

# DOCUMENTO PARA DESIGN DE GAMES

## Os Três Poderes

### **Autores:**

Eric Tachdjian

Gabriel Pascoli

Luiz Augusto Ferreira

Pedro Rezende

Pedro Munhoz

Sergio Lucas

Vinícius Souza

**Data de criação:** 07 de Fevereiro de 2022

**Versão:** 3.3

## Controle do Documento

### Histórico de revisões

Data	Autor	Versão	Resumo da atividade
11/02/2022	C.F 88	0.1	Foram preenchidos os tópicos do projeto de 1.1 a 1.5
09/03/2022	C.F 88	0.2	Foram preenchidos os tópicos do projeto: 1.6 e 1.7 - completos - <b>(1.7.4)</b>
25/02/2022	C.F 88	1.1	Foram preenchidos as secções do projeto: 2, 3, 4 e 6.1
07/03/2022	C.F 88	2.1	Foi reestruturado o tópico do projeto: 6.1
11/03/2022	C.F 88	2.2	Implementação da definição de padrões de qualidade no tópico 5.1 Atualização do segmento 1.7.4
25/03/2022	C.F 88	3.1	Foram Preenchidos os tópicos: 6.3 e 6.4
28/03/2022	C.F 88	3.2	Atualizado o Tópico 1.6 (Value Proposition Canvas)
03/04/2022	C.F 88	3.3	Atualizações: Outlines, todos as secções do GDD - <b>1, 2, 3, 4 e 5</b> Melhora no Design em algumas secções Implementação do playtest na secção 5.
	C.F 88	3.4	

---

## Sumário

<b>1. Visão Geral do Projeto &lt;ADALOVE – Definir Proposta de Valor&gt;</b>	<b>6</b>
1.1 Objetivos do Jogo	6
1.2 Características gerais do Jogo	6
1.3 Público-alvo	6
1.4 Diferenciais	6
1.5 Análise do cenário: Matriz SWOT	6
1.6 Proposta de Valor: Value Proposition Canvas	
1.6.1 Teoria da Mudança	6
1.7 Requisitos do Jogo <ADALOVE – Documentar requisitos>	7
1.7.1 Requisitos coletados na entrevista com o cliente	7
1.7.2 Persona	7
1.7.3 Gênero do Jogo	7
1.7.4 Histórias do jogo (Game stories) ou Histórias dos usuários (user stories)	7
1.7.5 Mecânica	7
1.7.6 Fontes de Pesquisa / Imersão	7
<b>2. Game Design &lt;ADALOVE – Elaborar fluxograma do jogo&gt;</b>	<b>8</b>
2.1 História do Jogo	8
2.2 Fluxo do Jogo e Níveis (os níveis são opcionais)	8
2.3 O Mundo do Jogo	8
2.3.1 Locações Principais e Mapa	8
2.3.2 Navegação pelo Mundo	9
2.3.3 Escala	9
2.3.4 Ambientação	9
2.3.5 Tempo	9
<b>3. Level Design (opcional) &lt;ADALOVE – Implementar mecânicas básicas do jogo - parte 1 a 5&gt;</b>	<b>10</b>
3.1 Fase <NOME DA FASE 1>	10
3.1.1 Visão Geral (opcional)	10

3.1.2 Layout Área (opcional)	10
3.1.2.1 Connections (opcional)	10
3.1.2.2 Layout Effects (opcional)	10
3.1.2.3 Quests e Puzzles (opcional)	10
3.1.3 Balanceamento de Recursos (opcional)	10
3.1.4 The Boss	11
3.1.5 Outros Personagens	12
3.1.6 Easter Eggs	13
<b>4. Personagens &lt;ADALOVE – Implementar mecânicas básicas do jogo - parte 1 a 5&gt;</b>	<b>14</b>
4.1 Personagens Controláveis	14
4.1.2 <NOME DO PERSONAGEM PRINCIPAL n>	14
4.1.2.1 Backstory	14
4.1.2.2 Concept Art	14
4.1.2.3 Ações Permitidas	14
4.1.2.4 Momento de Aparição	14
4.2 Common Non-Playable Characters (NPC)	15
4.2.1 <NOME DO NPC COMUM n>	15
4.3 Special Non-Playable Characters (NPC)	15
4.3.1 <NOME DO NPC ESPECIAL n>	15
<b>5. Teste de Usabilidade &lt;ADALOVE – Desenvolver relatório de resultados do playtest&gt;</b>	
5.1 Usabilidade	16
5.2 Definir padrões de qualidade	16
<b>6. Relatório - Física e Matemática</b>	<b>17</b>
6.1 Funções	17
6.2 Cinemática Unidimensional	17
6.3 Vetores	17
6.4 Cinemática Bidimensional e mais	17
<b>7. Bibliografias</b>	<b>18</b>
<b>Apêndice</b>	<b>19</b>

# 1. Visão Geral do Projeto <ADALOVE – Definir Proposta de Valor>

## 1.1 Objetivos do Jogo

Resumo:

- Nosso jogo visa ensinar, de forma simples e compreensível, para jovens principalmente, sobre a função e atribuição de cada um dos Três Poderes da Federação, por meio da criação e defesa de uma Proposta de Emenda à Constituição (PEC), para que os cidadãos se tornem sujeitos mais conscientes e conhecedores de meio político pelo qual estão envolvidos.
- A fonte principal do jogo será livre e aberta, sendo possível o acesso a partir da plataforma “GitHub”. Além disso, o jogo será disponibilizado para ser jogado de forma online (website).

- Contamos com diversas diretrizes para realizar o projeto com sucesso, sendo elas:

### 1. Desafios e interesses pessoais no tipo de jogo criado

Nosso interesse pessoal é de entender sobre a constituição e o meio político brasileiro, como por exemplo, a função dos Poderes da Federação e suas atribuições, e o grande desafio é traduzir o que aprendemos para o jogador de forma simples, didática e, sobretudo, divertida.

### 2. Para que serve o jogo

O jogo tem a finalidade de difundir conhecimento sobre como funciona o trabalho e atividades dos Poderes Executivo, Legislativo e Judiciário, utilizando apenas a tramitação de uma PEC como exemplo. Além disso, buscamos instigar o jogador a procurar mais sobre os conhecimentos apresentados dentro da plataforma.

### 3. Para que o jogo está sendo criado

A criação do game visa suprir uma lacuna de conhecimento político e disseminar entendimento de fácil acesso à população, propondo a oferta tanto do conhecimento sobre

estrutura e organização dos Três Poderes e suas funções no processo de aprovação de PEC, quanto noções sobre Direitos e Deveres do Cidadão e do Governo, de forma simples e didática.

#### **4. Visão geral e contexto do jogo**

O jogo é uma aplicação web que pode ser acessada por computadores, com foco principal em conteúdo educacional, para ser uma forma divertida e engajadora de aprender.

Por análise do cliente, o contexto da realidade atual gira em torno de um país polarizado politicamente, carente de informações imparciais sob a ótica da questão política, constitucional e social. Dessa forma, o jogo deve abordar tais quesitos de forma oposta, colocando como foco o aprendizado em que o jogador tire para si os conceitos concretos e crie suas próprias escolhas e ideais.

#### **5. Contexto onde este jogo está sendo criado**

- Esse projeto está sendo desenvolvido por alunos do curso de graduação do Inteli (Instituto de Tecnologia e Liderança), onde será gerado um MVP (Minimum Viable Product) que irá oferecer uma solução tecnológica que visa ensinar jovens e pessoas interessadas sobre a função dos seus representantes e Poderes Constitucionais. Tal proposta, foi lançada pela maior ONG (Organização Não Governamental) de educação cívica no Brasil - Projeto Constituição na Escola.

---

## **1.2 Características gerais do Jogo**

É um jogo de plataforma website, sendo interativo e didático que visa imergir o usuário no processo de aprovação de uma PEC e a funcionalidade dos Três Poderes, valendo-se de escolhas feitas pelo usuário e minigames que impactarão diretamente na jornada do jogador.

O enredo é uma tramitação de PEC e seu processo para ser aprovada ou rejeitada, utilizando elementos da área do direito com linguagem simplificada, para melhor compreensão. É um jogo com vitória e derrota devidamente estabelecidos a partir da aprovação da PEC ou da sua rejeição.

