UNIVERSIDADE DO ESTADO DE SANTA CATARINA

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

SUMÁRIO EXECUTIVO DO JOGO BANCO IMOBILIÁRIO - VERSÃO LITE

André Dias Ruas

Kenny Kenjiro Fukuya

Engenharia de Software

Joinville

2018

**SUMÁRIO**

INTRODUÇÃO

1. ESCOPO DO PRODUTO

2. COMPOSIÇÃO DA EQUIPE DO PROJETO

3. METODOLOGIA UTILIZADA

4. ESTIMATIVAS DE TEMPO

5. ESTIMATIVAS DE ORÇAMENTO

6. CRONOGRAMA

7. LEVANTAMENTO DE REQUISITOS

8. FEEDBACK PÓS PRIMEIRA FASE

9.CRONOGRAMA SEGUNDA FASE

10.TESTES DE UNIDADE

11.REPOSITÓRIO GITHUB

12.CONCLUSÃO

REFERÊNCIAS BIBLIOGRÁFICAS

# INTRODUÇÃO

Este relatório é referente a atividade solicitada durante a disciplina de Engenharia de Software (SOFT) no segundo semestre do ano de 2018. A atividade tem como objetivo a implementação de um jogo de tabuleiro em uma linguagem de programação de escolha livre e a documentação de todo o processo de desenvolvimento, desde a declaração do escopo até a correção de erros de implementação.

# 1. ESCOPO DO PRODUTO

O jogo escolhido a ser implementado em C é uma versão lite alterada do Jogo Banco Imobiliário.

O Jogo Banco Imobiliário - Versão Lite é jogado por 2 jogadores e possui um tabuleiro circular de 12 casas, onde 11 dessas casas são imóveis que podem ser adquiridas pelos jogadores e 1 não possui utilidade, é apenas o ponto de partida.

Ambos jogadores começam na mesma posição do tabuleiro e possuem um saldo inicial de R$ 10.000.

Cada uma das 11 casas possui um valor de compra e um valor de multa.

Além disso há um sistema de cartas. A cada início de turno o jogador do turno saca uma carta aleatória de um vetor de cartas pré-definidas, essas cartas podem beneficiar o jogador com alguma quantia de saldo ou maleficiar o jogador tomando alguma quantia de saldo dele.

A movimentação é feita de maneira aleatória através de um dado, ou seja, o jogador pode andar de 1 a 6 casas no tabuleiro.

O jogador do turno depois de sacar a carta, rodar o dano e andar X casas, é deparado com a opção de comprar ou não o imóvel daquela casa e caso o imóvel já tenha sido comprado pelo oponente, o jogador além da opção de comprar a casa será obrigado a pagar um valor de multa ao oponente.

O objetivo do jogo é um jogador ser dono de todas as casas simultâneamente ou fazer com que o oponente atinja a falência, ou seja, um dos jogadores tenha saldo negativo.

Abaixo está esboçado o tabuleiro do jogo.



# 2. COMPOSIÇÃO DA EQUIPE DO PROJETO

A equipe do projeto é composta por André Dias Ruas e Kenny Kenjiro Fukuya, ambos são desenvolvedores freelancers que decidiram cooperar na produção deste projeto.  
 A divisão da produção do projeto será dividida de forma igualitária, onde será feita a designação de tarefas a serem cumpridas por cada um dos desenvolvedores.

