

UNIVERSIDADE ESTADUAL DO MARANHÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO E  
SISTEMAS

LEONARDO FRAZÃO XAVIER

**JAI: PROCESSO DE DESENVOLVIMENTO DE JOGOS DIGITAIS  
ACESSÍVEIS AO PÚBLICO IDOSO**

São Luís - Maranhão

2020

LEONARDO FRAZÃO XAVIER

**JAI: PROCESSO DE DESENVOLVIMENTO DE JOGOS DIGITAIS  
ACESSÍVEIS AO PÚBLICO IDOSO**

Dissertação apresentada à Universidade Estadual do Maranhão, como parte das exigências do Programa de Pós-Graduação em Engenharia da Computação e Sistemas, para obtenção do título de Mestre em Engenharia da Computação.

*Universidade Estadual do Maranhão - UEMA*

*Centro de Ciências Tecnológicas*

*Programa de Pós-Graduação em Engenharia de Computação e Sistemas*

*Orientadora: Prof. Dra. Eveline de Jesus Viana Sá*

São Luís - Maranhão

2020

Xavier, Leonardo Frazão.

JAI: processo de desenvolvimento de jogos digitais acessíveis ao público idoso / Leonardo Frazão Xavier. – São Luís, 2020.

88 f

Dissertação (Mestrado) – Curso de Engenharia de Computação e Sistemas, Universidade Estadual do Maranhão, 2020.

Orientadora: Profa. Dra. Eveline de Jesus Viana Sá.

1.Acessibilidade. 2.Idoso. 3.Jogos digitais. 4.Processo de software. I. Título.

CDU: 004.4-053.9

LEONARDO FRAZÃO XAVIER

**JAI: PROCESSO DE DESENVOLVIMENTO DE JOGOS DIGITAIS  
ACESSÍVEIS AO PÚBLICO IDOSO**

Dissertação apresentada à Universidade Estadual do Maranhão, como parte das exigências do Programa de Pós-Graduação em Engenharia da Computação e Sistemas, para obtenção do título de Mestre em Engenharia da Computação.

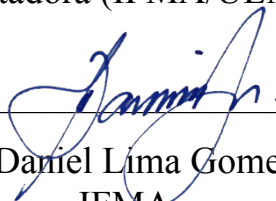
São Luís, 06 de novembro de 2020.

BANCA EXAMINADORA



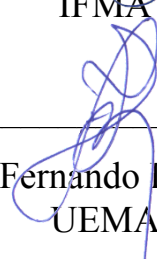
---

Prof. Dra. Eveline de Jesus Viana Sá  
Orientadora (IFMA/UEMA)



---

Prof. Dr. Daniel Lima Gomes Júnior  
IFMA



---

Prof. Dr. Antonio Fernando Lavareda Jacob Junior  
UEMA

## SUMÁRIO

<b>RESUMO</b>	<b>7</b>
<b>ABSTRACT</b>	<b>8</b>
<b>1. Caracterização do Problema</b>	<b>9</b>
<b>2. Objetivos e Metas</b>	<b>10</b>
<b>3. Metodologia</b>	<b>10</b>
3.1 Pesquisa	10
3.2 Construção	11
3.3 Revisão	11
3.4 Reconstrução	11
3.5 Polimento	12
<b>4. Justificativa</b>	<b>12</b>
<b>5. Fundamentação Teórica</b>	<b>14</b>
5.1 O Público idoso	14
5.2 Modelos de Processo de Softwares	15
5.3 Engenharia de Software no Desenvolvimento de Jogos	18
5.4 Acessibilidade digital para Idosos	20
5.5 Métodos de Testes para jogos	21
<b>6. Trabalhos Relacionados</b>	<b>24</b>
6.1 Recomendações de Acessibilidade para Idosos	24
6.2 Modelos e Processos de Desenvolvimento de Jogos	25
<b>7. Processo de Desenvolvimento de Jogos Digitais Acessíveis ao Público Idoso</b>	<b>28</b>
7.1 Critérios para Construção do Processo JAI	28
7.2 Visão Geral do Processo JAI	29
7.3. Iniciação	31
7.3.1 Público-alvo	32
7.3.2 Conceito aproximado	33
7.3.3 Ferramentas de trabalho	33
7.3.4 Estratégias de monetização, marketing, sistema de atualização e suporte ao cliente	34
7.3.5 Mercado	35
7.3.6 Análise de viabilidade	35
7.3.7 Integração da acessibilidade	36
7.3.8 Modelagem inicial do projeto e cronograma de atividades	37
7.4 Pré-produção	37
7.4.1 Diretrizes de acessibilidade para fase de pré-produção	39
7.4.2 Seleção dos participantes dos testes	39

7.4.3 Protótipo Interno	39
7.4.4 Protótipo Estrutural	40
7.4.5 Levantamento de requisitos da plataforma de distribuição	41
7.4.6 Backlog do produto	42
7.4.7 Nível de release	43
7.4.8 Revisão da modelagem	43
7.5 Produção	44
7.5.1 Os ciclos de produção	46
7.5.2 Diretrizes de acessibilidade para fase de produção	47
7.5.3 Produção dos níveis de release	47
7.5.4 Testes internos e testes de aceitação	48
7.5.5 Atividade paralelas	50
7.6 Beta	51
7.6.1 Preparação dos testes	53
7.6.2 Publicação Beta	53
7.6.3 Correção de bugs	53
7.6.4 Módulos adicionais	54
7.6.5 Polimentos	54
7.7 Publicação	54
7.7.1 Publicação oficial	55
7.7.2 Diretrizes de acessibilidade para o plano de manutenção	56
7.7.3 Planos de manutenção e expansão	56
7.7.4 Compartilhamento de conhecimento (post mortem)	56
<b>8. Sequência Simplificada do Processo JAI</b>	<b>57</b>
8.1 FASE DE INICIAÇÃO: Criando documentação inicial do jogo	57
8.2 FASE DE PRÉ-PRODUÇÃO: Prototipagem e revisão da modelagem	58
8.3 FASE DE PRODUÇÃO: Construindo o jogo através de ciclos iterativos	59
8.4 FASE BETA: Trabalhando com a versão beta do jogo	59
8.5 FASE DE PUBLICAÇÃO: Lançamento oficial	60
<b>9. Resultados e Discussões</b>	<b>60</b>
9.1 Análise Comparativa	60
Ramadan (2013)	60
Hendrick (2009)	63
Blitz Game Studio (2020)	64
<b>10. Considerações Finais</b>	<b>67</b>
<b>REFERÊNCIAS</b>	<b>69</b>
<b>APÊNDICE A &lt;MODELO DE DOCUMENTO DE PROJETO DE JOGO (GDD)&gt;</b>	<b>74</b>
<b>APÊNDICE B &lt;INDICAÇÕES DE DIRETRIZES DE ACESSIBILIDADE PARA IDOSOS&gt;</b>	<b>81</b>
1. DIRETRIZES PARA FASE DE PROTOTIPAGEM	81

2. DIRETRIZES PARA FASE DE PRODUÇÃO	<b>82</b>
3. DIRETRIZES PARA PUBLICAÇÃO E MARKETING	<b>86</b>
4. DIRETRIZES PARA O PLANO DE MANUTENÇÃO DO JOGO	<b>86</b>

## RESUMO

A população mundial vem envelhecendo e os idosos estão participando cada vez mais das atividades do dia a dia de uma sociedade, produzindo, consumindo, movimentando a economia e compartilhando momentos. Porém o avançar da idade debilita o corpo humano, agrava os problemas de saúde e enfraquece as capacidades físicas e cognitivas. Isso faz com que até os simples afazeres da vida, como ir ao banco, fazer compras ou usar um aparelho eletrônico da casa, sejam cada vez mais difíceis de executar e causem sensação de frustração, o que pode levar o indivíduo ao isolamento e depressão.