---

## 1.3 Público-alvo

Estudantes do ensino médio e jovens adultos brasileiros, entre 14 e 24 anos, porém também será aberto e disponibilizado para qualquer cidadão que esteja interessado em conhecer sobre as leis do país.

Mesmo que o jogo esteja sendo apresentado para essa faixa etária, se vê necessário um conhecimento um pouco mais básico sobre as questões políticas. Isso se deve pelo fato de caso alguém que já possua uma noção básica de política, conseguiria ter uma melhor experiência e noção de alguns conceitos.

---

## 1.4 Diferenciais

O jogo “Os Três Poderes” possui um grande diferencial que é trazer o conhecimento sobre as **funções** daqueles poderes e representantes desses poderes, os quais “governam” as vidas dos cidadãos, a partir de uma PEC que pode trazer mudanças em certos quesitos dentro da sociedade, assim mudando o cotidiano de muitas pessoas.

Assim, será possível demonstrar para o jogador a importância da sua escolha de representante a partir do voto. Pois, não somente saberá a função daquele que representa seu país, como também saberá a consequência que traz uma má decisão das escolhas que podem trazer mudanças críticas ao país.

- Logo, nosso grande diferencial seria:

**ELIMINAR:** Falta de conhecimento político - Representantes políticos, Estrutura de uma PEC e suas funções, Função dos 3 Poderes.

**ELEVAR:** Diversão no aprendizado; Educação base de política;

**REDUZIR:** Uso excessivo da linguagem culta;

**CRIAR:** Um jogo que dissemine o aprendizado político de uma maneira simples e didática.

---

## 1.5 Análise do cenário: Matriz SWOT

<b>FORÇAS :</b> <ul style="list-style-type: none"><li>• Facilidade de compreensão e acessível para todos.</li><li>• Um meio didático dinâmico para escolas.</li><li>• Estimular o protagonismo no aprendizado.</li></ul>	<b>FRAQUEZAS:</b> <ul style="list-style-type: none"><li>• Área com um benchmark pouco explorado.</li><li>• É um assunto que, para muitos jovens, não desperta interesse.</li></ul>
<b>OPORTUNIDADES:</b> <ul style="list-style-type: none"><li>• Grande mercado de jogos para celular e grande possibilidade de ascensão do mercado.</li><li>• Poucos concorrentes em relação ao público alvo.</li><li>• Um tema pouco abordado na indústria dos games.</li></ul>	<b>AMEAÇAS:</b> <ul style="list-style-type: none"><li>• Falta de interesse por parte dos usuários.</li><li>• Jogos já consolidados no mercado.</li><li>• Poucas maneiras de torná-lo interessante</li></ul>



## 1.6 Proposta de Valor - Modelo Canvas e Teoria da Mudança

### 1.6.1 Proposta de Valor - Modelo Canvas

#### Proposta de Valor

Produto / Serviço	Ganho dos Criadores
Game: "Os Três Poderes"	<ul style="list-style-type: none"><li>- Aprendizado sobre programação</li><li>- Conhecimento político:<ul style="list-style-type: none"><li>+ Estrutura dos Três Poderes e suas funções</li><li>+ Estrutura de uma PEC</li></ul></li><li>- Aprendizado sobre UX</li><li>- Aprendizado sobre Persona</li></ul>
Soluções	
<ul style="list-style-type: none"><li>. Bom Design</li><li>. Vocabulário Fácil</li><li>. Neutralidade</li><li>. Simplificar as informações</li><li>. Disponibilizar outros conhecimentos</li></ul>	

#### Perfil do Cliente



## 1.6.2 Teoria da Mudança

Insumos	Atividades	Produtos	Resultados	Impactos
<ul style="list-style-type: none"><li>- Godot.</li><li>- GitHub.</li><li>- Adalove.</li><li>- Aulas.</li><li>- Apoio à Constituição na Escola.</li><li>- Infraestrutura do Inteli.</li><li>- Framework Scrum.</li><li>- Jira.</li><li>- Canva.</li></ul>	<ul style="list-style-type: none"><li>- Criação de códigos.</li><li>- Pesquisa de Imagens.</li><li>- Estudo sobre política.</li></ul>	Jogo educativo que aborda política, visando aumentar o conhecimento político dos estudantes.	<ul style="list-style-type: none"><li>- Ensino político.</li><li>- Aumentar o conhecimento sobre pecs</li></ul>	<ul style="list-style-type: none"><li>- Cidadãos bem educados em relação ao mundo da política.</li><li>- Melhor escolha eleitoral.</li><li>- Cidadãos com melhor noção de seus direitos.</li></ul>

## 1.7 Requisitos do Jogo <ADALOVE – Documentar requisitos>

### 1.7.1 Requisitos coletados na entrevista com o cliente

<https://docs.google.com/document/d/1L7plhqpPEEYoApHSwwLbvhJ5cEG-aJa5/edit?usp=sharing&oid=112225462967162168991&rtpof=true&sd=true> → perguntas feitas ao Felipe Neves

O jogo em questão se baseia nos Três Poderes e na estrutura de uma Proposta de Emenda Constitucional, um game que retrata o ensino sobre política e constituição no Brasil, em que o jogador tem em mãos a possibilidade de interagir com os Três Poderes (Executivo, Legislativo e Judiciário), por meio do desenvolvimento de PECs, a fim de sanar problemas populares e garantir um mandato de qualidade.

A narrativa envolve o Presidente da República, sendo que o diferencial do jogo: “Os Três Poderes”, é ser o pioneiro no mercado a abordar o ensino político e constitucional brasileiro de forma intuitiva e interessante, estimulando o jovem a aprender por conta própria a partir do jogo.

Assim, o principal diferencial é o fato de que é abordado o Sistema dos 3 Poderes e exemplificado suas funções a partir da passagem de uma PEC. Além disso, o objetivo é que o jogador tenha noção da função dos seus representantes a partir da visão dessa passagem da PEC e entenda como se estrutura conhecimentos gerais e técnicos. Dessa forma, o usuário saberia a função dos representantes políticos e teria uma maior noção durante a escolha de tais.

---

### 1.7.2 Persona



NOME: Enrique

IDADE: 19

OCUPAÇÃO: Estudante

**"Gosto de política e sou gamer casual"**

#### Biografia:

É um jovem de 19 anos que estudou em um Instituto Federal, sempre gostou de jogar jogos casualmente. Nunca teve um ensino sobre a constituição brasileira.

## Características (personalidade, conhecimentos, interesses, habilidades):

Gosta  
de jogos  
casuais

É  
interessado  
em política

Está no  
ensino  
superior

Tem um  
conhecimento  
básico de  
política

Classe  
econômica  
média

### Motivações com jogos:

Se diverte com jogos mais  
simples e didáticos.  
Tenta jogar alguns jogos  
para ver se despertam seu  
interesse, para assim  
continuar jogando

### Dores com jogos:

Os jogos atuais tem  
muitas propagandas  
que atrapalham a  
gameplay

Os jogos demandam  
muito tempo,  
atrapalhando os estudos  
e se tornando maçante  
com o tempo, resultando  
na perda de interesse  
gradativo

### Motivações com o problema:

Ter uma melhor  
perspectiva sobre o  
processo de eleição  
brasileira

Aprender e  
compartilhar com  
seus amigos

### Dores com o problema:

Ele não sabe sobre a  
constituição e precisa  
conhecer seus  
direitos básicos como  
cidadão

