

SENAC TECH PORTO ALEGRE

*CURSO TÉCNICO DE DESENVOLVIMENTO DE
SISTEMAS*

PROJETO INTEGRADOR:

— TAMAGOTCHI —

***GRUPO 2: BIBIANA DO BEM, TIAGO CASSOL,
ARTHUR SCHINDLER, VIKTOR DOMINIC RAMOS***

PORTO ALEGRE 2022

SUMÁRIO

1.	INTRODUÇÃO	3
2.	SOBRE O PROJETO.....	5
3.	OBJETIVO GERAL.....	7
4.	OBJETIVO ESPECÍFICO.....	8
5.	MODELO DE CASO DE USO.....	9
6.	DESCRIÇÃO DE CASO DE USO.....	10
7.	REQUISITOS.....	23
8.	DIAGRAMA DE ATIVIDADES.....	24
9.	DIAGRAMA DE ESTADO.....	25
11.	PROTÓTIPO.....	26
12.	CONCLUSÃO.....	28

1. INTRODUÇÃO

Criado em 1996, em uma época onde o acesso à internet não era algo tão facilitado, um bichinho eletrônico japonês que era similar a um chaveiro, trouxe a crianças daquela época e até mesmo de épocas posteriores — com menos peso, já que, ao passar do anos, outros tipos de vídeo games e acesso à internet eram mais recorrentes — um novo vício, uma nova maneira de brincar, onde até mesmo o sofrimento era passível de existir ali, já que com falta de cuidados era possível que seu bichinho morresse. Contudo, não foi apenas a versão lançada pela Bandai que trouxe essa vivência para os seus pequenos jogadores; o bichinho eletrônico era tão popular, que foram feitas até mesmo outras versões criadas por outras empresas, um exemplo disso é sua versão no *iPood Classic* feita em 2008, já com uma cara diferente, já que ele estava como um aplicativo e não mais tão familiar há um “chaveirinho eletrônico”.

Mas apesar de suas novas versões, podemos perceber que o maior impacto do tamagotchi não foi em suas novas versões de aparelho, já que, como muitos, acredito que um dos pontos mais interessantes do tamagotchi é a forma como ele nos é apresentado — Em seu formato oval, seus três botões e sua telinha de *gameboy* — mas sim, em como ele posteriormente trouxe novas versões de jogos para nós, que talvez, para crianças de outras gerações que não tiveram contato ao tamagotchi não traga nada à tona em sua memória, mas para quem teve contato

com o tamagotchi, com certeza sentiu um gosto um pouquinho diferente daquele pequeno brinquedinho; um belo exemplo disso, é o **Pou**, que com sua forma única e já em uma versão de um mundo onde o uso de smartphones e acesso à internet são muito mais recorrentes, também teve grande sucesso com a mesma ideia que antes fora feita no tamagotchi, mas, é claro, com mais alguns detalhes, onde trouxe grande fascínio as crianças, e até mesmo em grupos de faixas etárias distintas.

Pensando nisso, resolvemos fazer nossa versão do tamagotchi, com nosso doce sentimento de infância atrelado ao bichinho eletrônico, trouxemos também nossa versão desse *game*.

2. SOBRE O PROJETO

A ideia surgiu através de um brainstorm realizado pelo grupo, onde uma colega trouxe à tona esse projeto com uma ideia um pouco peculiar e diferente em relação aos outros grupos ainda existentes, mas que com a aprovação do professor Jair Ferraz, pode ser feita sua implementação, de tal modo que seguisse o que foi pedido.

Com algumas dificuldades em relação à dimensão e também em outros aspectos — já que, a ideia de fazer um jogo, por mais tentadora que pudesse ser, também nos trazia grandes desafios, já que há vários aspectos que nunca havíamos trabalhado antes — conseguimos realizar um jogo funcional, que trouxe a ideia que era desejada por nós, pelos motivos ditos anteriormente. Para fazer isso, foi necessário a busca sobre o que por nós era desejado implementar por conteúdo que estava além do que havíamos aprendido em curso, já que se tratava de uma ideia um tanto quanto diferente; O que é ótimo e válido, já que podemos expandir nosso conhecimento além daquilo que nos fora dado, aumentando ainda mais nossa vontade de aprender e de realizar esse projeto.