Existem esforços em produzir tecnologias que ajudam os idosos no cuidado com a saúde, na realização das atividades domésticas, no retorno ao convívio social e na promoção do bem-estar. Alguns pesquisadores vêem nos jogos eletrônicos uma ferramenta para diversão, reabilitação física e mental de pessoas idosas, pois alguns jogos podem oferecer desafios que despertam habilidades cognitivas e reflexos físicos que vinham sendo perdidos.

Para contribuir com a produção de tecnologias acessíveis aos idosos, este trabalho concentra-se em conhecer os processos de criação de jogos e como os produtores de conteúdo digital estão trabalhando para remover as barreiras que as novas tecnologias impõem para aqueles com menos familiaridade. Neste contexto, este trabalho propõe um processo de criação de jogos digitais que une características de desenvolvimento ágil à necessidade de criar produtos que alcance a população idosa.



## **ABSTRACT**

The world population is aging and they are increasingly participating in social activities, producing, consuming, moving the economy and sharing moments. However, advancing age weakens the human body, aggravates health problems and decreases physical and cognitive abilities. This makes even the simple chores of life, such as going to the bank, shopping or using an electronic device, become increasingly difficult to perform and cause a sense of frustration, which can lead the individual to isolation and depression.

There are efforts to produce technologies that help the elderly in health care, carrying out domestic activities, returning to social life and promoting well-being. Some researchers see electronic games as a playful tool and capable of offering physical and mental rehabilitation to the elderly, as some games can offer challenges that awaken lost cognitive skills and physical reflexes.

To contribute to the production of technologies accessible to the elderly, this work focuses on researching the processes of creating games and how digital content producers are working to remove the barriers that new technologies impose to the oldery. In this context, this work proposes a process for creating digital games that combines features of agile software development and the desire to create products that reach the elderly population.

## 1. Caracterização do Problema

Segundo a Organização Mundial de Saúde (OMS, 2018a), a expectativa de vida da população continua aumentando e em 2020 o número de pessoas com 60 anos ou mais superará a quantidade de crianças menores de 5 anos. Além disso, a população com mais de 60 anos, que em 2015 era cerca de 900 milhões, deverá alcançar a marca de 2 bilhões até 2050. Essa população idosa costuma estar mais propensa ao surgimento de doenças crônicas e distúrbios mentais e neurológicos. Por estas e outras razões, o envelhecimento saudável e o bem-estar da terceira idade tornou-se um desafio de saúde pública global.

Diante deste cenário, os jogos digitais podem desempenhar um papel positivo melhorando o bem-estar físico e mental dos idosos, aumentando suas conexões sociais e oferecendo um meio agradável de aproveitar o tempo (IJSELSTEIJN et al., 2007). Segundo Tarouco e outros (2014), os jogos digitais também podem proporcionar a melhora na flexibilidade cognitiva, pois funcionam como uma ginástica mental, aumentando a rede de conexões neurais e alterando o fluxo sanguíneo no cérebro quando em estado de concentração.

Como outros softwares, o processo de desenvolvimento de um jogo requer a aplicação de técnicas de engenharia de software para estruturar de forma racional cada etapa da criação de um jogo: da concepção abstrata à entrega do produto final e manutenção.

Em seu trabalho de revisão sistemática sobre engenharia de softwares para desenvolvimento de jogos (do inglês Game Development Software Engineering - GDSE), Aleem (2016) identifica uma série de modelos de GDSE, desde aqueles que herdam de técnicas tradicionais como o Cascata, como também os já derivados das técnicas ágeis como XP e SCRUM. Entretanto, estas técnicas dedicam pouco (ou nenhum) espaço para avaliação e adaptação de acessibilidade dos jogos ao público que sofre de limitações, e isso acaba por dificultar a inclusão de idosos ao contexto das tecnologias e entretenimento, já que a idade avançada impõe restrições e dificuldades em lidar com recursos tecnológicos, havendo a necessidade de uma atenção especial no desenvolvimento de jogos digitais direcionados a adultos mais velhos (Machado & Ishitani, 2015).

No esforço de usar a GDSE para produzir conteúdo inclusivo, este trabalho propõe o processo JAI (Jogos Acessíveis aos Idosos), que orienta criadores de jogos digitais a produzirem jogos que atendam às demandas de acessibilidade do público idoso e de portadores de deficiências.

## **2. Objetivos e Metas**

### **2.1 Objetivos Gerais**

O objetivo geral deste trabalho é elaborar um processo que orienta criadores de jogos digitais a produzirem jogos que atendam às demandas de acessibilidade do público idoso.

### **2.2 Objetivos Específicos**

1. Servir como referencial para a criação de jogos que contemplem as necessidades do público idoso;
2. Propor um conjunto de orientações que norteiam o processo proposto;
3. Explorar os métodos de testes de jogos e orientar como aplicá-los durante a criação de um jogo;
4. Tornar o processo adaptável a projetos de baixa complexidade e a estúdios de pequeno porte.

## **3. Metodologia**

O desenvolvimento deste trabalho se passou essencialmente por cinco fases: pesquisa, construção, revisão, reconstrução, polimento.

### **3.1 Pesquisa**

Primeiro passo foi a análise de diversas bibliografias que abordam a aplicação da engenharia de software ao processo de desenvolvimento de jogos digitais. Esse levantamento inicial de dados serviu para identificar as características predominantes de cada modelo e processo de desenvolvimento de jogos e quais destas podem agregar maior valor ao produto final almejado.

Ainda na fase de pesquisas, foi necessário reunir conhecimentos sobre a forma como a acessibilidade vem sendo trabalhada em jogos e quais proposições de pesquisadores para melhorar essa característica. Trabalhos como as heurísticas para avaliação de gameplay de Machado & Ishitani (2015), as diretrizes de acessibilidade para jogos digitais (CHEIRAN, 2013) e o documento online *Game Accessibility Guidelines* (2020a) ajudaram a compor um conjunto de recomendações para que os jogos digitais atendam certo grau de acessibilidade.

### 3.2 Construção

O segundo momento consiste da compilação das informações encontradas para compor a base da proposta, além de planejar como as diretrizes de acessibilidade podem ser adaptadas naturalmente ao processo sem torná-las destoantes às demais atividades envolvidas.

A figura 3.1 é uma abstração de alto nível do primeiro do processo proposto.

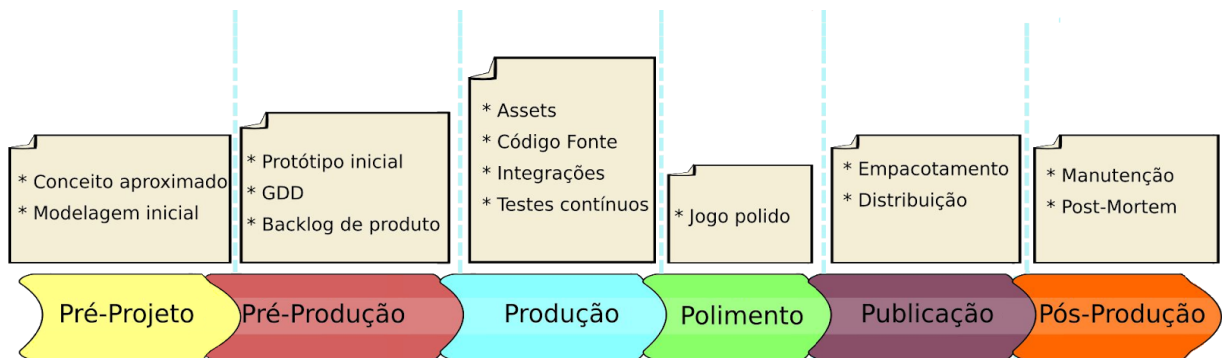


Figura 3.1. Primeira estrutura de processo da proposta. Fonte: do autor.

