**

Em uma pequena cidade chamada Tangamandapio, situada em meio a belas paisagens montanhosas, os moradores viviam sob um céu imprevisível. O clima oscilava entre ensolarado, chuvoso, nublado e tempestuoso de forma quase aleatória, deixando-os despreparados para enfrentar o dia.

Entre esses moradores estava Jaiminho, um jovem sonhador que adorava explorar as belezas naturais de Tangamandapio. No entanto, sua paixão pela aventura frequentemente era frustrada por mudanças climáticas repentinas. Ele se cansou de levar guarda-chuvas desnecessários em dias ensolarados e de ser pego desprevenido em tempestades.

Então para resolver os problemas como o de Jaiminho, apresentamos o aplicativo 'Será que Chove?', a solução definitiva para suas preocupações com o clima!

Será Que Chove? é o seu novo melhor amigo para se preparar para qualquer condição climática. Com uma interface simples e amigável, você poderá obter informações precisas sobre o tempo em qualquer lugar do mundo com apenas alguns cliques!

Basta digitar sua localização, e o 'Será que Chove' trará todas as informações climáticas que você precisa. Temperatura atual, temperatura máxima e mínima, previsão atual e para as próximas horas, isso e muito mais! Tudo ao seu alcance na tela do seu computador. Agora, você pode planejar suas atividades ao ar livre com confiança. Seja um piquenique no parque, uma trilha na montanha ou um encontro com os amigos.

Baixe já o 'Será que Chove?' e diga adeus às surpresas desagradáveis do clima.

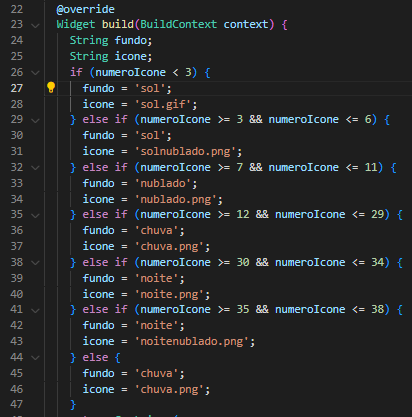
**TESTES DE SOFTWARE**

Os testes de software são essenciais para identificar erros, garantir a qualidade do software, melhorar a usabilidade, facilitar a manutenção e economizar custos. Eles ajudam a encontrar falhas antes que o software seja entregue aos usuários, proporcionando um produto final confiável e satisfatório. Investir em testes de software é crucial para desenvolver um software de alta qualidade.