Em nosso código temos diversos usos de imagens onde elas passam por uma verificação de *true or false*, em sua grande maioria, pois essa verificação é feita de acordo com a energia das barrinhas do tamagotchi. Para que essa verificação de como ele esta seja verdadeira ou falsa, é preciso realizar um cálculo de acordo com suas necessidades, onde nos trás o retorno, de por exemplo, se o tamagotchi está com fome ou não.

Esse retorno, para o jogador, é possível ser analisado não somente por sua barrinha, mas também pela imagem do tamagotchi. Foi implementado no código essas imagens, de acordo com o que está sendo ali feito.

Trabalhamos também com a questão de *game over*, onde, caso o tamagotchi chegue a zerar tudo, ele é dado como morto.

Fora o que acontece dentro do jogo, também foi trabalhada outras questões, como o botão *help*, que caso o usuário não conheça o jogo ou tenha alguma dúvida, é possível ter ali uma breve explicação sobre o que se trata do jogo, temos também a opção de *resetar* o game, caso seja de vontade do usuário, onde temos como retorno o tamagotchi do zero, temos também a opção de *save*, para o jogador salvar seus avanços, entre outros.

Outra implementação que foi feita para trazer esses retornos antes ditos, foram os comando *if* e *else*, onde passa por um check de procedência desses valores, além de toda a implementação de método; como dormir, jogar, limpar, comer...

3 . OBJETIVO GERAL

O jogo em si é simples, já que em *game*, ele tem como acesso algumas funções, que estão atreladas a dois objetivos; não deixar as energias do seu bichinho acabar — que se, caso ocorra, pode deixar seu bichinho doente, triste, ou até mesmo o levar à óbito (por mais que seja o objetivo da *gameplay*, já que é interessante trazer esse desafio de mantê-lo vivo e saudável ao jogador — e o outro objetivo, e como principal, trazer a diversão aos jogadores.