### 3.3 Revisão

A primeira versão do processo foi reprovada quanto a flexibilidade e interatividade entre as fases do processo. Sua estrutura linear acabou tornando difícil a associação das diretrizes de acessibilidade com o processo, uma vez que não havia espaço para revisão contínua das diretrizes propostas e nem definições de quais diretrizes seriam implementadas e em quais momentos isso seria feito.

Portanto, a fase de revisão da proposta envolveu mais estudos sobre modelos de processo e diretrizes de acessibilidade, o que ajudou a formalizar o conhecimento sobre as prioridades das adaptações de acessibilidade e os momentos mais adequados para testá-las.

### 3.4 Reconstrução

A segunda estrutura do processo, representada através da figura 3.2, possui apenas cinco fases e as atividades de modelagem e testes foram evidenciadas para a fim de refinar o máximo da usabilidade do jogo. Neste ponto, o processo já descrevia muito bem quando e como as diretrizes de acessibilidade podiam e deviam ser implementadas e testadas.

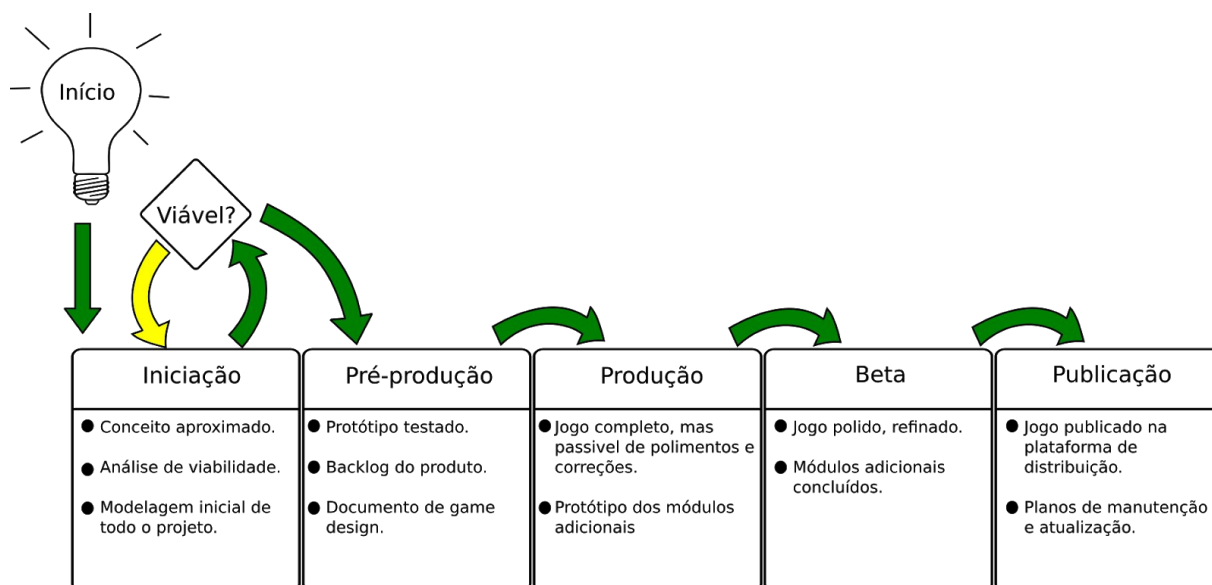


Figura 3.2. Segunda estrutura de processo da proposta. Fonte: do autor.

### 3.5 Polimento

Foi chamado de polimento o trabalho de refinamento do processo proposto. A maior alteração realizada foi trazer o elemento público-alvo para o centro de toda a discussão do processo. Todas as etapas passaram a considerar fortemente o público para o qual o jogo está sendo criado. O ponto de partida do processo passou a ser a definição do público-alvo e as atividades de testes foram melhor especificadas para ajudar o usuário do processo a assegurar a qualidade do jogo.

A seção 7 deste trabalho descreve detalhadamente cada fase do processo e orienta como cada atividade pode ser realizada.

### 4. Justificativa

A população mundial está envelhecendo e a expectativa da OMS (2018a) é que até 2020 a população com mais de 60 anos supere a quantidade de crianças de até 5 anos de idade. No Brasil, a projeção da população divulgada em 2018 pelo Instituto Brasileiro de Geografia e Estatística (IBGE, 2018) indica que em 2043, um quarto da população brasileira deverá ter mais de 60 anos, enquanto a proporção de jovens até 14 anos será de apenas 16,3%.

Em pesquisa publicada no “Health Science Journal”, Roupa et al. (2010) afirma que o envelhecimento causa a dedução das funções biológicas e cognitivas, levando os idosos a exaustão e perda de paciência mais facilmente. Roupa et al. (2010) conclui que as pessoas

mais velhas precisam de apoio e orientação adequados para enfrentar satisfatoriamente as dificuldades da vida cotidiana, como a constante evolução da tecnologia.

A OMS (2005) chamou de “Envelhecimento Ativo” o conjunto de estratégias anunciadas para se alcançar um envelhecimento saudável da população. Neste documento a OMS evidencia a importância do estímulo físico e cognitivo como forma de fortalecer a contínua participação dos idosos em questões sociais, econômicas e espirituais. Em 2015, quando a OMS definiu o foco dos seus trabalhos sobre o envelhecimento para os anos entre 2015 e 2030, o termo “Envelhecimento Ativo” foi atualizado para “Envelhecimento Saudável” (OMS, 2018b). Em sua pesquisa sobre as formas como os jogos digitais podem trabalhar com o envelhecimento ativo, Neto (2014) descreve uma série de projetos que foram desenvolvidos com auxílio de jogos do Nintendo Wii como suplemento para tratamento fisioterapêutico. Neto (2014) também cita trabalhos envolvendo jogos digitais do gênero “*puzzle*” (tipo de jogos de quebra-cabeça) que apontaram resultados positivos no retardo do declínio cognitivo em público entre 50 e 64 anos.

No campo de engenharia de software, muitos são os esforços para construir metodologias que formalizam o processo de desenvolvimento de jogos digitais. Aleem (2016) identificou 21 diferentes processos no ciclo de vida da criação de jogos, sendo 8 destes na fase de pré-produção, 7 na fase de produção e 6 na pós-produção. Enquanto apenas o processo de “especificação de requisitos” (pré-produção) e os processos de testes (pós-produção) envolvem a participação do público-alvo na criação do jogo, Braad (2016) enfatiza a importância de aproximação do jogador alvo às etapas de criação do jogo e considera no projeto as informações do contexto em que o jogo será utilizado. Apesar de não terem sido encontradas metodologias de desenvolvimento direcionadas para o público idoso, existem guias e diretrizes que norteiam o produto final e servem como métricas para adequação dos jogos às necessidades dos idosos.

Buscando promover a maior inclusão da população idosa às novas tecnologias, este trabalho visa reunir conhecimentos das bibliografias já produzidas sobre a GDSE em um processo de desenvolvimento de jogos que atenda às especificidades do público idoso (JAI), bem como especificar um conjunto de orientações norteadoras desse processo. Obtendo êxito, o processo também atenderia paralelamente demais públicos portadores de sintomas similares aos advindos da terceira idade: baixa visão, dificuldades na coordenação motora, déficit cognitivos e outros.