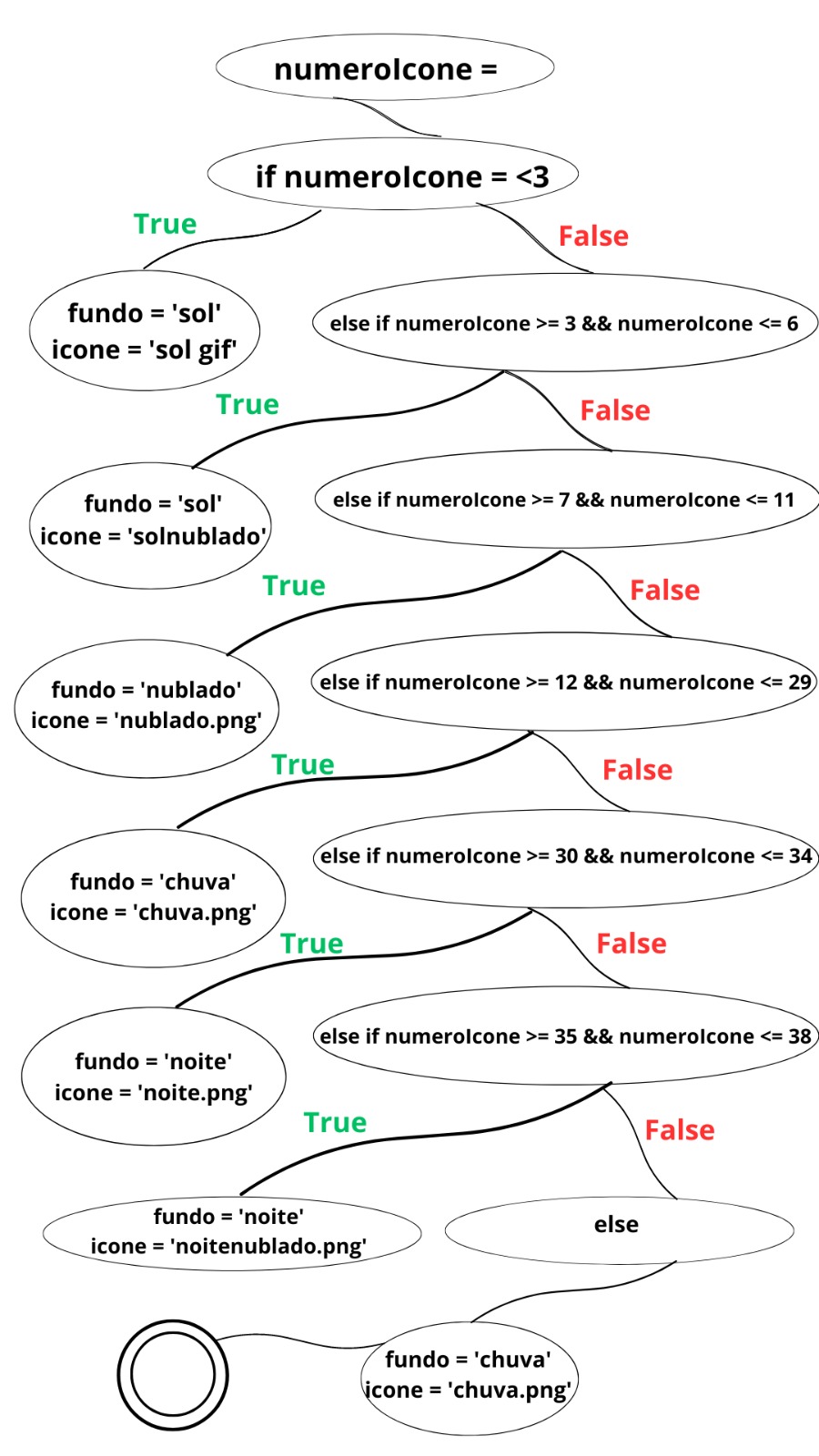
**Teste de Caixa Branca**

O teste de caixa branca é uma abordagem de teste de software em que os testadores têm conhecimento detalhado da estrutura interna do código-fonte. Eles examinam a lógica interna, a estrutura do código e as decisões de programação para projetar e executar casos de teste. O objetivo é avaliar a qualidade e eficiência do código, garantindo cobertura abrangente e identificando possíveis falhas.