Falta de ensino  
qualitativo e que  
desperte interesse

### 1.7.3 Gênero do Jogo

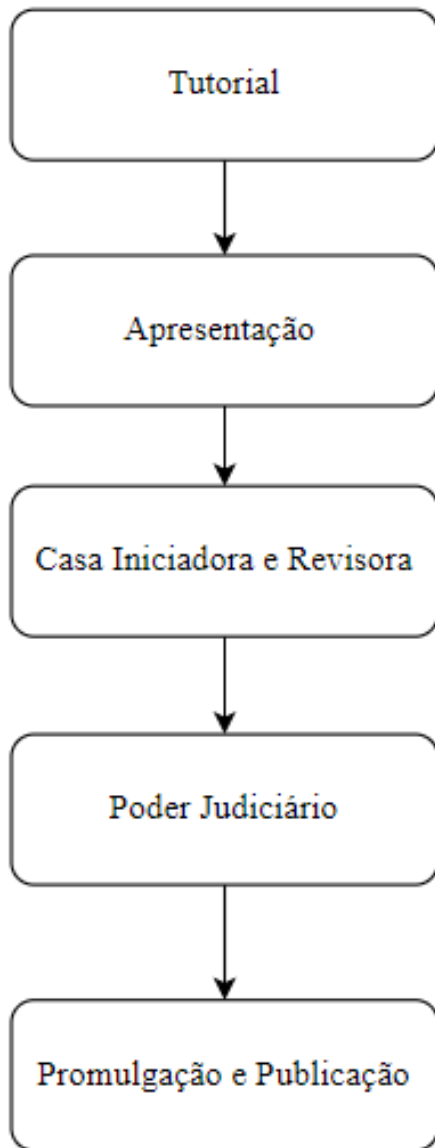
“Os Três Poderes” é um jogo de estratégia, uma vez que o jogador necessita tomar decisões baseadas numa lógica de jogo suficiente para que o presidente cumpra seu mandato no governo brasileiro.

Além disso, o jogo se baseia na Constituição Brasileira, o que faz com que possua um gênero educacional.

#### 1.7.4 Histórias do jogo (Game stories) ou Histórias dos usuários (user stories)

---

##### Fluxo do Game em 5 partes



Épico	Parte	Game Story	Complexidade (Fibonacci)	Status
Tutorial	0 - Orientações (Conselheiro)	Eu, jogador, devo acompanhar as orientações do conselheiro para ter melhor noção do enredo e de como progredir durante o jogo.	8	Feito
Apresentação	1 - Escolha da PEC (Executivo)	Eu, jogador, preciso escolher uma Proposta de Emenda à Constituição, para resolver problemas relacionados às temáticas apresentadas.	2	Feito
Apresentação	1- Explicação da PEC (Executivo)	Eu, jogador, preciso entender 'o que é' e 'como funciona' a estrutura de uma PEC.	8	Em Progresso
Apresentação	2 - Minigame (Executivo)	Eu, jogador, após ter terminado o mini game corrida pela saúde, quero ser pontuado. / catch votos impressos, quero ser pontuado.	8	Em Progresso
Casa Iniciadora e Revisora	2- Minigame (Legislativo)	Eu, jogador, após ter minha performance no minigame pontuada, quero conseguir acessar o poder legislativo.	3	Feito
Casa Iniciadora e Revisora	3- Debate (Legislativo)	Eu, jogador, acessarei o prédio do Legislativo para começar a primeira parte do Debate, que seria na Casa Iniciadora (Câmara dos Deputados)	21	Em Progresso
Casa Iniciadora e Revisora	3 - Debate (Legislativo)	Eu, jogador, após ser aceito nos dois turnos da Casa Iniciadora, irei para a Casa Revisora (Senado Federal) onde irei ter mais dois turnos de Debate.	21	Em Progresso
Poder Judiciário	4 - Revogação ou Aprovação (cutscene)	Eu, jogador, irei ser julgado pelo STF (Supremo Tribunal Federal), se haverá possibilidade de passagem da PEC ou não.	8	Pendente
Poder Judiciário	4 - Revogação ou Aprovação (Judiciário)	Eu, como poder judiciário, devo analisar a PEC para habilitar tal ser promulgada ou não. Caso contrário, será revogada.	13	Pendente
Promulgação e Publicação	5 - PEC habilitada (Executivo)	Eu, jogador, tive minha PEC habilitada para Promulgação e publicação. Dessa forma, agora tenho uma PEC "aprovada".	8	Pendente

### 1.7.5 Mecânica

É um jogo 2D, onde o jogador deve interagir com as “3 Casas de Poderes Políticos no Brasil” de maneira com que ele consiga aprovar a maior quantidade de PECs durante seu mandato.

Ele deve começar no poder executivo, passar para o legislativo e, então chegar ao judiciário.

Utilizando de vários botões e funções, o jogador é guiado pelo conselheiro para que ele possa, de maneira compreensiva e fácil, criar → entender → desenvolver → aprovar uma Proposta de Emenda Constitucional (PEC).

### 1.7.6 Fontes de Pesquisa / Imersão

Indicar as principais fontes de pesquisa do jogo para a criação de conteúdo (feitos em aula): imagens, filmes, animações, livros e outros que realmente foram usados para a etapa de imersão ao tema. Fazer um sumário do conteúdo pesquisado a partir dos referidos materiais.

Fonte
1. Jogo Branches of Power
2. Material de conteúdo da Constituição Brasileira, sobre os 3 Poderes, Estrutura de uma PEC, todos foram pegos de sites oficiais do governo ou disponibilizado pelo cliente (site: Politizei!) - Inseridos na Bibliografia
3. Para a programação do Game foram utilizados o GDscript e vídeos no Youtube.
4. O áudio aplicado no jogo foi feito por um dos integrantes do grupo.
5.

---

## 2. Game Design <ADALOVE – Elaborar fluxograma do jogo>

### 2.1 História do Jogo

- **Tema (*storyline*)**

- O protagonista terá interação com os três poderes no jogo, controlando-os de forma externa para criar e aprovar PECs, de forma que ele esteja na posição de Presidente e seja tal imagem que apresenta as PECs.
- Assim, busca-se instanciar o processo de cada um dos três poderes e o que eles fazem nesse processo de aprovação ou rejeição de tal PEC.

- **Conceito**

- Controlar a passagem da PEC escolhida, dentre os Três Poderes, para definir como será seu processo na tentativa de promulgação e publicação, além de analisar como cada um deles influencia nessa PEC.

- **Plano de fundo da história (*backstory*) \*\*\***

- “Com o passar do tempo as PECs tem sido cada vez mais relevantes no nosso dia a dia porém muitos desconhecem o que elas fazem e como são criadas e para que elas são criadas você protagonista tem a missão de entender sua importância e o que os três poderes agem sobre isso”

- **Premissa**

- O jogador acompanha o processo de criação de PECs onde participará ativamente em seu processo de aprovação onde as entrelinhas desse processo será ensinado de forma didática e divertida.

- **Sinopse**

- Acompanhe a saga onde você jogador está encarregado do processo de criação e publicação de uma PEC, onde terá o controle de seu processo e instanciação nos três poderes.

- **Estrutura narrativa escolhida**

- O jogo será estruturado em uma narrativa na 3.<sup>a</sup> pessoa durante o processo de criação de aprovação da PEC. Junto a um NPC ( Conselheiro) de ajuda e dica.



- **Elementos do roteiro para a estrutura narrativa escolhida** (animação, cut-scenes, McGuffin, diálogos, foreshadowing, inciting incident, etc) – {Colocar detalhes dos elementos escolhidos por fase do jogo}

- Há um jornal no qual são veiculadas notícias referentes ao processo de aprovação da PEC. Além disso, as consequências das decisões tomadas pelo jogador, abordam os impactos reais na vida dos cidadãos do país, sejam eles bons, sejam ruins. Como exemplo, por agora temos a PEC do voto impresso, a qual irá falhar independente das escolhas.