## **5. Fundamentação Teórica**

Este capítulo inicia oferecendo detalhes sobre o público de idosos hoje no mundo, as perspectivas para o aumento da sua parcela na população global, as necessidades de mudanças e adequações para atendê-los e algumas tecnologias que já vem sendo implantadas para melhorar a qualidade de vida dessas pessoas na terceira idade. Continuando na seção 5.2, o leitor encontrará um apanhado técnico sobre a engenharia de software e processos de desenvolvimento de softwares. A seção 5.3 discorre sobre a engenharia de software aplicada na criação de jogos digitais. A seção 5.4 discute sobre como as deficiências dos idosos vem sendo tratada pela engenharia de software e criadores de conteúdos digitais. E o final do capítulo 5 traz breves descrições sobre métodos de teste que podem ser utilizados no processo proposto.

### **5.1 O Público idoso**

As pessoas ao redor do mundo estão vivendo mais e estes idosos passaram a representar uma parcela muito significativa da população (mais de 900 milhões em 2015 e expectativa de ultrapassar os 2 bi até 2050). Em seus artigos, a Organização Mundial da Saúde refere-se à população idosa como aquela composta por indivíduos com 60 anos ou mais. Mas outras organizações já atualizaram essa definição considerando o ganho de longevidade da população, como é o caso da Organização das Nações Unidas (ONU, 2019) e da Organização para Cooperação e Desenvolvimento Econômico (OCDE, 2020) que passaram a tratar idosos apenas os indivíduos com 65 anos ou mais.

Todo ser humano devia ter a oportunidade de viver uma vida longa e saudável, entretanto o ambiente a nossa volta pode favorecer ou prejudicar a saúde durante essa jornada. A exposição a fatores de risco como poluição e violência, acesso a serviços de saúde e de cuidado social e oportunidade de participação ativa na sociedade afetam diretamente a qualidade de vida da população e as chances de um envelhecimento saudável.

A OMS (2018b) define envelhecimento saudável como “o processo de desenvolvimento e manutenção da capacidade funcional que possibilita o bem-estar na terceira idade”. A capacidade funcional é ter as condições que permitam a pessoa ser e fazer aquilo que ela valoriza, como por exemplo: realizar suas necessidades básicas, aprender,

tomar decisões, se movimentar, criar e manter relacionamentos, contribuir com a sociedade, etc.

Alterar a forma como nos comunicamos e realizamos tarefas do dia-a-dia é especialidade das novas tecnologias e que agora buscam atender cada vez mais as necessidades dos mais velhos. A empresa Samsung, uma das líderes entre os provedores de tecnologias para smartphones e IoT, publicou em 2018 um artigo (HOLLAND, 2018) anunciando algumas de suas soluções tecnológicas que vêm sendo desenvolvidas para o público sênior. Entre estas, estão os óculos de realidade virtual melhorados para atender condições específicas de deficiências visuais, como degeneração macular, catarata e miopia severa (LANGLEY, 2018).

Aplicação de Big Data e inteligência artificial (IA) também são aliadas na missão de construir um ambiente favorável ao envelhecimento saudável. Diversas bases de dados sobre doenças, diagnósticos e tratamentos são analisadas por algoritmos de IA para ajudar no diagnóstico e predição de doenças como câncer, doenças visuais e até mesmo Alzheimer; oferecendo maiores taxas de sucesso em curas e tratamentos. (BRITISH COUNCIL, 2019)

Para a criação de um ambiente favorável ao envelhecimento saudável, além da aplicação direta na saúde, as novas tecnologias carregam a oportunidade, e responsabilidade, de incluir o público idoso em todas atividades cotidianas que lhes forem competentes, seja nos trabalhos domésticos, nos relacionamentos sociais, ou mesmo nos momentos de lazer. E inspirados a desfazer as barreiras da segregação digital aos idosos, este trabalho vem propor um processo de desenvolvimento de jogos digitais que atendam mais intimamente as necessidades dos mais velhos e permitam que desfrutem do lazer dos jogos digitais indiscriminadamente.

## **5.2 Modelos de Processo de Softwares**

Quando se trabalha na construção de um sistema (software), é importante seguir um roteiro contendo um passo-a-passo que ajuda na criação de um resultado de alta qualidade. Este roteiro é chamado de processo de software (PRESSMAN, 2015, p. 30).

Pressman (2015, p. 31-34) define que um processo de software possui genericamente 5 atividades estruturais que são comuns a todo projeto: comunicação, planejamento, modelagem, construção e implantação. Estas atividades funcionam como etapas e são divididas em tarefas menores, tais como: análise de requisitos, modelagem do projeto,



construção de programas, testes, suporte e outros. As figuras 5.1 a 5.4 mostram exemplos de como o fluxo das atividades pode ser organizada em um processo de software.

Não há uma regra estática para a aplicação destas atividades e tarefas na criação de softwares. Sommerville (2016, p. 44-45) afirma que os processos de softwares são complexos e, como todo processo criativo e intelectual, contam com as decisões de julgamentos tomados pessoas. Como não há um processo universal que seja ideal para todo tipo de software, a maioria das empresas de softwares desenvolveram seus próprios processos de desenvolvimento.



Figura 5.1 - Fluxo linear. Fonte: Pressman (2015, p. 32).

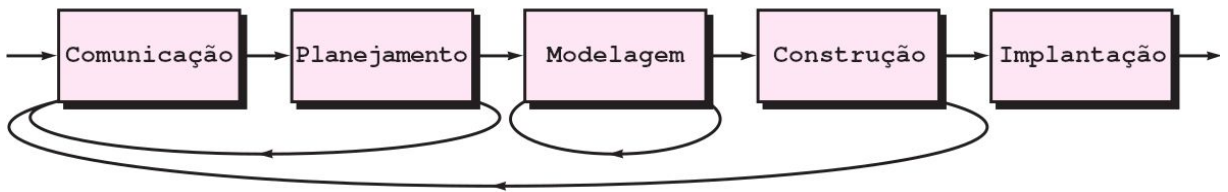


Figura 5.2 - Fluxo iterativo. Fonte: Pressman (2015, p. 32).

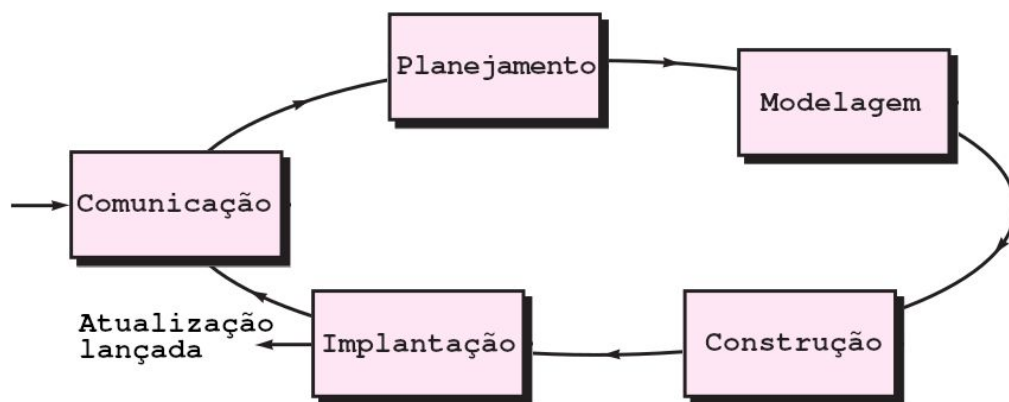


Figura 5.3 - Fluxo evolucionário. Fonte: Pressman (2015, p. 32).