2.1 Dados de identificação dos integrantes do projeto:

Nome: André Dias Ruas  
 Email: andrediasruas@hotmail.com

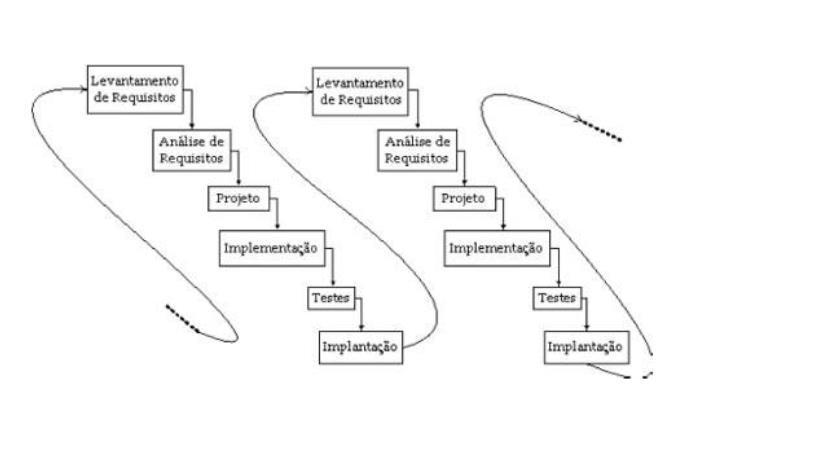
Nome: Kenny Kenjiro Fukuya

Email: kennyfukuya@gmail.com

# 3. METODOLOGIA UTILIZADA

A metodologia escolhida para o desenvolvimento do jogo é a metodologia iterativo incremental. Ela visa dividir o sistema em várias partes e seu progresso é gradativo e cada parte é ligada a outra e no fim se juntam tornando-se algo íntegro e assim facilita a correção de erros, caso seja necessária.

A imagem abaixo representa o modelo utilizado.



# 4. ESTIMATIVAS DE TEMPO

Foram feitas duas estimativas de tempo do projeto. A primeira foi através do Planning Poker, onde os desenvolvedores se juntaram e cada um estimou um tempo ao projeto e então foi escolhido a previsão que fazia mais sentido quanto a explicação dada pelo desenvolvedor. Essa previsão é de 15 horas de trabalho total.  
 A segunda forma de estimativa feita foi através do COCOMO, onde foi utilizado fórmulas para sustentar a estimativa.

O projeto teria complexidade baixa e desenvolvido na linguagem C. Foi previsto um total de 800 Linhas de Código (LOCs) e 6 funções, das quais 3 seriam de Saída Externa (SE) e 3 de Consultas Externas (CE).

Pelo método do COCOMO, na linguagem C um PFNA é equivalente a 128 LOCs, logo temos que um , onde é equivalente a 3 SE e 3 CE de baixa complexidade resultando em, 12+9 = 21. Teremos então um PFNA de 6.25 e um PF de 5.375.

Pela fórmula de estimativa de Esforço, temos que Esforço = 2.4 \* onde KLOC = 0.8, logo, temos um Esforço esperado de 1.8986.

Com o Esforço podemos estimar a Duração do projeto, a fórmula da Duração estimada de um projeto simples é a seguinte, Duração = 2.5 \* , temos então uma Duração estimada de 3.1896 dias de trabalho, onde cada dia possui 8 horas de trabalho, totalizando em 25.5168 horas de trabalho.