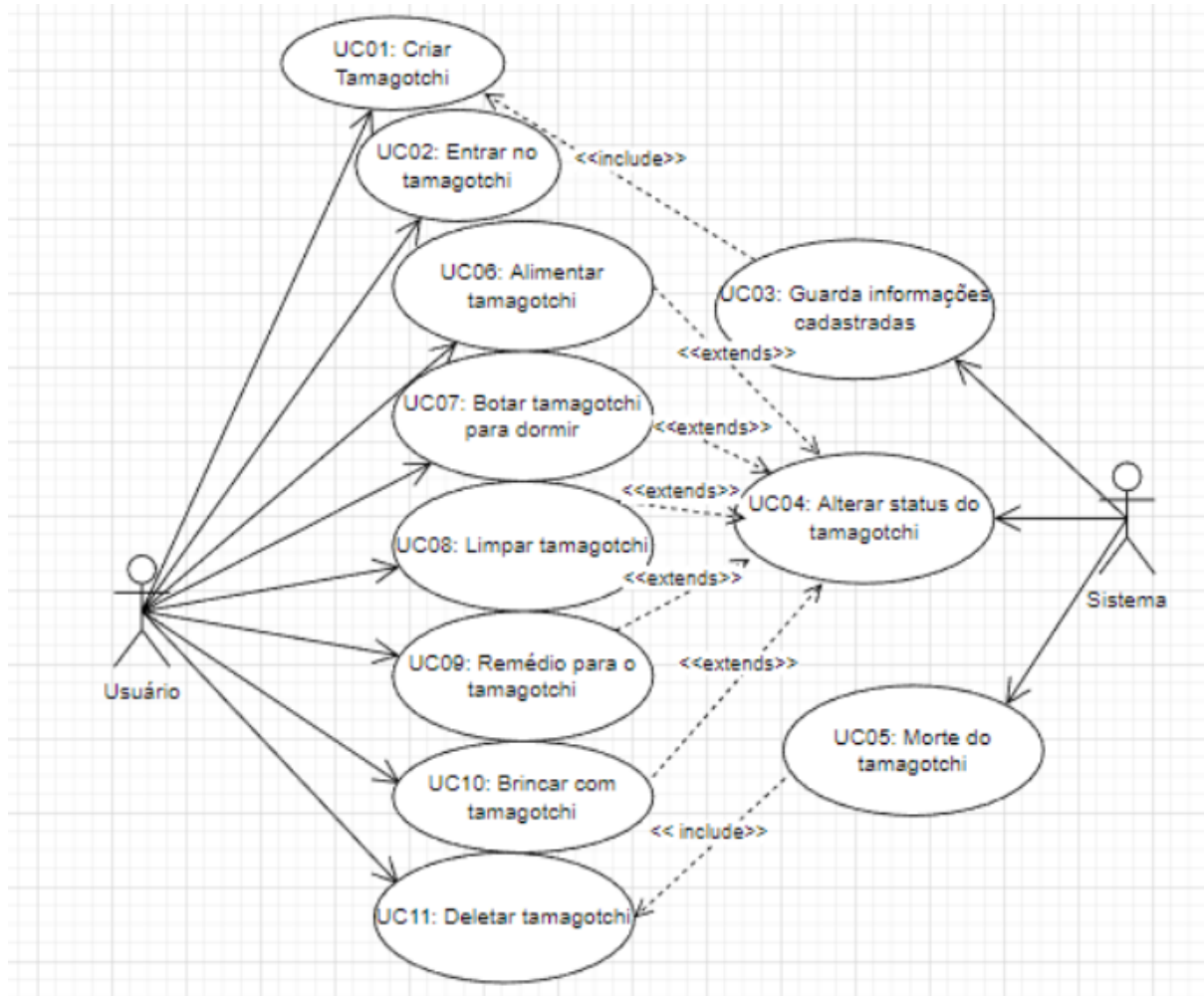
Nosso jogo conta com uma tela de *loading*, onde o usuário pode dar o nome desejado ao seu tamagotchi, e logo em seguida vamos ao jogo, onde já temos acesso ao jogo e suas funções: **Alimentar** o tamagotchi, afim de não deixá-lo morrer de fome; **Brincar**, com a finalidade de não deixá-lo ficar deprimido; **Banhá-lo**, onde seu objetivo é não deixá-lo sujo, e por fim, **descansar**, onde é necessário que o tamagotchi durma, até mesmo porque ele não pode entrar em exaustão.

Com suas *stamina* à mostra ao jogador, o seu objetivo, como dito anteriormente, é não levá-lo a óbito, onde o resultado seria um *game over* para o jogador.

4. OBJETIVO ESPECÍFICO

Realizar um jogo similar à proposta que há no tamagotchi oficial, com a finalidade de trazer isso em uma versão realizada pelo grupo através do console.

5. MODELO CASO DE USO



6. DESCRIÇÃO CASO DE USO

Nome do caso de uso: UC01 - Criar Tamagotchi

Ator: usuário

Descrição: O usuário cria seu tamagotchi para poder cuidar.

Pré-condições: Instalar tamagotchi

Pós-condições: usuário libera as funcionalidades do tamagotchi

Cenário principal:

1.0: Tela de criação

1.1: Usuário dá nome ao tamagotchi

1.2: Usuário Salva tamagotchi criado

Cenário alternativo:

1.0: Tela de criação

1.1: Usuário dá nome invalido ao tamagotchi

1.2: Tamagotchi não é criado

1.3: Pede ao usuário digitar um nome válido

Nome do caso de uso: UC02: entrar no tamagotchi

Ator:Usuário

Descrição: O usuário entra no tamagotchi para cuidar do tamagotchi.