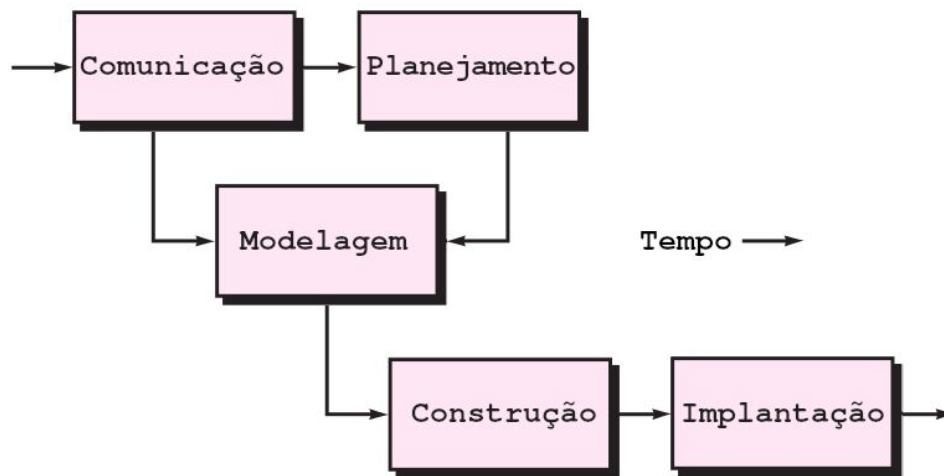


Figura 5.4 - Fluxo paralelo. Fonte: Pressman (2015, p. 32).

O fluxo de um processo linear (Figura 5.1) executa cada uma das atividades estruturais em sequência, enquanto que o fluxo iterativo (Figura 5.2) repete uma ou mais atividades antes de prosseguir para a próxima. O fluxo de processo evolutivo (Figura 5.3) executa as atividades de maneira circular, onde cada ciclo de atividade gera uma versão mais completa do software. E o fluxo de processo paralelo (Figura 5.4) executa uma ou mais atividades em paralelo com outras.

Um modelo de processo de software é uma representação simplificada de um processo de software. Cada modelo de processo representa um processo de uma perspectiva particular e fornece um roteiro que define o fluxo de todas as atividades, ações e tarefas, o grau de interação, os produtos de trabalho e a organização do trabalho que deve ser realizado (PRESSMAN, 2015, p. 40-41).

O modelo cascata é o modelo de processo de software mais antigo da engenharia de software e talvez o mais conhecido também. Ele sugere uma abordagem linear que parte da especificação de requisitos do cliente, prossegue pelas atividades de planejamento, modelagem, construção e implantação, culminando no suporte contínuo do software concluído (PRESSMAN, 2015, p 42).

Entretanto, o fluxo sequencial e enrijecido do modelo cascata o fez receber muitas críticas a ponto de questionarem a real eficácia e aplicabilidade do modelo. Atualmente, o desenvolvimento de softwares é rápido e sujeito a uma sequência imensurável de alterações e o modelo cascata é inapropriado para este trabalho (PRESSMAN, 2015, p 42-43).

Outros modelos surgiram derivados do modelo cascata para suprir sua carência de flexibilidade e aplicabilidade. A adição de novas atividades e a criação de novos fluxos entre as etapas tornaram a próxima geração de modelos mais adaptável ao mercado e mais capazes em atender às diferentes complexidades e demandas dos diversos tipos de softwares que precisam ser produzidos.

Até meados dos anos 2000 já haviam sido criados vários novos modelos de processo, alguns derivados diretamente do cascata, como o incremental e o espiral, e outros derivados dos novos paradigmas como o modelo de método formal, modelo baseados em componentes e o modelo orientado a aspecto. Mas todos esses modelos convencionais carregam com si a filosofia de muita formalização e excesso de documentação (PRESSMAN, 2015, p. 44-54).

Segundo Sommerville (2016, p. 73), as empresas operam em um ambiente global de rápidas mudanças e precisam responder às novas oportunidades e mudanças econômicas. Os softwares, como parte de quase todos os negócios, precisam ser desenvolvidos rapidamente para se obter os benefícios dessas novas oportunidades. A demanda de rápida criação de softwares que sejam capazes de lidar com constantes mudanças incentivou a criação de modelos de processo voltados para produção de softwares rapidamente. Esse desenvolvimento rápido de software tornou-se conhecido como desenvolvimento ágil ou métodos ágeis.

Em 2001, quando Kent Beck e outros desenvolvedores se uniram para criar o que chamamos de Manifesto Ágil do Desenvolvimento de Software, deu-se início aos primeiros passos da engenharia de software ágil. Essa combinação de nova filosofia e diretrizes encoraja a busca pela satisfação do cliente e a entrega rápida do software. As grandes equipes, vasta documentação de software e extensas etapas de modelagem e projeto foram substituídas por equipes pequenas e motivadas, focadas em desenvolver mais software e menos documentos e mantendo um canal de comunicação contínua e ativa entre os desenvolvedores e clientes (PRESSMAN, 2015, p. 68-70).

Neste moderno ambiente de negócios em que vivemos, o desenvolvimento ágil representa uma alternativa aos métodos convencionais e vem atendendo ao ritmo acelerado do mercado que exige constantes mudanças e entrega de sistemas de qualidade com rapidez.

### **5.3 Engenharia de Software no Desenvolvimento de Jogos**

Nos primórdios do desenvolvimento de jogos, um jogo podia ser desenvolvido em poucos meses por apenas uma única pessoa e sem aplicar uma metodologia de

desenvolvimento (um método pré-definido). Mas com a evolução dos hardwares de videogames, um único programador não era mais capaz de produzir um jogo que extraísse o verdadeiro potencial dessas novas máquinas. Então companhias formavam grandes equipes com especialistas multidisciplinares que podiam ter até trinta ou quarenta pessoas envolvidas. Isso aumentou drasticamente os custos de criação dos jogos e os riscos envolvidos. Para gerenciar esses projetos e reduzir riscos, muitas companhias passaram a adotar o Modelo de Cascata já usado em outras indústrias de softwares, entretanto a experiência com o passar dos anos mostrou que o modelo tradicional não se adequa ao desenvolvimento de jogos (KEITH, 2010).

Para o cliente, o real valor do jogo está no potencial de diversão oferecido e para avaliar a diversão de um jogo é preciso jogá-lo. Neste ponto o Modelo Cascata se torna um problema e não uma solução ao processo de criação de jogos, pois a maior parte do esforço investido na “modelagem do jogo” é realizado nas primeiras etapas, enquanto a maioria dos testes só é realizado nas etapas finais. Se a equipe só poder avaliar a diversão do jogo nas etapas finais de criação, pode ocorrer que todo o esforço até este momento tenha produzido um jogo ruim e que requer mudanças no projeto.

A forma como o Modelo Cascata trabalha torna alterações no projeto exponencialmente mais custosas à medida que o projeto se aproxima das etapas finais. O resultado é que muitos projetos não chegam a serem concluídos, ou são entregues estourando os prazos e orçamentos e tornando o jogo não rentável. A experimentação de modelos incrementais passou a ser uma alternativa até 2001 com o surgimento do desenvolvimento ágil (KEITH, 2010).

Os Modelos Ágeis gerenciam melhor os imprevistos que surgem durante qualquer etapa do projeto e sofrem um menor impacto no orçamento e na agenda de produção. A comunicação contínua entre desenvolvedores e clientes, a divisão das tarefas em pequenos ciclos que englobam da modelagem à implantação de cada novo recurso e o espírito de equipe sempre valorizado por estes modelos tornam as equipes flexíveis, adaptáveis e motivadas a superar os obstáculos que surgem em qualquer projeto de jogo digital (KEITH, 2010). Portanto, quando comparados ao modelo tradicional, os modelos ágeis se mostram mais alinhados ao mercado atual de desenvolvimento de jogos, principalmente quando consideramos os desenvolvedores indies, que trabalham com orçamento e equipes limitadas.

Por fim, os métodos ágeis apresentam características muito atraentes para colaborar com os objetivos deste trabalho: a possibilidade de adaptar os métodos ágeis à gerência de equipes pequenas (duas a cinco pessoas), a constante comunicação com o cliente e a capacidade de guiar um projeto incremental com prototipação e testes constantes. Estas e outras filosofias e diretrizes de metodologia ágil foram fatores decisivos para escolha desse modelo de desenvolvimento de software como referência para construção do modelo proposto por este trabalho.