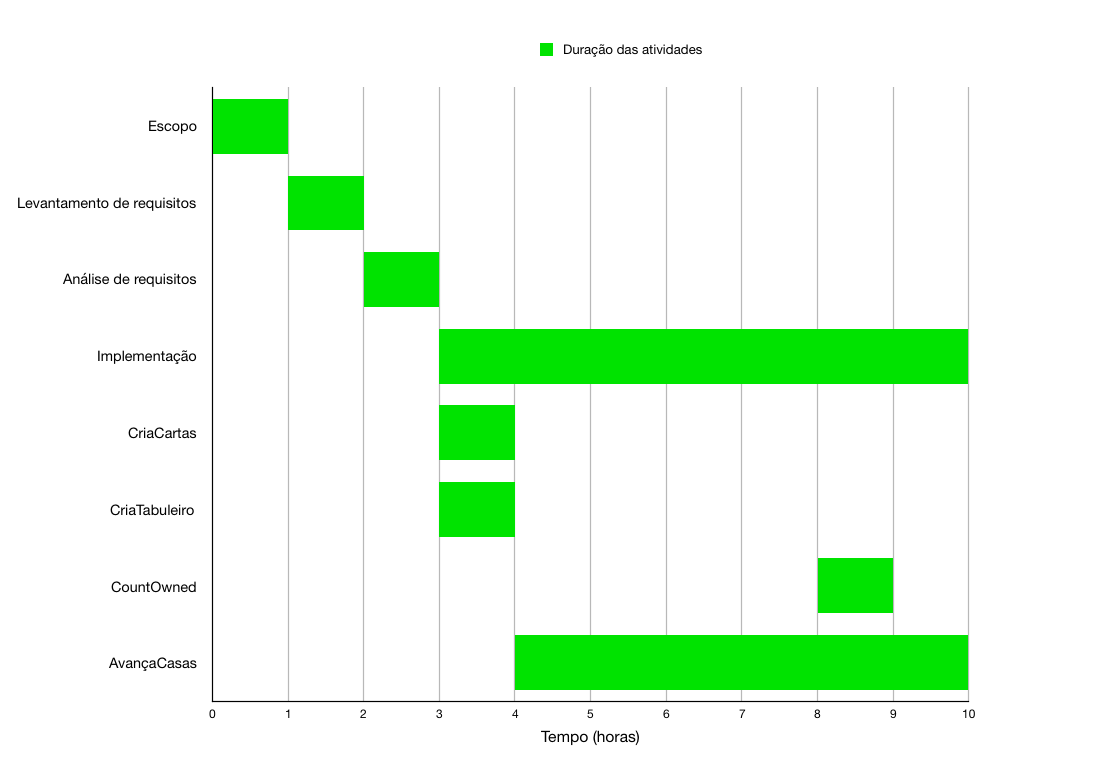
Foi feita então a escolha da estimativa através do COCOMO ao invés do planning poker, devido ao fato de ambos integrantes serem desenvolvedores iniciantes, fazendo com que a estimativa através do Planning Poker seja inconfiável. Além disso o COCOMO possui uma base teórica o que garante maior credibilidade à esta estimativa.

# 5. ESTIMATIVAS DE ORÇAMENTO

Segundo Luciane Costa do blog Vivendo de Freela diz em uma publicação ao blog Tutano Trampos, um desenvolvedor freelancer recebe em média R$ 37.5 reais por hora de trabalho. Logo, como a equipe é composta por dois desenvolvedores, teremos uma expectativa de orçamento de R$ 75 reais por hora de trabalho, considerando a estimativa pelo COCOMO, a estimativa é de R$ 1.913,76.

# 6. CRONOGRAMA

A representação do cronograma para a primeira fase do projeto, que inclui a definição do escopo do produto, levantamento de requisitos, análise dos requisitos e a implementação da versão inicial do jogo, está representada no seguinte diagrama de Gantt.



A divisão foi feita da seguinte maneira; foi feito de maneira colaborativa o escopo do produto, o levantamento dos requisitos e a análise dos mesmo. Já na implementação foi feita uma divisão maior, onde o desenvolvedor André implementou as funções CriaCartas() e CriaTabuleiro(), e o desenvolvedor Kenny implementou as funções CountOwned() e AvançaCasas().

Todas atividades estão com uma estimativa de tempo definida de forma colaborativa entre os desenvolvedores.

# 7. LEVANTAMENTO DE REQUISITOS

**Requisitos Funcionais:**

* Realizar o controle do saldo dos jogadores e de seus imóveis;
* Verificar a cada final de turno as condições de vitória;
* Realizar os movimentos dos jogadores de maneira aleatória.

**Requisitos Não Funcionais:**

* Interface através do console.
* Permitir que 2 jogadores joguem;

**Restrições**:

* O jogador deve esperar o seu turno.