- **Níveis de interatividade do jogo**

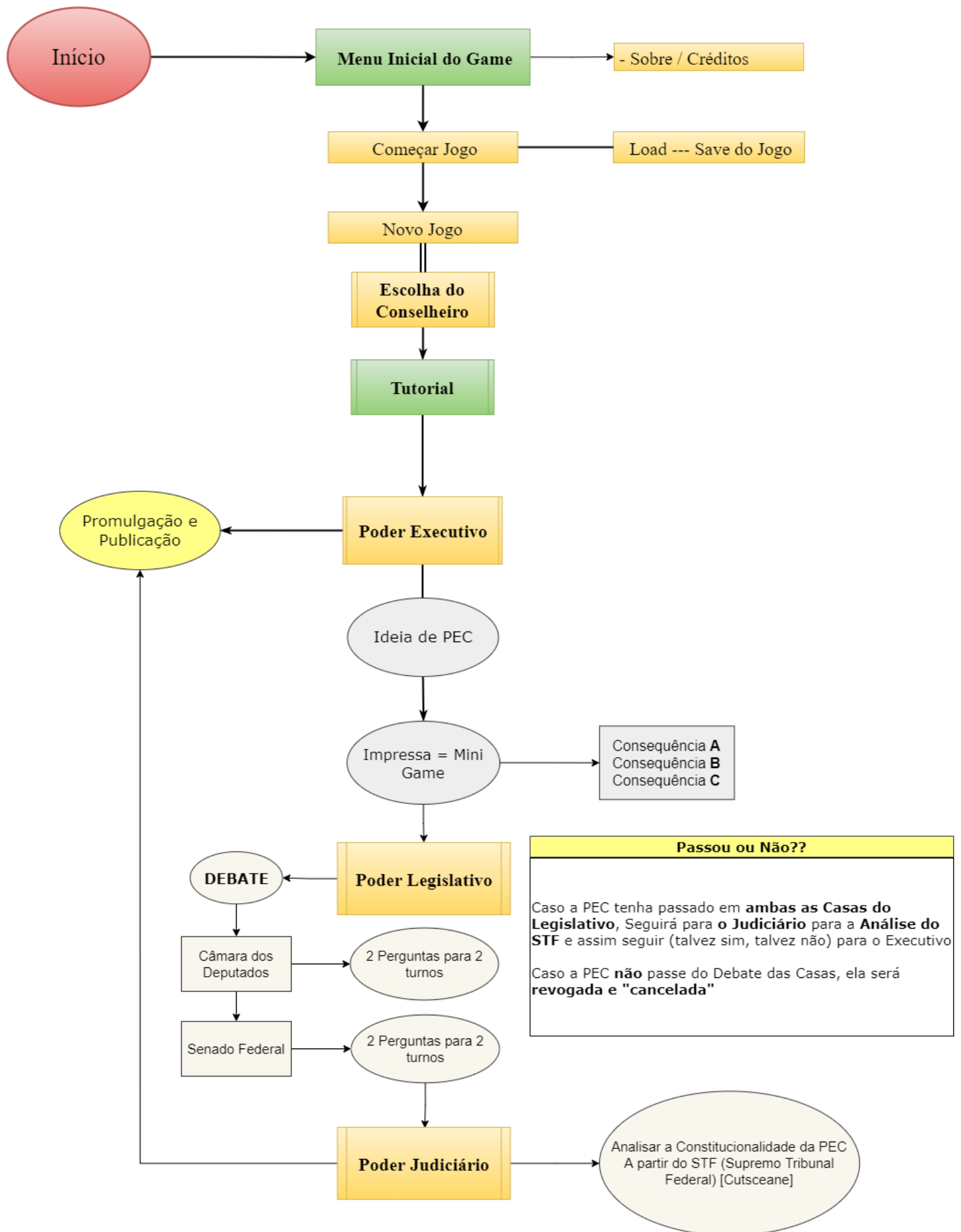
- Interatividade alta, sendo que o jogador é 100% responsável pelo caminho a ser trilhado na narrativa do jogo .

---

## **2.2 Fluxo do Jogo**

- No início do jogo é dada uma ênfase maior para o aprendizado, isto é, “como jogar”.
- Posteriormente, há um aumento significativo na dificuldade, devido a apresentação de alguns mecanismos e conteúdos diversificados. Sendo passado do Executivo → Legislativo → Judiciário → terminando no Executivo.

### **- Fluxograma do Jogo: “Os Três Poderes”**



## **2.3 O Mundo do Jogo**

### **2.3.1 Locações Principais e Mapa**

- O jogo se passa em 3 cenários principais, que são:
- + Palácio do Planalto = Executivo;
- + Congresso Nacional = Legislativo;
- + Supremo Tribunal Federal = Judiciário.

### **2.3.2 Navegação pelo Mundo**

- O jogador se moverá pela 'interface' (os três prédios dos poderes) no decorrer da aprovação do projeto de pec, passando, com cliques, pelos diferentes Poderes Brasileiros, sendo que, em algum momento, será direcionado a um dos minigames propostos, como, por exemplo, um endless runner - referente ao caminho do projeto da área da saúde - ou um minigame do voto impresso.

### **2.3.3 Escala**

- A maior parte do jogo se passará em interfaces, porém a escala irá variar dependendo dos minigames propostos, por esse motivo as escalas estão abaixo:
- endless runner: O personagem se encontra em uma dimensão relativamente pequena em comparação ao mundo do mine-game, sendo uma escala de 1.2
- voto impresso: a urna coleta votos relativamente pequenos em comparação ao tamanho normal. 5.1

### **2.3.4 Ambientação**

- A ambientação do jogo não será muito diversa, o jogo se passará de dia sem mudanças climáticas e sem um cenário interativo.

### **2.3.5 Tempo**

- O tempo será um mandato de 4 anos em que se passará em turnos de 12 rounds. Cada round é referente a um quadrimestre do ano, em que as decisões tomadas simbolizam a passagem do tempo.
  - Por enquanto está em progresso, Mas a nova ideia é um mandato indeterminado - devido a possibilidade de seleção de diversas PECs.
-

---

## 3. Level Design (opcional) <ADALOVE – Implementar mecânicas básicas do jogo - parte 1 a 5>

### 3.1 Fase <NOME DA FASE 1>

#### 3.1.1 Visão Geral (opcional)

- Depois da inicialização do game você é direcionado ao main screen do jogo onde se encontra um tutorial explicando os três edifícios do game: o Poder Executivo, Legislativo e o Judiciário, .

- Sendo o executivo o edifício que você pode propor sua PEC, passando pelo processo para sua aprovação, a fim de dar continuidade ao jogo para te levar ao objetivo final, que é a publicação ou não da PEC.

#### 3.1.2 Layout Área (opcional)

- A área do Executivo se liga ao do Legislativo a partir da PEC escolhida e do Mini Game feito.

- Logo depois da conexão com o Legislativo e o desenvolver do Debate, há um direcionamento ao Judiciário.

##### 3.1.2.1 Connections (opcional)

- No cenário da página principal, prédios dos Três Poderes, há, após o desenvolvimento do tutorial, o prédio do Legislativo e do Judiciário bloqueados por meio de uma técnica que os deixa invisíveis enquanto o poder Executivo não for acessado. Após o jogador passar pelo Executivo e voltar ao menu principal, os demais poderes da República estarão clicáveis e com cores ao passar o mouse em cima deles, enquanto antes ficavam cinza escuro.

##### 3.1.2.2 Layout Effects (opcional)

- Efeitos visuais aplicados com vitória/conquista/derrota em decorrer do jogo, sons (trilha sonora e efeitos sonoros feita por membro do grupo)

##### 3.1.2.3 Quests e Puzzles (opcional)

- Quest: Vencer o Primeiro Debate

- Quest: Vencer o Segundo Debate

- Quest: Promulgação e Publicação de PEC

### **3.1.4 The Boss**

- Temos um Boss em uma das etapas do Debate da PEC, onde, a partir de um jogo de perguntas e respostas, devemos responder as perguntas de uma maneira correta para que possamos ganhar pontos sobre o adversário, e caso a resposta for errada, perderemos pontos de vida. Ou seja, a recompensa de responder certo as questões está no fato de não perder barras de vida e tirá-las do adversário (Deputados e Senadores)

### **3.1.5 Outros Personagens**

- O nosso jogo conta com um NPC que irá te ajudar durante sua gameplay com dicas e meios de fazer sua jornada mais fácil, o 'Conselheiro', que atua como um guia durante o tutorial.

### **3.1.6 Easter Eggs**

— Ao clicar no botão “power” do monitor, no judiciário, o usuário é encaminhado para uma cena em que o monitor está desligado.

---

## 4. Personagens <ADALOVE – Implementar mecânicas básicas do jogo - parte 1 a 5>

Aqui, descrever brevemente a relação dos personagens na história e uma tabela com os personagens do jogo, apontando a fase em que aparecem (se houver personagens).