#### **5.4 Acessibilidade digital para Idosos**

Acessibilidade significa evitar barreiras desnecessárias que impedem que pessoas com várias deficiências acessem ou desfrutem de um produto ou serviço (Game Accessibility Guidelines, 2020b). Segundo a OMS (2018b), mais de um bilhão de pessoas vivem hoje com algum tipo de deficiência, isso equivale a aproximadamente 15% da população. As Diretrizes de Acessibilidade para o Conteúdo da Web (WCAG, do inglês Web Content Accessibility Guidelines) é um documento publicado pela Iniciativa de Acessibilidade Web da World Wide Web Consortium (W3C) e carrega uma série em recomendações de acessibilidades para criação e publicação conteúdo Web.

Muitos pesquisadores adaptam o WCAG para outros contextos que envolvem acessibilidade a conteúdos digitais. Como as diretrizes do WCAG abrangem uma série de deficiências incluindo visual, auditiva, cognitiva, motora e de fala, ela também pode ser aplicada aos indivíduos idosos que naturalmente sofrem perda de habilidades provenientes da idade avançada (W3C, 2018). Portanto, assim como o WCAG, este trabalho busca contemplar diferentes tipos de deficiências, tanto de forma isoladas e também combinadas, para assim aproximar os resultados deste trabalho às reais necessidades dos idosos.

No contexto da engenharia de software muito esforço já vem sendo aplicado para criação de interfaces que ofereçam uma comunicação eficiente com o usuário. As três regras de ouro para modelagens de interface de usuário são o alicerce que guia este importante aspecto da criação de softwares (PRESSMAN, 2015, p. 318-321):

1. **Manter o usuário no controle**, fazendo-o sentir livre para tomar as decisões desejadas por meios de interações que não force ou induza o usuário realizar ações indesejadas.

2. **Reduzir a carga de memorização do usuário** através de interfaces que façam uso de metáforas reais que poupem o usuário de memorizar padrões exclusivos em cada software, além disso, cada tela deve manter apenas o conteúdo significativo para as necessidades do usuário naquele momento.

3. **Fazer uma interface consistente** que permita ao usuário reconhecer “sua localização” no software e navegar através de padrões já formalizados e que mantenham um sentido constante durante todas as telas.

Pressman (2015, p. 323) cita uma colocação de Shneiderman (2004) dizendo que para construir uma interface de usuário eficaz "toda modelagem deve começar com a compreensão dos usuários pretendidos, incluindo os perfis de idade, sexo, habilidades físicas, educação, origem cultural ou étnica, motivação, objetivos e personalidade". Além disso, Pressman (2015, p. 323) diz que os usuários podem ser classificados como iniciantes, usuários informados, intermitentes ou usuários frequentes informados.

Assim como na engenharia de software convencional, a literatura da engenharia de software no desenvolvimento de jogos não prevê uma metodologia de desenvolvimento bem definida para atender às necessidades dos usuários de acessibilidades. Esta carência é abordada por guias, recomendações e diretrizes propostas por projetos paralelos e iniciativas independentes, normalmente advindas de pesquisas acadêmicas, institutos de padronização e outros trabalhos colaborativos. A seção a seguir traz uma série de iniciativas que pretendem colaborar com o cenário de desenvolvimento de jogos para criação de produtos mais inclusivos e adequados ao público idoso.

## **5.5 Métodos de Testes para jogos**

A etapa de teste é um componente crítico do processo de desenvolvimento de jogos (Test Bytes, 2018) e é uma das atividades mais importantes para garantir qualidade e desempenho ideais para o produto final (Test Bytes, 2017b).

A partir dos testes é possível identificar bugs, verificar irregularidades da estrutura ficcional do jogo, inconsistências na arte conceitual, desnivelamento da curva de dificuldade do jogo, problemas de desempenho do programa, dificuldade de usabilidade, além de outros critérios da jogabilidade como: fluxo do jogo, diversão e motivação. Portanto, é imperativo aprender sobre uma das partes vitais do processo de desenvolvimento de jogos - Teste de Jogos (Test Bytes, 2018).

Assim como para testes de softwares convencionais, existe um série de métodos de testes aplicáveis ao desenvolvimento de jogos, incluindo alguns citados no processo proposto neste trabalho:

- Teste Fumaça: também conhecido como "Teste de verificação de compilação", é um tipo de teste de software que compreende um conjunto não exaustivo de testes que visa garantir que as funções mais importantes funcionem. O resultado desse teste é usado para decidir se uma construção é estável o suficiente para prosseguir com testes adicionais (Software Testing Fundamentals, 2020d)
- Teste de Jogabilidade: é usado para avaliar recursos não funcionais, como fatores de diversão, níveis de dificuldade, balanceamentos, etc e ajuda a verificar se o jogo funciona de maneira bem estruturada (Test Bytes, 2017a).
- Teste de Usabilidade: pode ser explicado de forma sucinta como “um tipo de teste feito da perspectiva do usuário final para determinar se o sistema é facilmente utilizável” (Software Testing Fundamentals, 2020e). Em um teste de usabilidade, o jogo é testado com jogadores que representam o público-alvo do jogo. O objetivo é identificar os desafios do jogo que não foram planejados pelos desenvolvedores do jogo. É indispensável a realização de testes de usabilidade no decorrer do desenvolvimento de um jogo acessível, pois ele fornece informações diretas e objetivas sobre como jogadores reais jogam o jogo e quais são os problemas de usabilidade que os jogadores enfrentam enquanto jogam (LAITINEN, 2020).
- Teste de Regressão: é um tipo de teste de software que pretende garantir que mudanças realizadas no software não tenham causado efeitos indesejados em outras partes do software (Software Testing Fundamentals, 2020c). Portanto, as partes inalteradas do software são verificadas novamente para analisar se o funcionamento das funções anteriores do aplicativo funcionam bem e se as novas alterações não introduziram novos erros ou vulnerabilidades (Test Bytes, 2017a). O site “Software Testing Help” (2020) oferece mais informações sobre como proceder com os testes de regressão.
- Teste de Performance: é um tipo de teste de software para determinar o desempenho de um sistema em termos de capacidade de resposta e estabilidade sob uma determinada carga (Software Testing Fundamentals, 2020b). É um teste importante a ser feito, pois jogos lentos e com performance ruim podem ser a certeza de um

fracasso (HELPPPI, 2020). Para melhor compreensão da performance do jogo em relação à realidade da sociedade, o teste deve ser feito tanto em dispositivos de ponta como em dispositivos de entrada.