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

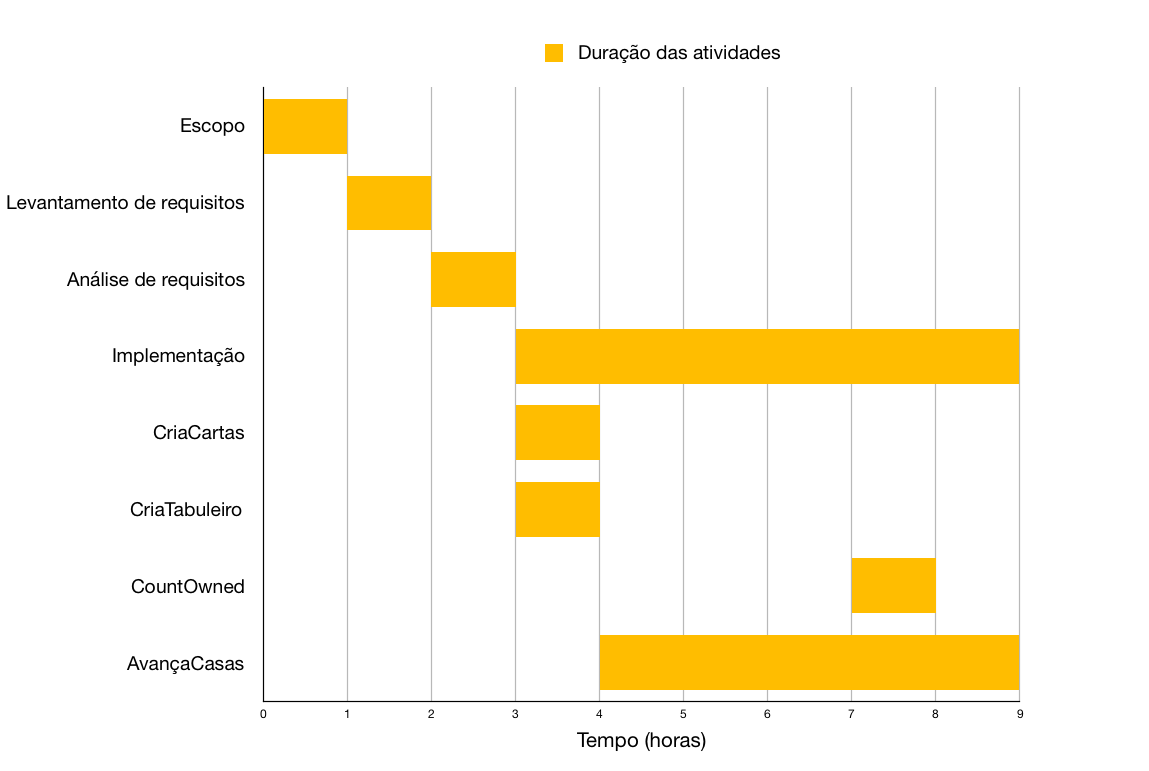
# 

# 

# 8. FEEDBACK PÓS PRIMEIRA FASE

Na primeira fase foi feito a definição do escopo do jogo, o levantamento de requisitos, análise dos requisitos e a implementação de uma versão inicial do jogo.

O diagrama de Gantt a seguir representa como foi utilizado e alocado o tempo do projeto.



Pode-se notar então que houve uma redução de 1 hora quanto à estimativa inicial feita pré primeira fase.

Considerando a estimativa de duração do COCOMO de 25 horas, podemos alocar mais 16 horas para a realização do restante do projeto. Houve um ganho de 1 hora na primeira fase.