### + Character Appearance Chart

Personagem	Turno 1	Turno 2	Turno 3	Turno 4	Turno 5	Turno 6	Turno 7
Presidente do Brasil							
Presidente do STF							
Deputados							
Conselheiro							

### 4.1 Personagens Controláveis

#### 4.1.2 <NOME DO PERSONAGEM PRINCIPAL >

##### 1) Presidente da República do Brasil:

- Personagem principal do jogo. Contém a função mais importante, sendo suas decisões controladas pelo jogador. Seu objetivo é conseguir a aprovação do máximo de PECs possíveis, considerando os desejos da população, daqueles que compõem o governo e dos Três Poderes. Personagem personalizável, pois o jogador poderá escolher sua etnia e gênero antes do início do jogo.

##### 4.1.2.1 Backstory

- A princípio será um personagem com o poder de escolher o rumo do jogo e que tomará as decisões. Seu objetivo é alcançar o maior número de PECs aprovadas e adquirir conhecimento político ao longo do jogo e entender a função dos Três Poderes . Ele possuirá um “mentor” (conselheiro) que irá auxiliá-lo em suas escolhas e esclarecer possíveis dúvidas e explicar seus erros.

##### 4.1.2.2 Concept Art

Esboços do personagem.

(Esboços do personagem disponibilizados pela equipe do cliente )





#### **4.1.2.3 Ações Permitidas**

- Criação de PECs visando melhorar as condições de vida no país/ estabilizar os conflitos internos, consequentemente aumentando a sua reputação.
- Gameplay do Mini Game.
- Escolha de respostas durante o Debate.

#### **4.1.2.4 Momento de Aparição**

- O personagem aparecerá em todas as situações onde haverá uma ação do jogador, seja em algum diálogo ou algum minigame e no debate. Seu objetivo é se tornar um avatar onde o jogador pode se imaginar no jogo para aumentar a sua imersão no decorrer do jogo



## 4.2 Common Non-Playable Characters (NPC)

### 4.2.1 <NOME DO NPC COMUM>

#### População:

- Serão uma quantidade significativa de personagens que representam os desejos da população e irão reagir conforme as PECs propostas sejam aprovadas ou negadas em sua jornada de tramitação, essas reações podem ser tanto positivas quanto negativas.

## 4.3 Special Non-Playable Characters (NPC)

### 4.3.1 <Conselheiro>

Para cada NPC especial (mini-boss, boss, mentor/guia etc), descrever sua história, comportamento perante o personagem (agressivo, amistoso, indiferente etc), seus dados pessoais como pontos de vida e outros que forem implementados (pontos de magia, habilidades etc). O comportamento do personagem será estático ou dinâmico. Se +dinâmico definir como o mesmo deve ser alterado.

- O conselheiro consiste em um NPC que não é envolvido em batalha, e tem como missão guiar o jogador no tutorial, ajudando a tomar as decisões corretas e explicando o porquê das decisões erradas estarem erradas. Ele servirá para direcionar e instruir o personagem principal e, conseqüentemente, o jogador ao conceito de PECs e sua implementação

---

## 5. Teste de Usabilidade e Padrões de qualidade <ADALOVE – Desenvolver relatório de resultados do playtest>

### 5.1 - Usabilidade

**Número de testes:** 24 testes (playtest feito na aula) e 5 (pessoas de fora do Inteli)

- **Notas Em Geral dos seguintes tópicos:**

- Nível de Clareza: Razoável
- Nível do Conteúdo: Bom
- Nível de Fluxo: Bom
- Nível de Balanceamento: Bom
- Nível de Duração: Bom (ficou meio Ambíguo)
- Nível de Integração: Bom - Ótimo
- Nível de Diversão: Bom
- Nível de Interação: Bom
- Avaliação Geral do Jogo: Razoável - / Bom +
- **Possíveis melhorias, sugestões...:**
  - Melhorar arte,
  - respostas mais intuitivas,
  - contexto entre mini game e o debate,
  - implementar o que falta,
  - deixar claro o fluxo do jogo,
  - **Deixar o texto visível,**
  - **Textos muito densos,**
  - **Muito texto,**
  - os minigames estão muito bons
  - Deixar os botões personalizados
  - Gostaram da disponibilização de sites informativos (adicionar conteúdo informativo)
- **Bugs:**
  - Alguns bugs de hitbox no mini game da corrida
  - Cenário por trás do mini game dos votos
  - Design de alguns cenários

**Pontos positivos (observados nos testes em geral):**

→ A grande maioria dos testers gostaram bastante do visual e do design simples mas muito bom e agradável.

Apreciaram a música. Caso o tutorial seja lido é possível tirar proveito tranquilamente durante o jogo.

Praticamente todos os testers adoraram os mini games, tendo descontraindo e dando um bom divertimento

**Bem variado o nível de interesse dos jogadores pelo assunto.**

### **Pontos de melhoria (observados nos testes em geral):**

→ Muitos falaram sobre a necessidade de amenizar no Conteúdo para deixar mais intuitivo e interessante. Além disso, a maioria perdia a atenção durante o tutorial que é muito longo, deram ideias de separar a explicação de cada prédio para os seus respectivos (onde quando o jogador entrasse seria habilitado a explicação).

A maioria também não entendeu o impacto do mini game dentro do jogo, requisitando uma explicação.

Necessidade de mudar o estilo da fonte.

### **Número do teste: Tester número 3 (fora do Inteli)**

Nome e perfil dos tester:

- Nome - Anônimo (para não haver compartilhamento) [Amigo do Pedro Rezende], 18 anos, cursando engenharia civil.

Gosta de jogos, porém casualmente joga, não se atrai muito por jogos educacionais mas dá uma chance de vez em quando. Se interessa por uma explicação simples ou pela disponibilização oficial do conceito.

De 0 a 10, o tester se interessa em um nível 3 sobre conceitos políticos.

O teste ocorreu por uma call e compartilhamento da tela do Game, onde o Tester comandava as escolhas do que era necessário ser feito.

### **O que observar e perguntar durante o teste:**

#### **- Observar e registrar:**

#### **Conseguiu começar o jogo? Entendeu a mecânica do jogo? Aprendeu como jogar?**

→ Conseguiu começar o jogo tranquilamente e obteve uma noção básica sobre a mecânica do jogo. Aprendeu a jogar de acordo com a intuitividade própria, porém mostrando a necessidade de um direcionamento.

#### **- Conseguiu controlar o jogo?**

→ O tester conseguiu controlar o decorrer do jogo, já que até o momento está em uma forma simples.

#### **- Progrediu no jogo? Passou de fase? Fez pontos? Chegou ao final? Perdeu rápido?**

→ Conseguiu progredir e desenvolver o decorrer do jogo, porém não entendeu as funções de algumas etapas e meio que “seperdeu” até a etapa final do game.

#### **- Entendeu as regras do jogo? Teve dificuldade de compreensão? Teve dificuldade ao jogar?**

→ O tester conseguiu entender tranquilamente as regras do jogo, devido a leitura do tutorial, mas alertou sobre como estava longo e cansativo a leitura.

Relatou não ter dificuldade em jogar nem em compreender a estrutura do jogo.

- **Foi muito fácil? O jogo foi desafiador?**

→ De acordo com o tester, não se trata muito do jogo ser fácil ou difícil, mas sim o quanto conhecimento é possível adquirir a partir daquele Game, além do quanto o jogador está disposto a consumir daquele conhecimento. Nesse quesito, se torna um pouco desafiador o adquirento do aprendizado político com foco, ao mesmo tempo que se diverte.

- **Perguntar a quem testou:**

**Numa escala de 0 a 10, quanto você se divertiu nesse jogo?**

→ De acordo com o Tester a nota foi '7'.

**O que você gostou no jogo?**

→ Visual e o poder judiciário

**O que poderia melhorar no jogo?**

→ Tirar um pouco da seriedade e densidade das perguntas, além de distribuir melhor as explicações e tutoriais, para deixar menos cansativo.