- **Teste de compatibilidade:** devido a grande diversidade de dispositivos disponíveis no mercado, é um dos testes essenciais para aplicativos de dispositivos móveis, pois verifica se um jogo pode ser executado em dispositivos específicos (Test Bytes, 2017a). Outro ponto importante é que esse tipo de teste pode ajudar a identificar incompatibilidades de qualquer parte do jogo com módulos externos ou aparelhos que o usuário final conecta ao dispositivo de jogo, como fones de ouvido e controles joysticks (HELPPPI, 2020). Mais informações sobre testes de compatibilidade em (Guru99, 2020).
- **Teste de Conformidade:** é um tipo de teste para determinar a conformidade de um sistema com padrões internos ou externos. Os padrões internos podem ser regras estabelecidas internamente pela equipe, projeto ou empresa, por exemplo: os jogos devem seguir um padrão de acessibilidade pré-definido pela empresa. Enquanto que os padrões externos podem ser normas estabelecidas por entidades como organizações de saúde, plataforma de distribuição do aplicativo ou lei governamentais (Software Testing Fundamentals, 2020a).
- **Teste Ad Hoc:** é um método de teste não planejado, geralmente usado para quebrar o sistema. Os testadores testam o aplicativo aleatoriamente sem casos de teste ou documentos (Test Bytes, 2017a). Portanto, essa metodologia de teste intuitiva e criativa exige que o testador seja extremamente hábil e possua um profundo conhecimento do sistema. O resultado desse teste pode identificar bugs que passam despercebidos por métodos tradicionais (Software Testing Help, 2020a).
- **Teste Funcional:** de acordo com Helppi (2020), o teste funcional é talvez o método de teste mais comum para jogos em dispositivos móveis e eficiente para identificar bugs e erros que podem afetar a experiência do usuário. Em oposto ao método ad hoc, o método funcional é um teste minucioso que determina se o aplicativo está funcionando de acordo com as especificações do projeto e leva mais tempo para ser executado, pois os testadores procuram por problemas de jogabilidade, problemas gráficos, problemas audiovisuais e outros (Test Bytes, 2017a).



## **6. Trabalhos Relacionados**

Os trabalhos relacionados são divididos em dois grupos: 6.1 Recomendações sobre Acessibilidade para Idosos; e 6.2 Modelos de Desenvolvimentos de Jogos. O primeiro contempla trabalhos voltados para a análise e aplicação de métricas de acessibilidades digitais, enquanto que o segundo grupo de trabalhos relacionados lista algumas estratégias adotadas pelos autores para estruturação de um projeto de jogo.

### **6.1 Recomendações de Acessibilidade para Idosos**

Os trabalhos relacionados a seguir contemplam testes de diretrizes, propostas de adaptações de diretrizes da WCAG ao desenvolvimento de jogos, mecanismo de avaliação de interface para o público idoso, técnicas de projeto de jogos para idosos e processos de avaliação de fatores motivacionais nos jogos para idosos.

Cheiran (2013) faz um compilado de diversas diretrizes de acessibilidade de outros autores, as analisa, avalia e propõe as adaptações para serem aplicadas ao desenvolvimento de jogos. Suas diretrizes propostas foram estruturadas no padrão técnico WCAG e passaram por um processo de avaliação manual para encontrar problemas de compreensão e mensurar as dificuldades de aplicação.

Sua principal contribuição foi a formação de uma ampla coleção de diretrizes que abrangem uma vasta gama de recomendações de acessibilidade e uma proposta de organização baseada numa estrutura já reconhecida pela comunidade.

Machado e outros (2015) propuseram um conjunto de heurísticas para avaliar a jogabilidade de jogos digitais quando aplicados ao público idoso. Para validar as heurísticas foi realizada uma avaliação com especialista em jogabilidade e “game design” e com idosos que interagiram com jogos disponíveis no mercado para dispositivos móveis. Os resultados de seus testes confirmaram teses como a necessidade de adequação da dificuldade dos jogos às restrições decorrentes da idade avançada e a influência do conteúdo audiovisual como aspecto importante para motivação dos jogadores idosos.

Duque e Ishitani (2016) analisaram técnicas de aplicação de dificuldade e desafios em jogos digitais e colaboram com propostas de como balancear os desafios aplicados aos jogadores idosos. Seu trabalho envolveu aplicação de testes com idosos que experimentaram jogos representantes dos três diferentes conceitos de adaptação de dificuldade avaliados pelos pesquisadores: dificuldade adaptativa, dificuldade aumentada e dificuldade estática.

Duque e Ishitani (2016) demonstram seu modelo teórico do fluxo de estado mental dos idosos quando diante de um desafio de um jogo. A forma como os desafios são propostos é fator substancial para levar o jogador a um estado de motivação e que o faça desejar continuar jogando, mas quando o balanceamento de dificuldade é mal aplicado, pode causar frustração, tédio e desmotivação.

Duque et al. (2018) propõe a aplicação de técnicas de “Design Centrado no Usuário” e “Design Participativo” para envolver os idosos na participação de jogos digitais. Sua pesquisa visou identificar as contribuições que o usuário idoso pode fornecer no processo de desenvolvimento de jogos educacionais. Os idosos participantes da pesquisa jogaram quatro jogos móveis diferentes e por fim responderam a questionários para identificar questões como: familiaridade com o conteúdo do jogo, grau de experiência com a tecnologia e com jogos, características que gostaram ou não gostaram e nível de adequação dos jogos à usabilidade do idoso.

Os resultados alcançados por Duque et al. (2018) reforçam a importância da participação ativa do idoso no processo de desenvolvimento de um jogo, para melhor compreensão e adequação dos requisitos levantados e para identificar novos requisitos e funcionalidades para o jogo.

O último trabalho relacionado a ser apresentado é o *Game Accessibility Guidelines* (2020a), cujo trabalho colaborativo de estúdios de jogos, acadêmicos e pesquisadores foi compilado em um documento online que oferece dezenas de diretrizes para acessibilidade em jogos digitais.

As diretrizes estão agrupadas nas categorias básica, intermediária e avançada, e levam em consideração o público alcançado, o impacto inferido a esse público e o custo de implementação para cada uma dessas diretrizes. O documento ainda oferece um guia de como trabalhar com tais diretrizes e como aplicá-las a um projeto de jogo.

## **6.2 Modelos e Processos de Desenvolvimento de Jogos**

A maioria dos processos de desenvolvimento de jogos possui estágios (ou fases), independentemente de usarem de metodologias ágeis ou não. Esses estágios mudam a maneira como as equipes desenvolvem o jogo (KEITH, 2010). A revisão sistemática realizada por Aleem (2016) lista diversas estratégias que estúdios de jogos e pesquisadores aplicaram para organizar o processo de desenvolvimento de jogos.

Kruchten (2000) utiliza um modelo denominado Processo de Desenvolvimento Unificado (do inglês Unified Development Process) que deriva da engenharia de software tradicional e é focado na análise de requisitos e em como convertê-los em componentes funcionais para o software. Seu processo era definido em 5 fases: requisitos, análise, modelagem, implementação e testes.

O Blitz Game Studio (2020), assim como os próximos da lista, faz uso de um modelo próprio derivado das metodologias ágeis. O estúdio Blitz trabalha com um modelo em 6 fases: pitch (modelagem inicial e conceito do jogo), pré-produção (GDD - do inglês Game Design Document), produção principal (implementação do conceito do jogo), alpha (testes internos), beta (testadores terceirizados) e master (lançamento do jogo).

Hendrick (2014) propõe um ciclo de vida para GDSE dividido em 5 fases, consistindo de Prototipação (modelagem do protótipo inicial), Pré-produção (GDD), Produção (criação de ativos, código fonte e integrações), Beta (feedback de usuários) e Live (pronto para jogar).

McGrath (2014) dividiu o ciclo de vida do GDSE em 6 fases: Modelagem (modelagem inicial e GDD), Construção/Reconstrução (desenvolvimento da motor do jogo), Avaliação (se não passar, então reconstruir), Testes (testes iniciais), Revisão de lançamento (testes terceirizados), Pós-produção (atividades de post-mortem).

Ramadan (2013), que também foi analisado por Aleem (2013), elaborou uma proposta de processo de desenvolvimento para jogos digitais, ou como descrito em seu trabalho, uma proposta de ciclo de vida do desenvolvimento de jogos (GDLC, do inglês Game Development Life Cycle). Enquanto analisava outras propostas e modelos já em produção, Ramadan (2013) identifica que um modelo de desenvolvimento de jogos pode ser genericamente representado em três fases: 1 - design e prototipação, 2 - produção e 3 - testes. Ramadan (2013) verificou tanto alguns modelos que seguem uma abordagem linear entre fases, como outros que trabalham a interatividade entre as fases da criação do jogo. O processo proposto por Ramadan (2013) consiste de 6 fases (iniciação, pré-produção, produção, testes, beta e lançamento) que interagem entre si durante o ciclo de produção e quatro estágios de prototipação com objetivos bem definidos.