Pré-condições: Ter criado um tamagotchi

Pós-condições: Abrir tela com as funções do tamagotchi

Cenário principal:

1.0: Usuário entra no jogo

1.1: Sistema carrega os status do tamagotchi já salvo

1.2: Entra no menu do tamagotchi

Cenário alternativo:

1.0: Usuário entra no jogo

1.1: Sistema carrega não encontra tamagotchi salvo

1.2: Sistema manda usuário para tela de criação de tamagotchi

Nome do caso de uso: UC03:Guarda informações cadastradas

Ator: Sistema

Descrição: Guarda todas informações e mudanças do usuário

Pré-condições: usuário ter um tamagotchi

Pós-condições: Guardar alterações em um banco de dados

Cenário principal:

1.0: Usuário faz alguma alteração no tamagotchi

1.1: Sistema guarda alteração feita no banco de dados

Cenário alternativo:

1.0: Usuário tenta fazer alteração inválida

1.1: Sistema impede alteração

1.2: Sistema não salva alteração no banco de dados

Nome do caso de uso: UC04: Alterar status do tamagotchi

Ator: Sistema

Descrição: Altera a porcentagem de fome, sono, sujeira, felicidade e saúde de acordo com as ações do usuário e muda o sprite dele de acordo com o status do tamagotchi

Pré-condições: Usuário fazer uma ação

Pós-condições: Modificar porcentagem dos status acordo com a ação do usuário

Cenário principal:

1.0: Usuário faz alguma ação que muda status do tamagotchi

1.1: Sistema responde a ação feita com uma animação

1.2: Sistema altera status do tamagotchi de acordo com a ação

1.3: Sistema muda o sprite de acordo com os status do tamagotchi

1.4: Sistema guarda essas informações

Cenário alternativo:

1.0: Usuário faz alguma ação que tenta mudar status que já estão no máximo ou no mínimo do tamagotchi.

1.1: Sistema impede ação do usuário

1.2: Sistema avisa o usuário

Nome do caso de uso: UC05: Morte do tamagotchi

Ator: Sistema

Descrição: Se os status do tamagotchi ficarem muito baixo, ele vai morrer, assim sendo deletado e usuário não poderá mais usar ele.

Pré-condições: Status do tamagotchi chegarem a um baixo nível

Pós-condições: Tamagotchi será deletado

Cenário principal:

1.0: Usuário não cuida direito de seu tamagotchi

1.1: Status do tamagotchi ficam baixos

1.2: Tamagotchi morre

1.3: Sistema apaga tamagotchi

Cenário alternativo:

1.0: Usuário não quer mais aquele tamagotchi

1.1: Usuário aperta em "reset" para apagar tamagotchi

1.2: Sistema apaga o Tamagotchi

Nome do caso de uso: UC06: Alimentar tamagotchi

Ator: Usuário

Descrição: Alimentar tamagotchi para diminuir seu status de fome

Pré-condições: Barra de comida não estar cheia

Pós-condições: Diminui a fome do tamagotchi

Cenário principal:

1.0: tamagotchi está com medidor de comida abaixo de 100%

1.1: Usuário clica no ícone de comida para alimentar tamagotchi

1.2: Sistema diminui a fome do tamagotchi

Cenário alternativo:

1.0: tamagotchi está com medidor de comida em 100%

1.1: Usuário clica no ícone de comida para alimentar tamagotchi

1.2: Sistema impede ação do usuário e avisa que o tamagotchi já está cheio

Nome do caso de uso: UC07: Botar tamagotchi para dormir

Ator: Usuário

Descrição: Botar tamagotchi para dormir para recuperar sua energia.

Pré-condições: Medidor de energia estar abaixo de 100%

Pós-condições: Sistema aumentará energia do tamagotchi

Cenário principal:

1.0: Energia do tamagotchi estar abaixo de 100%

1.1: Usuário clica no ícone de dormir

1.2: Sistema recupera a energia do tamagotchi

Cenário alternativo:

1.0: Energia do tamagotchi estar em 100%

1.1: Usuário clica no ícone de dormir

1.2: Sistema impede ação do usuário e avisa que o tamagotchi não está
com sono

Nome do caso de uso: UC08: Limpar tamagotchi

Ator: Usuário

Descrição: Usuário dá banho em tamagotchi para que ele fique limpo

Pré-condições: medidor de higiene não estar em 100%

Pós-condições: Sistema coloca medidor de higiene em 100%

Cenário principal:

1.0: Medidor de higiene estar abaixo de 100%

1.1: Usuário clica no icone de banho

1.2: Sistema recupera o medidor de higiene do tamagotchi

Cenário alternativo:

1.0: Medidor de higiene do tamagotchi estar em 100%

1.1: Usuário clica no ícone de banho

1.2: Sistema impede ação do usuário e avisa que o tamagotchi está limpo

Nome do caso de uso: UC09: Remédio para o tamagotchi

Ator: Usuário

Descrição: usuário da remedios para aumentar a saúde do tamagotchi

Pré-condições: Tamagotchi está com medidor de saúde baixo.

Pós-condições: O sistema aumenta o medidor de saúde do tamagotchi.

Cenário principal:

1.0: Medidor de saúde do tamagotchi estar abaixo de 100%

1.1: Usuário clica no ícone de remédio

1.2: Sistema recupera saúde do tamagotchi

Cenário alternativo:

1.0: Medidor de saúde do tamagotchi estar em 100%.

1.1: Usuário clica no ícone de remédio

1.2: Sistema impede ação do usuário e avisa que tamagotchi não precisa de remédios

Nome do caso de uso: UC10: Brincar com tamagotchi

Ator: Usuário

Descrição: O usuário brinca com tamagotchi para aumentar a felicidade.

Pré-condições: Tamagotchi está com medidor de felicidade baixo.

Pós-condições: O sistema aumenta o medidor de felicidade do tamagotchi.

Cenário principal:

1.0: Medidor de felicidade do tamagotchi estar abaixo de 100%.

1.1: Usuário clica no ícone de brincar(bola).

1.2: O sistema recupera a felicidade do tamagotchi.

Cenário alternativo:

1.0: Medidor de felicidade do tamagotchi estar em 100%.

1.1: Usuário clica no ícone de brincar(bola).

1.2: O sistema impede a ação do usuário e avisa que Tamagotchi não quer brincar.

Nome do caso de uso: UC11: Deletar tamagotchi

Ator:Usuário

Descrição: Usuário deleta tamagotchi para criar um novo.

Pré-condições: Tamagotchi está criado e vivo.

Pós-condições: Voltar para tela de início

Cenário principal:

- 1.0: O usuário entra no menu de opções.
- 1.1: Usuário escolhe a opção "reset".
- 1.2: Usuário clica na opção de apagar.
- 1.3: O sistema deleta tamagotchi e volta para o início.

Cenário alternativo:

- 1.0: O usuário entra no menu de opções.
- 1.1: Usuário escolhe a opção "reset".
- 1.2: Usuário clica na opção de não apagar.
- 1.3: O sistema não deleta o tamagotchi .

7. REQUISITOS

Requisitos Funcionais:

RF01 - Inserir o nome do tamagotchi

RF02 - Oferecer a opção de “*help*” para o usuário

RF03 - Excluir tamagotchi

RF04 - Brincar com o tamagotchi

RF05 - Alimentar o tamagotchi

RF06 - Tamagotchi deve dormir

RF07 - Tamagotchi deve se alimentar

RF08 - *Save* para o tamagotchi

Requisitos Não Funcionais:

RNF01 - A arte deve ser feita em um programa apropriado para desenho.

RNF02 - O jogo é desenvolvido na linguagem Java.

RNF03 - O jogo é jogado pelo mouse.

RNF04 - O jogo será rodado pelo console.