- **Analisar os quesitos de Melhoria apresentados pelos Testers dentro do Inteli e fora do Inteli, para buscar uma melhoria em que abranja maior quantidade de jogadores, agradando a todos.**

## 5.2 - Definir Padrões de Qualidade

CARACTERÍSTICAS	SUBCARACTERÍSTICAS	SIGNIFICADO
<b>Funcionalidade</b>  O conjunto de funções satisfazem as necessidades explícitas e implícitas para a finalidade a que se destina o produto?	<b>Adequação</b>	<b>Propõe-se a fazer o que é apropriado?</b>  → Mesmo que a ideia do jogo seja diferente daquela esperada, foi aprovada pelo cliente e apresentada em prática.

<p>→ Sim, a codificação do jogo possibilita tanto que o usuário aprenda sobre o processo de criação de leis dentro da Constituição Brasileira (explícito), quanto que se divirta pelos mini games (implícito).</p>	<p><b>Acurácia</b></p>	<p><b>Gera resultados corretos ou conforme acordados?</b></p> <p>→ Sim, gera resultados na medida em que ensina o usuário sobre o processo de criação de leis, difundindo conhecimento sobre a função de cada setor da política brasileira.</p>
	<p><b>Interoperabilidade</b></p>	<p><b>É capaz de interagir com os sistemas especificados?</b></p> <p>→ Sim, o programa utilizado, sendo ele o Godot, no momento, apresenta um uso mais específico para Web, sendo possível (futuramente) a adaptação para apps.</p>

	<b>Segurança de acesso</b>	<p><b>Evita o acesso não autorizado, acidental ou deliberado a programas e dados?</b></p> <p>→ O código é open source, logo é de fácil acesso para qualquer um, contudo quem pode modificar o código é controlado pelos desenvolvedores.</p>
	<b>Conformidade</b>	<p><b>Está de acordo com normas e convenções previstas em leis e descrições similares?</b></p> <p>→ Sim, está de acordo com todas as normas jurídicas previstas.</p>
<b>Confiabilidade</b>	<b>Maturidade</b>	<p><b>Com que frequência apresenta falhas?</b></p>

<p>O desempenho se mantém ao longo do tempo e em condições estabelecidas?</p> <p>→ Tanto o jogo quanto o grupo se mantém fiel a sua ideia inicial, respeitando-a sempre que possível. As únicas mudanças que ocorreram foram para a melhora/ melhor aproveitamento, mantendo sempre um bom e constante desempenho, com pouca oscilação.</p>		<p>→ O grupo apresenta eventuais discussões e diferentes visões, contudo nada sério e nem levado para o lado pessoal</p>
	<p>Tolerância a falhas</p>	<p>Ocorrendo falhas como ele reage?</p> <p>Com resiliência, o grupo se dispõe a procurar o problema em grupo e, quando nosso conhecimento é insuficiente para resolução, buscamos orientação dos professores.</p>
	<p>Recuperabilidade</p>	<p>É capaz de recuperar dados após uma falha?</p>

		<p>→ De acordo com os comentários e o salvamento do jogo em diferentes formas (nuvem do GitHub e computadores dos desenvolvedores), é possível a recuperação dos dados do jogo.</p> <p>Em questão de salvamento de dados dentro do jogo, será necessário a implementação de um banco de dados que suporte esse quesito.</p>
<p><b>Usabilidade</b></p> <p>É fácil utilizar o software?</p> <p>→ A nossa interface é intuitiva, sendo assim, fácil para o usuário se locomover pelo jogo e suas</p>	<p><b>Inteligibilidade</b></p>	<p>É fácil entender os conceitos utilizados?</p> <p>→ Sim, nosso jogo é intuitivo e de fácil entendimento da interface.</p>



<p>mecânicas, tornando o software de fácil utilização.</p>		<p>→ Sem falar que a maior parte do código é a movimentação entre interfaces. Sendo assim, grande parte dos conceitos utilizados no código é de fácil utilização</p>
	<p><b>Apreensibilidade</b></p>	<p><b>É fácil aprender a usar?</b></p> <p>→ Nosso jogo tem uma mecânica básica e já conhecida, sendo assim, fácil a adaptação para aprender as mecânicas do nosso jogo.</p>
	<p><b>Operacionalidade</b></p>	<p><b>É fácil de operar e controlar a operação?</b></p>

		<p>→ Sim, o nosso jogo é maior parte movido pelo mouse para se movimentar entre as interfaces, com exceção dos minigames, como por exemplo o “endless runner”, que está implementado onde o jogador somente usa a barra de espaço para pular e completar o objetivo.</p>
<p><b>Eficiência</b></p> <p>Os recursos e os tempos utilizados são compatíveis com o nível de desempenho requerido para o produto?</p> <p>→ O processamento dos recursos do jogo em relação ao tempo é extremamente bom, o desempenho</p>	<p><b>Comportamento em relação ao tempo</b></p>	<p><b>Qual é o tempo de resposta e de processamento?</b></p> <p>→ O jogo possui um tempo de resposta rápido, uma vez que se trata de um jogo simples (a respeito das mecânicas).</p>

<p>requerido é simples e rápido como</p> <p>nosso produto entrega.</p>	<p>Comportamento em relação</p> <p>aos recursos</p>	<p>Quanto recurso utiliza?</p> <p>→ O jogo no momento é extremamente leve, sem muitos requerimentos para o usuário.</p>
<p>Manutenibilidade</p> <p>Há facilidade para correções, atualizações e alterações?</p> <p>→ Sim, o código concorda com as éticas de programação e é muito bem comentado.</p>	<p>Analisabilidade</p>	<p>É fácil encontrar uma falha quando ocorre?</p> <p>→ Sim, por facilitação do Godot, não há dificuldades em identificar os problemas.</p>
	<p>Modificabilidade</p>	<p>É fácil modificar e remover defeitos?</p> <p>→ Requer análise da estrutura do código, tanto da mensagem de erro do Godot, quanto da possível mudança no código, porém, as</p>

		<p>mudanças não afetam o resto do código de uma maneira com que o mesmo pare de funcionar.</p>
	<p><b>Estabilidade</b></p>	<p><b>Há grandes riscos de bugs quando se faz alterações?</b></p> <p>→ Na estrutura atual, não se apresentam bugs.</p>
	<p><b>Testabilidade</b></p>	<p><b>É fácil testar quando se faz alterações?</b></p> <p>→ A transição de desenvolvimento para teste é quase imediata.</p>

<p><b>Portabilidade</b></p> <p>É possível utilizar o produto em diversas plataformas com pequeno esforço de adaptação?</p> <p>→ Deverá haver uma análise de posicionamento, porém com essas alterações feitas é fácil de realizar.</p>	<p><b>Adaptabilidade</b></p>	<p>É fácil adaptar a outros ambientes sem aplicar outras ações ou meios além dos fornecidos para esta finalidade no software considerado?</p> <p>→ O jogo não foi produzido pensando em plataformas além da Web e Windows. Para se adaptar a outras plataformas (como a plataforma Mobile, Android e iOS), seria necessário uma análise da ferramenta Godot e uma reapropriação dos recursos.</p>
	<p><b>Capacidade para ser instalado</b></p>	<p>É fácil instalar em outros ambientes?</p>

		<p>→ Utilizando a ferramenta Godot para adequar o jogo desenvolvido a outras plataformas, não se apresenta problemas neste quesito.</p>
	Capacidade para substituir	<p>É fácil substituir por outro software?</p> <p>→ Pode ser alterado e adaptado de forma fácil.</p>
	Conformidade	<p>Está de acordo com padrões ou convenções de portabilidade?</p> <p>→ O desenvolvimento foi feito de acordo com as normas e éticas de código (Indentação, Comentários, etc.).</p>

## 6. Relatório - Física e Matemática

### 6.1 Funções

Quais funções são usadas no jogo desenvolvido neste projeto?

Coloque os trechos do programa no Godot onde elas aparecem e explique sua utilidade no jogo.

<ADALOVE - Definir as funções matemáticas que serão utilizadas no jogo>

→ As funções matemáticas utilizadas no **mini game** estão presentes no PULO do personagem e na GRAVIDADE exercida nele. (O game principal ainda não possui estruturas que exijam funções matemáticas)

O exemplo matemático usado foi do Lançamento Vertical (sendo esse o do PERSONAGEM) e o Lançamento Horizontal (que tecnicamente seria do CENÁRIO, o qual se move naturalmente ao decorrer do jogo), sendo um exemplo bem parecido com o jogo do “Dinossauro do Google”, de quando cessa a internet.