# 

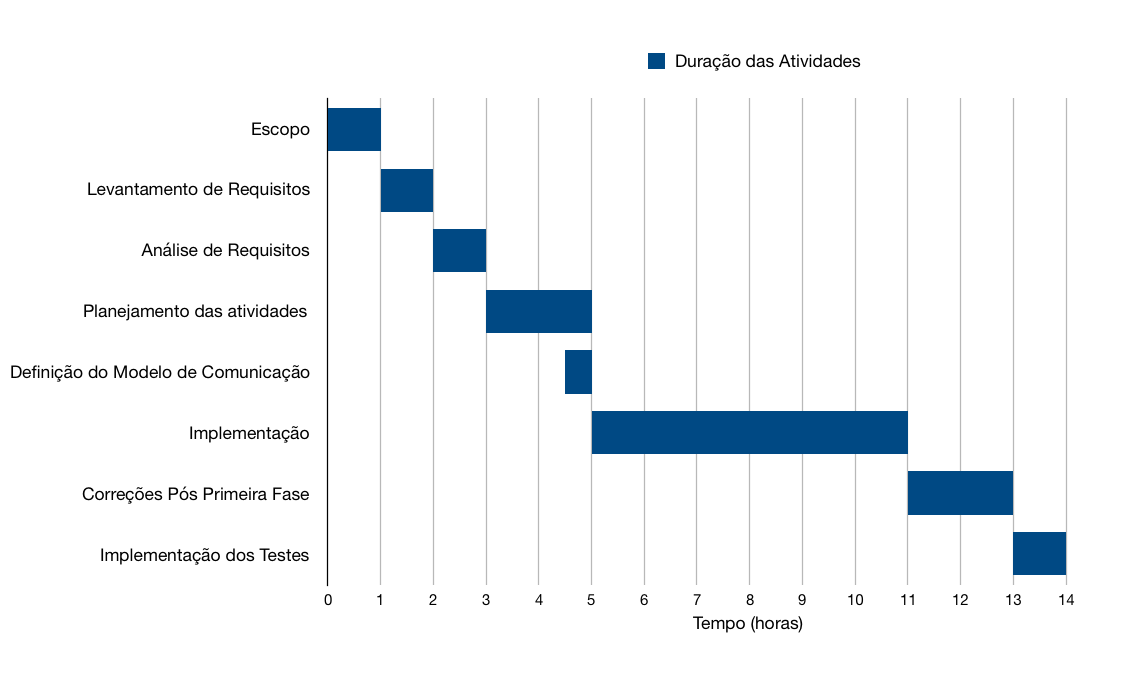
# 

# 9. CRONOGRAMA SEGUNDA FASE

Na segunda fase do projeto foram feitas as correções referentes a analise da primeira etapa, e a realização dos testes de unidade.

A divisão das atividades entre os desenvolvedores se deu da seguinte forma: o desenvolvedor Kenny realizou as correções dos erros referentes a primeira etapa e o desenvolver André implementou as funções de teste que foram definidas em conjunto.

Segue abaixo um diagrama de Gantt que representa um cronograma com os tempos efetivos das atividades da primeira etapa, atividades de controle de projeto e atividades da segunda etapa do projeto.



# 10. TESTES DE UNIDADE

Visto que não seria possível a realização de testes para 100% do código, a definição dos testes de unidade foi baseada nas funcionalidades existentes no software que possibilitam de forma direta a ocorrência de erros. Portanto foram criadas funções para testar a criação do tabuleiro, a criação das cartas e o mecanismo adotado para utilização do dado.

O teste para criação do tabuleiro cria um tabuleiro, e testa o número de casas dele, retornando verdadeiro se o número de casas encontrado foi o esperado. Já o teste para criação das cartas, além de verificar a quantidade de cartas, parte do princípio que a somatória dos valores das cartas deve ser zero (pois implementamos os valores dessa forma), logo, o teste verifica o somatório e retorna verdadeiro se tudo ocorreu como esperado. E o teste para verificação se o dado não é viciado, testa a função de aleatoriedade do dado 100 vezes e armazena esses valores, de forma a armazenar as porcentagens obtidas por cada face do dado, e proporcionando a verificação da confiabilidade dos resultados.

# 

# 11. REPOSITÓRIO GITHUB

O projeto foi guardado no repositório git a seguir: https://github.com/KennyFukuya/SOFT.

# 12. CONCLUSÃO

Ao longo do projeto foi possível pôr em prática o conteúdo aprendido ao longo do semestre, de forma que, as possíveis dificuldades para realização do projeto foram reduzidas. A definição do escopo do produto, o levantamento de requisitos, e a análise dos requisitos, proporcionaram uma maior facilidade na etapa de implementação, e de forma análoga a definição da metodologia e a estimativa de tempo, auxiliaram na definição das etapas e no cumprimento das mesmas e evitou precipitar acontecimentos.

A divisão das tarefas dentro do projeto também foi pensada para facilitar e aumentar a velocidade do desenvolvimento, permitindo uma maior autonomia aos desenvolvedores, conferindo-lhes uma maior margem de poder de decisão permitiu-se que o projeto caminha-se com maior fluidez.

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# REFERÊNCIAS BIBLIOGRÁFICAS

Blog Tutano Trampos. Quanto vale a sua hora como freelancer?. Disponível em: <http://tutano.trampos.co/13156-quanto-vale-preco-hora-como-freelancer/>.

Acesso em 19 de novembro de 2018