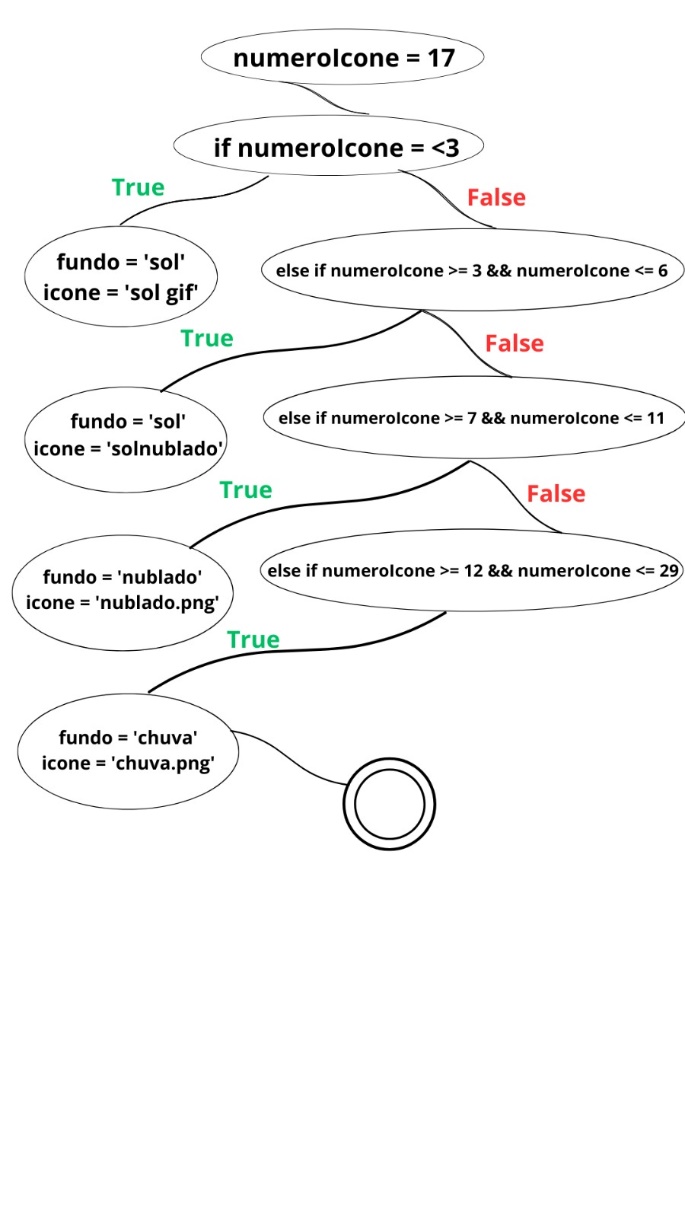
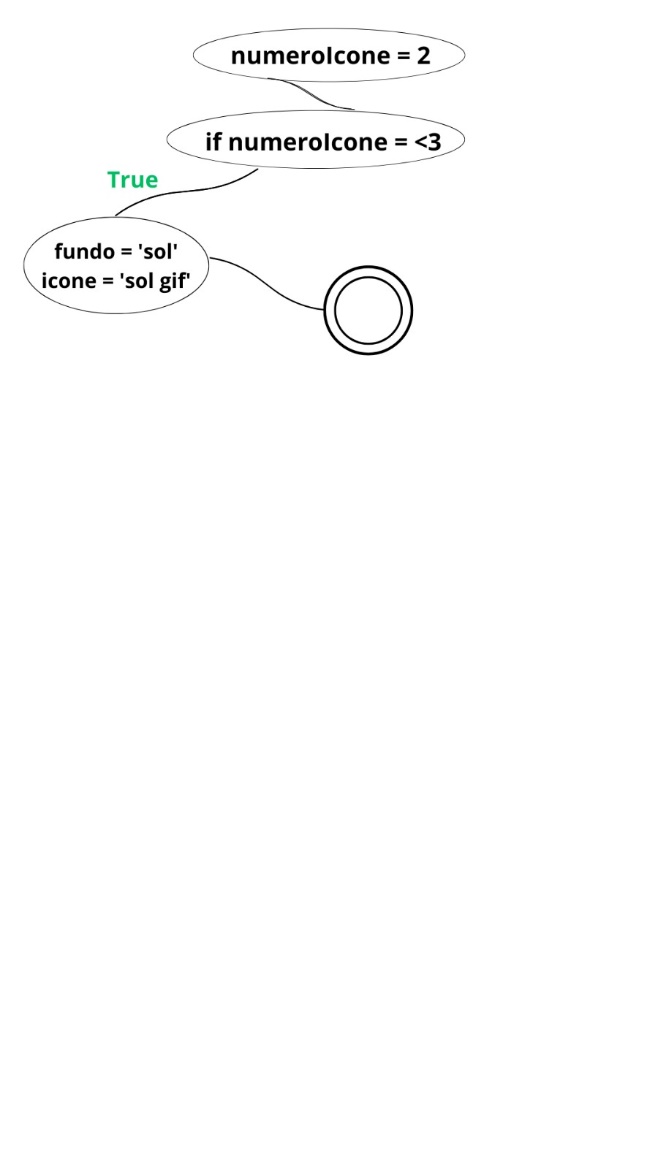
O teste de caixa branca visa explorar os caminhos do sistema, que seriam as sequências possíveis de serem executadas no fluxo de controle do programa. No nosso caso fizemos testes relacionados ao seguinte if responsável por selecionar o ícone e a imagem de fundo do app conforme o clima, no caso conforme o retorno da variável numeroIcone que é alimentada pela API AccuWather.



Então fizemos um teste das possibilidades de execução nesse if conforme o diagrama abaixo:



Ainda temos outros dois exemplos que simulam caso o retorno da variável numeroIcone fosse 17 e 2:

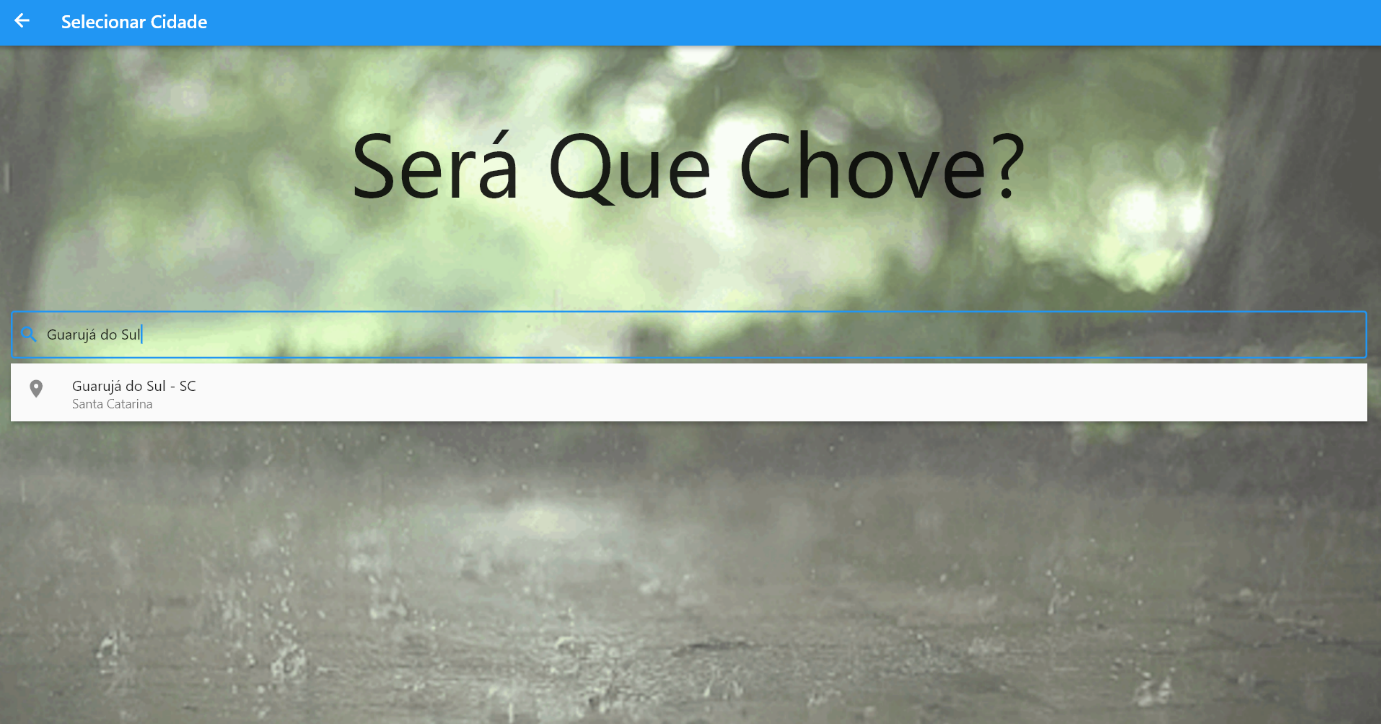
 

**Testes de caixa Preta e Release**

Os testes de caixa preta são uma abordagem de teste de software em que os testadores avaliam a funcionalidade externa do software, sem conhecimento detalhado da estrutura interna do código. O objetivo é verificar se o software atende aos requisitos funcionais, interage corretamente com os inputs e produz os resultados esperados. Essa abordagem é independente do código e baseia-se na especificação dos requisitos e na interface de usuário. Os testes de caixa preta são úteis para validar o comportamento externo do software e garantir sua conformidade com os requisitos, mas não revelam problemas relacionados à implementação interna.

Visa verificar funcionalidade baseado apenas nos dados de entrada e saída, visa verificar resultados finais, não importando a estrutura, estados e comportamento internos. Para fazer este teste disponibilizamos uma release através do GitHub (Git Actions), onde testamos entradas e saídas no sistema:

A cidade inserida está senod retornanda corretamente pelo App:

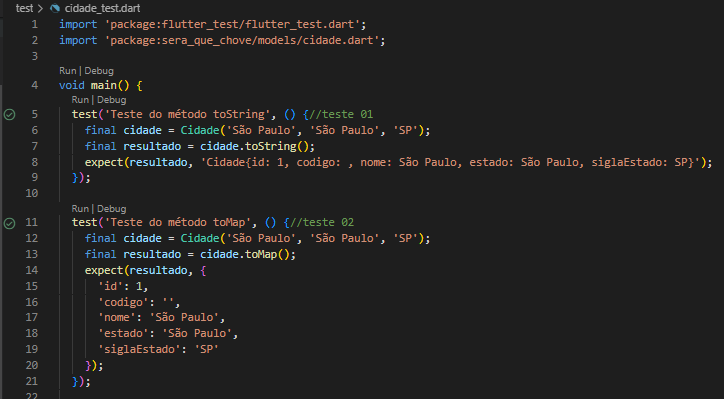




**Testes Unitários**

Os testes unitários são uma prática de teste de software em que unidades individuais de código são testadas de forma isolada. Eles se concentram em verificar se cada unidade de código funciona corretamente e produz os resultados esperados. Os testes são automatizados, executados em um ambiente controlado e garantem uma alta cobertura de código. Os testes unitários ajudam a detectar erros precocemente, melhorar a qualidade do código e facilitar a manutenção.