EQUAÇÕES MATEMÁTICAS: **1. Verticais** - Pulo do personagem / Gravidade exercida **2. Horizontais** - Cenário

- Obstáculos e objetos: O personagem estará na posição X para uma velocidade de 6.3 do cenário. Isso formará uma escala de 1: 6.3, pois a cada posição da personagem o cenário estará rolando a esta velocidade.

O jogador deve fazer o caminho de entrar no jogo, onde ele será apresentado com o menu principal, e clicando em “Começar Jogo”, ele é levado a cena dos 3 poderes, onde ele terá que clicar no prédio do poder executivo, clicar no botão de nova proposta, clicar no botão saúde, onde ele será transportado de volta para o menu do executivo, e então ele poderá clicar no botão “Imprensa” e entrar no jogo “Corrida Pela Saúde”.

No minigame, quando o usuário pressionar a barra de espaço, o código entende que a função “Pulo” foi apertada, o código faz uma passagem de estados (do estado “CORRER” para o estado “PULAR”) e executa o seguinte código

```
velocidade = Vector2.ZERO  
velocidade.y -= forca_pulo  
animacao.play("Pulo") e
```

---

```

1  extends KinematicBody2D
2
3  var velocidade = Vector2.ZERO
4  >| ## Criamos a variável velocidade, e como o jogo é um "auto-runner" (corre
5  >| ## automaticamente), nós queremos que a velocidade inicial seja 0, logo,
6  >| ## utilizamos do Vector2, que é a referência cartesiana do Godot.
7
8  export var forca_pulo = 600.0
9  export var gravidade = 20.0
10 >| ## Utilizamos variáveis exportadas pois queremos alterar elas sem ter que
11 >| ## voltar no script, fazendo assim com que elas possam ser alteradas
12 >| ## diretamente pelo "Inspector", embaixo de "Script Variables". Esse método
13 >| ## é de comum utilidade em testes.
14
15 enum {
16 >| ## enum é a função de enumerador. Estamos utilizando dela para fazer 3 estados
17 >| ## possíveis em que o jogador pode se encontrar; Pulando, Correndo
18 >| ## ou no meio do ar.
19 >| PULAR,
20 >| CORRER,
21 >| AR
22 }
23
24 var estado = CORRER
25

```

```

26 onready var animacao = $AnimatedSprite
27 >| ## Identificamos o nó de animação no código utilizando o símbolo do sifrão
28 >| ## e o guardamos em uma variável
29
30 func _ready():
31 >| pass
32
33 func _physics_process(delta):
34 >| ## Função do Godot que cuida das propriedades físicas de acordo com o
35 >| ## FPS (Frames por segundo)
36 match estado:
37 >| >| ## Aqui definimos o que cada estado fará em questão de código
38 >| >| CORRER:
39 >| >| >| animacao.play("Corrida")
40 >| >| >| ## Animação para o personagem correr.
41 >| >| PULAR:
42 >| >| >| velocidade = Vector2.ZERO
43 >| >| >| ## Aqui, é bom ter um reset da velocidade, para que o pulo não saia
44 >| >| >| ## como algo na diagonal.
45 >| >| >| velocidade.y -= forca_pulo
46 >| >| >| ## Como já foi explicado, o plano cartesiano se encontra "invertido"
47 >| >| >| ## então, para irmos para cima, precisamos subtrair a força do pulo.
48 >| >| >| animacao.play("Pulo")
49 >| >| >| ## Utilizamos o bloco de animação que foi criado no

```



```

49 >| >| >| >| ## Utilizamos o bloco de animação que foi criado no
50 >| >| >| >| ## Animated Sprite > Frames > Sprite Frames, e chamado de
51 >| >| >| >| ## "Pulo".
52 >| >| >| estado = AR
53 >| >| >| >| ## Se o jogador pulou, ele está no meio do ar.
54 >| >| >| AR:
55 >| >| >| pass
56 >| velocidade.y += gravidade
57 >| >| ## Em 2D, graças a inversão do Godot, caso tenhamos um y positivo,
58 >| >| ## nosso personagem se encontrará na direção abaixo, e caso negativo,
59 >| >| ## ele se encontrará acima. Logo, o efeito da gravidade será positivo
60 >| >| ## ao invés de negativo, então utilizamos a soma.
61 >| move_and_collide(velocidade * delta)
62 >| >| ## O "move_and_collide" serve para realizar o movimento do personagem,
63 >| >| ## e aqui, multiplicamos a velocidade pelo delta, que é uma maneira de
64 >| >| ## fazer com que a velocidade escale de acordo com o FPS
65
66 >| func _input(event):
67 >| ## Detecta qualquer input que entra, dentro de seu próprio loop no Godot.
68 >| ## Aqui, estamos utilizando ela para eventos, com a função básica de
69 >| ## assistir aos botões do teclado.
70 >| if estado == CORRER:
71 >| >| ## Se o jogador estiver no chão
72 >| >| if event.is_action_pressed("Pulo"):
73 >| >| >| ## Adicionamos uma condição que faz com que toda vez que o botão de

```

```

73 >| >| >| ## Adicionamos uma condição que faz com que toda vez que o botão de
74 >| >| >| ## pulo (configurado no menu Project > Project Settings > Input Map)
75 >| >| >| ## for pressionado, algo aconteça.
76 >| >| >| estado = PULAR
77 >| >| >| >| ## Se o jogador pressionar o botão de pular, ele está pulando.
78
79
80 →| func _on_Pes_body_entered(body):
81 >| ## Aqui, o nó "Pes" checa se algum outro corpo de colisão entrou em
82 >| ## contato com ele. Neste caso, o nó representa o pé do jogador. E nossa
83 >| ## intenção é saber se os pés dele estão no chão.
84 →| if body is StaticBody2D:
85 >| >| ## Aqui nós checamos se o corpo que está em contato com os pés do
86 >| >| ## jogador é o StaticBody2D, que, na cena "Frente" representa a caixa
87 >| >| ## de colisão do chão.
88 >| >| estado = CORRER
89 >| >| >| ## Se os pés estão no chão, o jogador está correndo.
90
91
92 →| func _on_Pes_body_exited(body):
93 >| ## Aqui, o nó "Pes" checa se algum corpo que estava em colisão com ele
94 >| ## deixou de estar. Neste caso, o nó representa o pé do jogador. E nossa

```

```

92 →| func _on_Pes_body_exited(body):
93 >| ## Aqui, o nó "Pes" checa se algum corpo que estava em colisão com ele
94 >| ## deixou de estar. Neste caso, o nó representa o pé do jogador. E nossa
95 >| ## intenção é saber se os pés dele saíram do chão.
96 →| if body is StaticBody2D:
97 >| >| ## Aqui nós checamos se o corpo que parou de entrar em contato com os pés ## do jogador é o StaticBody2D, que, na cena "Frente" representa a
98 >| >| ## caixa de colisão do chão.
99 >| >| estado = PULAR
100 >| >| >| # Se os pés estiverem no ar, o jogador está pulando.
101

```

---

Utilizamos da documentação de Shaders do Godot

([https://docs.godotengine.org/en/3.0/tutorials/shading/shading\\_language.html](https://docs.godotengine.org/en/3.0/tutorials/shading/shading_language.html)) para fazer o código.