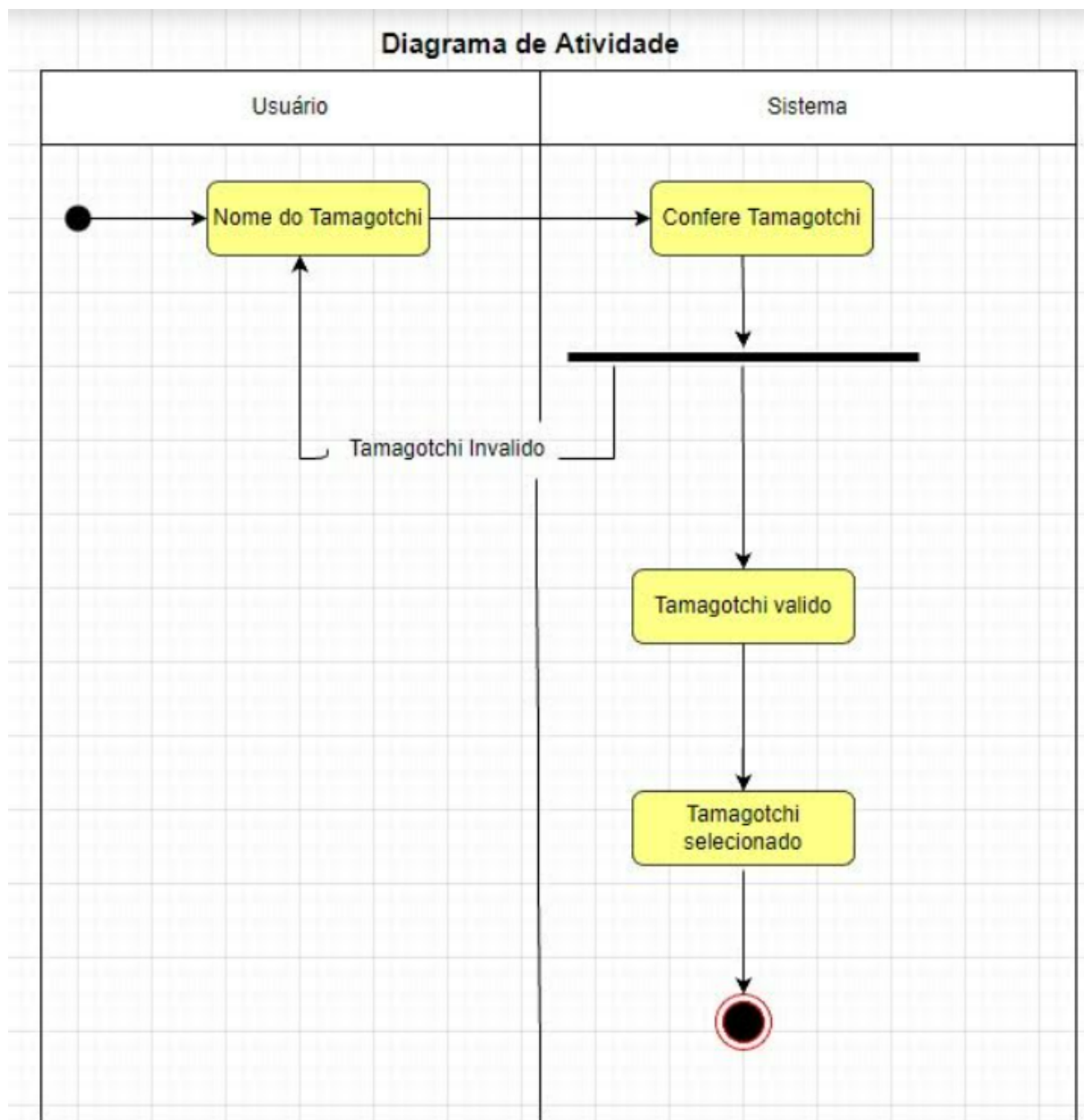
Regras de Negócios:

RN01 - O jogo não deve permitir que o usuário jogue sem inserir um nome
ao tamagotchi