Foi realiza testes unitários na classe modelo cidade, onde foi testado alguns métodos, como por exemplo o método ToString e o método ToMap:



Teste 01: verifica se o método toString retorna a representação em string correta da instância da classe Cidade.

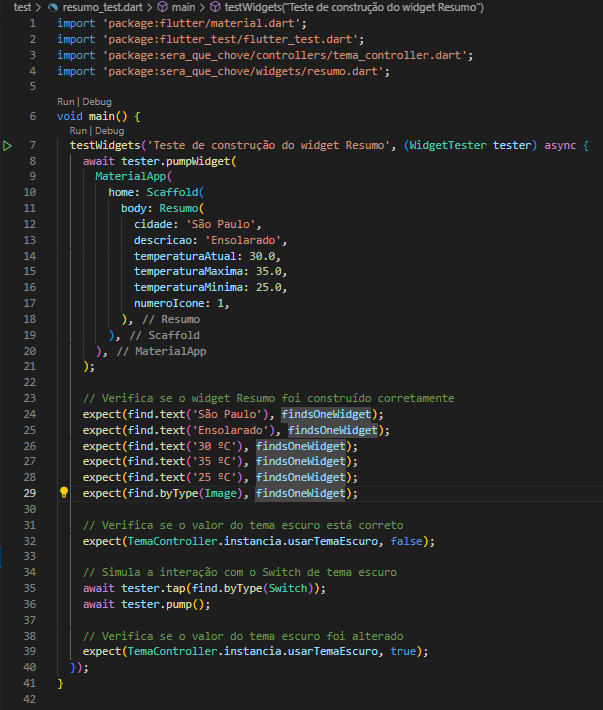
Teste 02: verifica se o método toMap retorna o mapeamento correto dos atributos da instância da classe Cidade para um Map<String, dynamic>.

Também testamos o widgets de resumo, que é o principal widget da tela principal;

Widget de resumo:



Tesste realizado referente ao widget de resumo:



**CONCLUSÃO**

O trabalho apresentado aborda o desenvolvimento do aplicativo "Será Que Chove?", que tem como objetivo fornecer informações precisas sobre o clima, incluindo temperatura, previsão de chuva e clima atual. O aplicativo visa sanar as dúvidas dos usuários ao sair de casa, ajudando-os a tomar decisões sobre a necessidade de levar um casaco ou guarda-chuva.

O texto destaca a importância das informações apresentadas no trabalho para o desenvolvimento de um aplicativo. Ele enfatiza a parte de engenharia de software, incluindo a utilização de diagramas, como o diagrama de casos de uso e o diagrama de classes, que auxiliam no processo de desenvolvimento. Além disso, são mencionados os testes implementados no aplicativo, os métodos ágeis utilizados para organizar o projeto e a criação de um storytelling e pitch para promover o aplicativo.

O trabalho destaca a coleta de dados meteorológicos de fontes confiáveis, como a API AccuWeather, para treinar o modelo de previsão do aplicativo, visando oferecer aos usuários informações atualizadas e precisas. A utilização de metodologias ágeis, como o Kanban, e a plataforma online Trello são mencionadas como parte do processo de desenvolvimento.

Por fim, apresentamos o storytelling do aplicativo, contando a história de Jaiminho, um morador de uma cidade com clima imprevisível, e como o aplicativo "Será Que Chove?" se torna a solução definitiva para suas preocupações com o clima, e também os testes de software, sua importância e o relato dos tipos de testes usados no App.

Em conclusão, o trabalho apresenta o desenvolvimento do aplicativo "Será Que Chove?", enfatizando a importância das informações de engenharia de software, como os diagramas, os métodos ágeis utilizados e a criação de um storytelling e pitch para promover o aplicativo. O aplicativo busca oferecer informações precisas sobre o clima, contribuindo para uma melhor tomada de decisões dos usuários.

**REFERÊNCIAS:**

Diagrama de caso de uso UML: O que é, como fazer e exemplos. LucidChart. Disponível em: https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml. Acesso em: 23 mai. 2022.

Orientações básicas na elaboração de um diagrama de classes. DevMedia. Disponível em: https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224. Acesso em: 1 dez 2022.

OPENAI. ChatGPT. Disponível em: <https://chat.openai.com/>.

‌