- A função utiliza de uma variável (velocidade rolamento) para utilizar de incremento no código, utilizando de UVs (mapeamento de pixels), guardados na variável u, incrementados no eixo X, e ocorrendo de acordo com o tempo (TIME)

```

1  ▾ shader_type canvas_item;
2  ▸  /*
3  ▸  Aqui definimos o tipo de Shader, o tipo de Shader para jogos 2D é
4  ▸  sempre canvas_item no Godot.
5  ▸  */
6
7  ▾ uniform float velocidade_rolamento;
8  ▸  /*
9  ▸  Aqui temos uma variável uniforme, do tipo de dado float, onde o que
10 ▸ declaramos é a velocidade de rolamento da tela. A função dessa linha de
11 ▸ código é, basicamente, para mudar o valor no "inspector", e a partir daí,
12 ▸ poderemos manipular o script tanto pelo inspector, quanto pelo código.
13 ▸ */
14 ▾ void fragment()
15 ▸  /*
16 ▸ Aqui utilizamos a shader "fragment", que tem como função executar em cada pixel
17 ▸ da cena que contem o nó "FundoInfinito", e então fará algumas modificações
18 ▸ para que ela consiga mover os UVs (Um mapeamento de onde certos pixels da
19 ▸ textura estão localizados no componente).
20 ▸ */
21 ▾ {

```

```

22 ▾ ▸ vec2 u = UV;
23 ▸  /*
24 ▸  Aqui estamos pegando o UV e o colocando em uma variável
25 ▸  */
26 ▾ ▸ u.x += velocidade_rolamento * TIME;
27 ▸  /*
28 ▸  Permite a velocidade do rolamento ser incrementada conforme o tempo, no eixo X
29 ▸  */
30 ▾ ▸ vec4 cor = texture(TEXTURE, u);
31 ▸  /*
32 ▸  Aqui utilizamos vec4, que é a cor, e armazenamos em uma variável, e então
33 ▸  a colocamos na textura, utilizando a coordenada UV. Basicamente, pegamos um
34 ▸  fragmento da textura.
35 ▸  */
36 ▾ ▸ COLOR = cor;
37 ▸  /*
38 ▸  Aqui, atribuímos a variável de volta a cor, utilizando o fragmento que
39 ▸  pegamos. Funcionalmente, resetando o plano de fundo.
40 ▸  */
41 ▾ }

```

## 6.2 Cinemática Unidimensional

Quais grandezas da cinemática são usadas no jogo desenvolvido neste projeto?

Coloque os trechos do programa no Godot onde elas aparecem e explique sua utilidade no jogo.

<ADALOVE - Aplicar os conceitos matemáticos no jogo>

Explicação das Grandezas apresentadas no código:

→ A grandeza da cinemática utilizada no jogo é a velocidade como vista no código abaixo, onde a força do pulo e da gravidade implicam diretamente na velocidade do movimento e determinam-a, fazendo-as parte das grandezas da cinemática.

```
export var forca_pulo = 650.0
export var gravidade = 25.0
> ## Utilizamos variáveis exportadas pois queremos alterar elas sem ter que voltar no script
```

## 6.3 Vetores

Quais vetores são usados no jogo desenvolvido neste projeto?

Coloque os trechos do programa no Godot onde eles aparecem e explique sua utilidade no jogo.

*Obs.: Anexar ao relatório as atividades realizadas em aula (prints, fotos, etc.)*

Aqui usamos dois vetores no nosso mini game onde um deles criamos a variável velocidade, e como o jogo é um "auto-runner" (correr automaticamente), nós queremos que a velocidade inicial seja 0, logo, utilizamos do Vector2, a referência cartesiana do Godot. e já no outro em função a ação do personagem para ter um reset da velocidade, para que o pulo não saia como algo na diagonal.

```
44 > > > PULAR:
45 > > > velocidade = Vector2.ZERO
46 > > > ## Aqui, é bom ter um reset da velocidade, para que o pulo não saia como algo na diagonal.
7 var velocidade = Vector2.ZERO
8 > ## Criamos a variável velocidade, e como o jogo é um "auto-runner" (corre automaticamente), nós queremos que a velo
9
```

<ADALOVE - Definir as funções matemáticas que serão utilizadas no jogo>

## 6.4 Cinemática Bidimensional e mais

Quais as grandezas físicas (não trabalhadas nos encontros anteriores) são usadas no jogo desenvolvido neste projeto?

Coloque os trechos do programa no Godot onde elas aparecem e explique sua utilidade no jogo.

Utilizamos a Gravidade, Força pulo e Velocidade

```
3 var pontos = 0
4 » ## Variável para manter conta dos pontos.
5
6
7 var velocidade = Vector2.ZERO
8 » ## Criamos a variável velocidade, e como o jogo é um "auto-runner" (corre automaticamente)
9
10 export var forca_pulo = 650.0
11 export var gravidade = 25.0
12 » ## Utilizamos variáveis exportadas pois queremos alterar elas sem ter que voltar no script
13
14
15
16 enum {
17 » ## enum é a função de enumerador. Estamos utilizando dela para fazer 3 estados possíveis e
18 » PULAR,
19 » CORRER,
20 » AR
21 }
```

```
37 func _physics_process(delta):
38 » ## Função do Godot que cuida das propriedades físicas de acordo com o FPS (Frames por segun
39 » match estado:
40 » » ## Aqui definimos o que cada estado fará em questão de código
41 » » CORRER:
42 » » » animacao.play("Corrida")
43 » » » ## Animação para o personagem correr.
44 » » PULAR:
45 » » » velocidade = Vector2.ZERO
46 » » » ## Aqui, é bom ter um reset da velocidade, para que o pulo não saia como algo
47 » » » velocidade.y -= forca_pulo
48 » » » ## Como já foi explicado, o plano cartesiano se encontra "invertido" então, pa
49 » » » animacao.play("Pulo")
50 » » » ## Utilizamos o bloco de animação que foi criado no Animated Sprite > Frames >
51 » » » estado = AR
52 » » » ## Se o jogador pulou, ele está no meio do ar.
53 » » AR:
54 » » » pass
55 » velocidade.y += gravidade
56 » ## Em 2D, graças a inversão do Godot, caso tenhamos um y positivo, nosso personagem se
57 » move_and_collide(velocidade * delta)
58 » ## O "move_and_collide" serve para realizar o movimento do personagem, e aqui, multipl
59 »
```

```

60 ▾ func _input(event):
61     » ## Detecta qualquer input que entra, dentro de seu próprio loop no Godot. Aqui, estamos ut
62 ▾ » if estado == CORRER:
63     »     » ## Se o jogador estiver no chão
64 ▾ »     » if event.is_action_pressed("Pulo"):
65     »     »     » ## Adicionamos uma condição que faz com que toda vez que o botão de pulo (configur
66     »     »     » estado = PULAR
67     »     »     » ## Se o jogador pressionar o botão de pular, ele está pulando.
68
69
70 ▾ func _on_Pes_body_entered(body):
71     » ## Aqui, o nó "Pes" checa se algum outro corpo de colisão entrou em contato com ele. Neste
72 ▾ » if body is StaticBody2D:
73     »     » ## Aqui nós checamos se o corpo que está em contato com os pés do jogador é o StaticBo
74     »     » estado = CORRER
75     »     »     » ## Se os pés estão no chão, o jogador está correndo.
76
77
78 ▾ func _on_Pes_body_exited(body):
79     » ## Aqui, o nó "Pes" checa se algum corpo que estava em colisão com ele deixou de estar. Ne
80 ▾ » if body is StaticBody2D:
81     »     » ## Aqui nós checamos se o corpo que parou de entrar em contato com os pés do jogador é
82     »     » estado = PULAR
83     »     »     » ## Se os pés estiverem no ar, o jogador está pulando.
84

```

<ADALOVE - Fazer orientação espacial (2D)>

---

## 7. Bibliografias

Toda referência citada no texto deverá constar nessa seção, utilizando o padrão de normalização da ABNT). As citações devem ser confiáveis e relevantes para o trabalho. São imprescindíveis as citações dos *sites de download* das ferramentas utilizadas, bem como a citação de algum objeto, música, textura ou outros que não tenham sido produzidos pelo grupo, mas utilizados (mesmo no caso de licenças gratuitas, *royalty free* ou similares).

- Links de conteúdo:
- Plataformas Utilizadas:
- Materiais de Exemplos:
- Texturas e designs:
- Ferramentas de programação e desenvolvimento de conteúdo:

---

# Apêndice

Os apêndices representam informações adicionais que não caberiam no documento exposto acima, mas que são importantes por alguma razão específica do projeto. Em geral, os apêndices do GDD podem incluir os rascunhos das fases, outros *concept arts* do jogo, diagramas diversos etc.