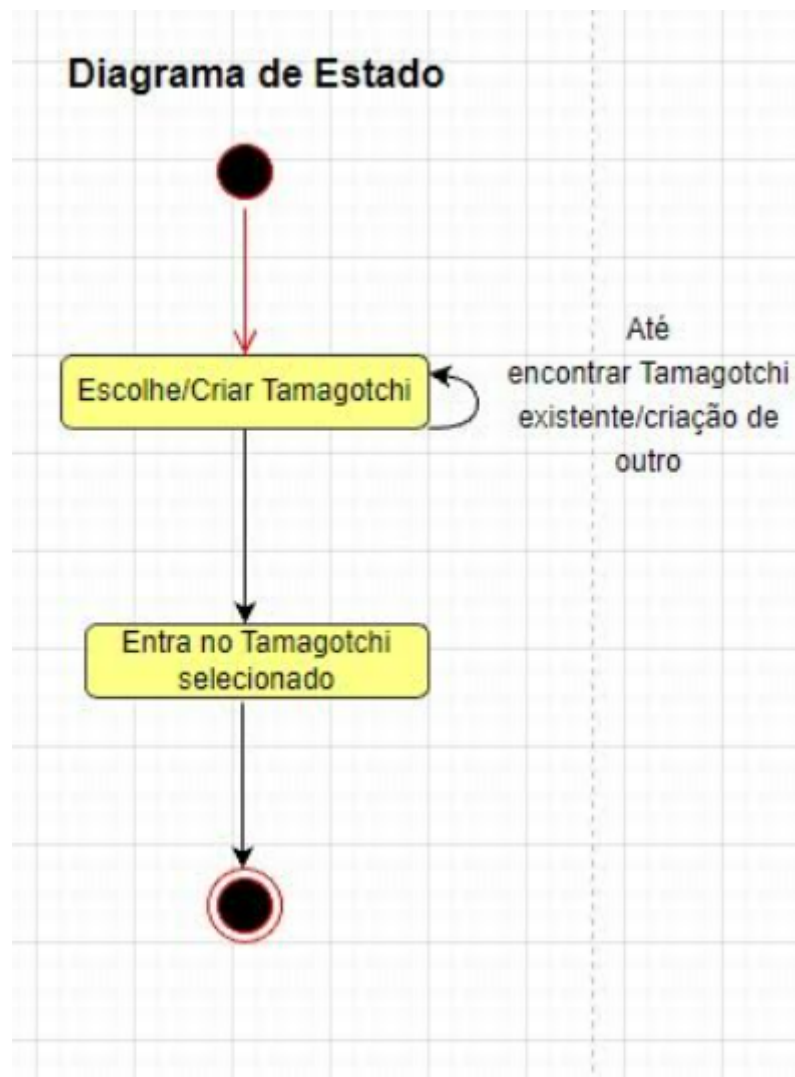
RN02 - Deve ter alterações no frame

RN03 - O tamagotchi deve mudar de acordo com suas barras de energia

8. DIAGRAMA DE ATIVIDADE

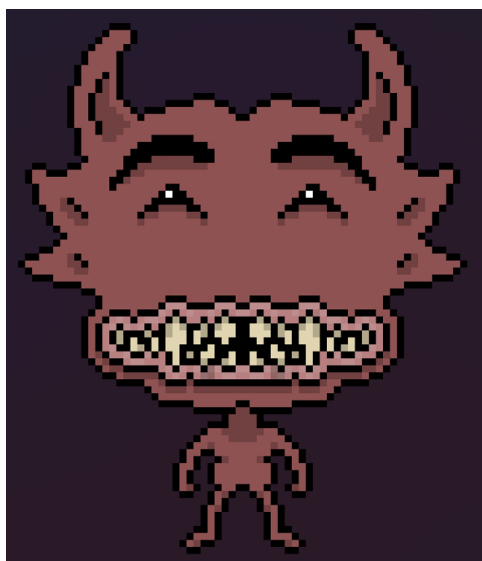


9. DIAGRAMA DE ESTADO



10. PROTÓTIPO

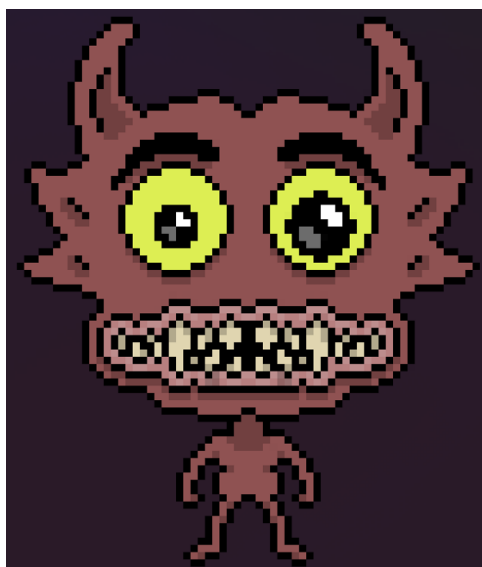
Como se trata de um jogo, há diversos tipos de frames que projetamos para colocar no tamagotchi, então por esse motivo, vamos colocar alguns frames feitos *(Como foi de criatividade total do integrante responsável pela arte não há referências a serem colocadas)*.



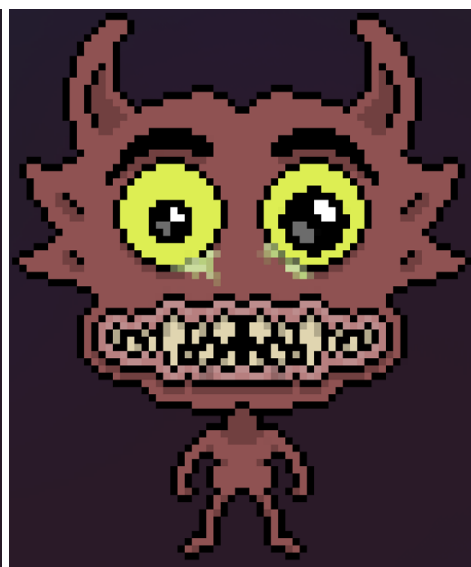
status: feliz



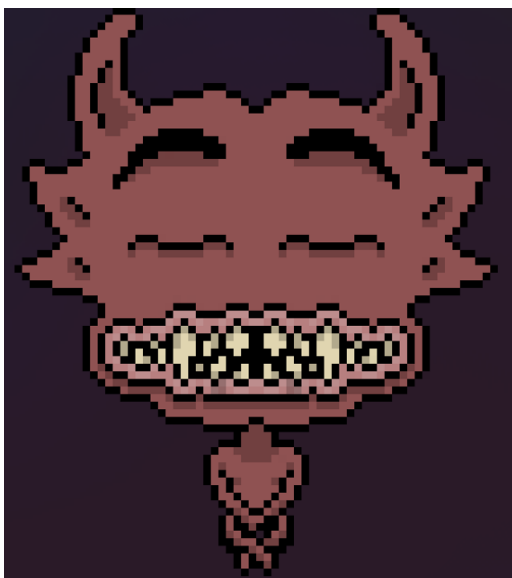
status: feliz cansado



status: normal



status: normal doente



status: dormindo



status: dormindo doente



status: boca aberta brincando



status: boca aberta

11. CONCLUSÃO

Esse projeto foi desenvolvido devido aos interesses comuns dos integrantes do grupo, com bons pensamentos, resolvemos desenvolver esse projeto onde todos tiveram seu interesse de participar e de focar em algo, assim como houve esse apreço pela idealização desse projeto, queríamos levar isso para tela do computador também, ele tem como foco a diversão, trazendo esse tamagotchi feito de diversos frames de uma forma engraçada de interagir.

Como estamos no meio do curso e já tivemos outro P.I mais relacionado à pandemia antes, nessa nova ideia resolvemos trazer algo diferente, talvez um pouco fora do esperado, mas que fez total sentido para cada indivíduo do grupo.

Tivemos que ir atrás de conteúdos que iam além do nosso conhecimento em aula, o que aumenta ainda mais o gosto pela programação; sair da caixa, ir atrás de novas informações, não ficar na mesmice. Mas é claro que há semelhanças em nosso código do que um sistema normalmente deve ter, mas também trouxemos novas informações que vão além do conhecimento em aula.

Sabemos que esses pontos citados foram reconhecidos pelo professor, e é de profundo agradecimento que podemos fazer esse P.I com uma cara um pouco diferente.