A aplicação da metodologia Scrum no desenvolvimento de jogos descrita por Keith (2010) sintetiza, para fins didáticos, o processo de criação de jogos em 4 fases: Conceito, Pré-produção, Produção e Pós-produção. O conceito é uma fase de *brainstorm* de ideias. As ideias são geradas, possivelmente prototipadas e regularmente descartadas. Tudo isso para

alcançar um conceito aproximado do que se busca produzir. A pré-produção é uma fase iterativa e incremental onde a equipe descobre iterativamente o que é divertido e como criar ativos que contribuem para isso, em seguida a equipe adapta gradualmente o planejamento do desenvolvimento com esse conhecimento adquirido. Na fase de produção a equipe foca em criar uma experiência com base nos principais mecanismos e processos identificados na fase de pré-produção. Essa fase se concentra em eficiência e melhorias incrementais. A equipe realiza menos iterações na mecânica principal pois estará criando uma grande quantidade de ativos baseados nessa mecânica já definida. Durante a fase de pós-produção a equipe se concentra em aprimorar toda a experiência de jogo de forma incremental que depois passará por testes de hardware e integração.

Conforme descrito por Keith (2010), os estágios de um processo não são isolados entre si, o processo evolui gradualmente. Metodologias ágeis valorizam muito o conhecimento adquirido durante o processo, portanto é possível que atividades de conceito e pré-produção sejam realizadas durante a fase de produção a fim de refiná-la. A estrutura do Scrum é mantida durante todo o processo, mas a equipe deve ajustar as práticas em cada estágio.

Na fase de **conceito**, os sprints são mais curtos e o backlog do produto, ainda muito pequeno, possui apenas os primeiros insights para o projeto. O principal objetivo da fase de conceito é criar conhecimento para a equipe e stakeholders, e não valor para os consumidores. As metas de entrega são tratamentos conceituais e talvez protótipos para demonstração às partes interessadas.

Durante a **pré-produção**, o Scrum é usado para descobrir a diversão do jogo e de forma incremental e iterativa, construir valor e conhecimentos sobre os custos do jogo. O desenvolvimento é regulado por sprints e a definição das metas de entrega são as principais metas dessa fase.

Como em outros modelos e processos de GDSE, na fase de **produção** definida por Keith (2010) a equipe produz ativos que foram definidos na fase de pré-produção e melhoram de forma incremental a sequência dos ativos. Embora os sprints e entregas ainda sejam utilizados, o ritmo da produção dos ativos se torna a métrica para medir a velocidade do projeto.

E por último, a fase de **pós-produção** é mantida para tarefas de ajustes, polimentos e correções de bugs. Apesar dessas atividades serem realizadas diariamente durante os sprints de produção, o backlog do produto direciona mais esforço para estas tarefas (polimentos e

correção de bugs) durante a pós-produção. A fase de pós-produção começa com a data da versão alpha do jogo e se estende pela versão beta até a data de lançamento.

## **7. Processo de Desenvolvimento de Jogos Digitais Acessíveis ao Público Idoso**

Esta seção apresenta a proposta do autor para um processo de desenvolvimento de jogos que contemple as necessidades de idosos e, paralelamente, portadores de deficiências, doravante chamado de JAI (Jogos Acessíveis ao Idoso) ou processo JAI.

O processo de criação de jogos do Blitz Game Studio (2020), a proposta de Hendrick (2009) para gerenciamento de um projeto de desenvolvimento de jogos, as orientações do livro sobre desenvolvimento ágil para jogos com Scrum escrito por Keith (2010), o guia para o ciclo de vida do desenvolvimento de jogos escrito por Ramadan (2013) e as diretrizes de acessibilidade descritas detalhadamente no documento online *Game Accessibility Guidelines* (2020a) são alguns dos trabalhos com maior influência na proposição desse processo para desenvolvimento de jogos acessíveis.

A seção 7.1 explica os critérios adotados pelo autor para guiar a formulação do processo JAI, a seção 7.2 oferece uma visão geral das fases do processo e as seções 7.3 a 7.7 descrevem criteriosamente cada uma destas fases.

### **7.1 Critérios para Construção do Processo JAI**

Os critérios a seguir foram selecionados para nortear a construção dos fluxos de trabalho deste modelo, assim como para aumentar a viabilidade de sua adoção.

**Design Participativo e Design Centrado no Usuário:** O foco do processo JAI é promover amplamente a usabilidade e os elementos motivacionais nos jogos que serão criados. Para isso é indispensável a participação ativa do “usuário alvo” no desenvolvimento do jogo, tanto para avaliação das interfaces como para medir a satisfação do jogador a cada mudança aplicada no jogo.

**Metodologia Ágil:** As metodologias ágeis oferecem maior capacidade de enfrentar mudanças inesperadas no decorrer de um projeto. Como o processo JAI visa conciliar os projetos a diferentes requisitos de acessibilidades, é natural que surjam diversas adaptações e mudanças a ser realizadas após cada interação de testes com os usuários.

**Acessibilidade em todo o processo:** A discussão sobre as diretrizes de acessibilidade devem ocorrer tanto nas fases iniciais do projeto como também ao longo dos ciclos de

produção e devem ser levadas em conta durante a construção do plano de testes e também após a publicação.

**Adequação a equipes pequenas:** O processo JAI deve ser capaz de trabalhar com equipes pequenas (de 2 a 6 pessoas). Mesmo as metodologias ágeis convencionais visam equipes bem estruturadas com ao menos 6 integrantes envolvidos. O cenário dos criadores de jogos no âmbito acadêmico ou independentes não costumam ter recursos e capital para iniciar com grandes ou médias equipes. De acordo com o 2º Censo da Indústria Brasileira de Jogos Digitais (SAKUDA, 2018) a média de profissionais por empresa de jogos é de 17,6 pessoas, entre sócios e funcionários. Das 50 empresas pesquisadas, apenas 4 (7,4%) possuía mais de 50 pessoas, enquanto que 40 empresas (74%) possuíam até 10 profissionais e quase metade (44,4%) possuía 5 ou menos pessoas envolvidas. Quanto aos recursos de financiamentos das empresas, a pesquisa mostrou que, das 76 empresas respondentes à pesquisa, 30 empresas (39,5%) utilizam como principal fonte de financiamento recursos próprios, da família, ou de outros indivíduos.

## 7.2 Visão Geral do Processo JAI

Assim como os demais processos de desenvolvimento, o processo proposto segmenta todo o esforço de realização do jogo em etapas e tarefas distintas para melhor gerir o fluxo de trabalho. A proposta consiste em cinco fases macro:

- **Iniciação:** A modelagem inicial do jogo é produzida contendo o conhecimento necessário para criação dos primeiros protótipos da fase de pré-produção.
- **Pré-produção:** É a fase focada na prototipação e amadurecimento da modelagem do projeto. O documento de projeto do jogo e demais conhecimentos obtidos na fase de pré-produção servirão de guia para a fase de produção.
- **Produção:** Reconhecido pelos ciclos de produção, que envolvem toda a criação de código, ativos audiovisuais, testes contínuos e integrações.
- **Beta:** O jogo é exposto abertamente para testes enquanto são realizados refinamentos e correção de bugs, além da finalização de módulos adicionais.
- **Publicação:** Marca o momento de lançamento do jogo. Também é criado o plano de manutenção do jogo e ocorrem atividades referentes ao post-mortem.

A figura 7.1 ilustra o sequenciamento entre essas etapas e os artefatos que devem ser entregues em cada uma.

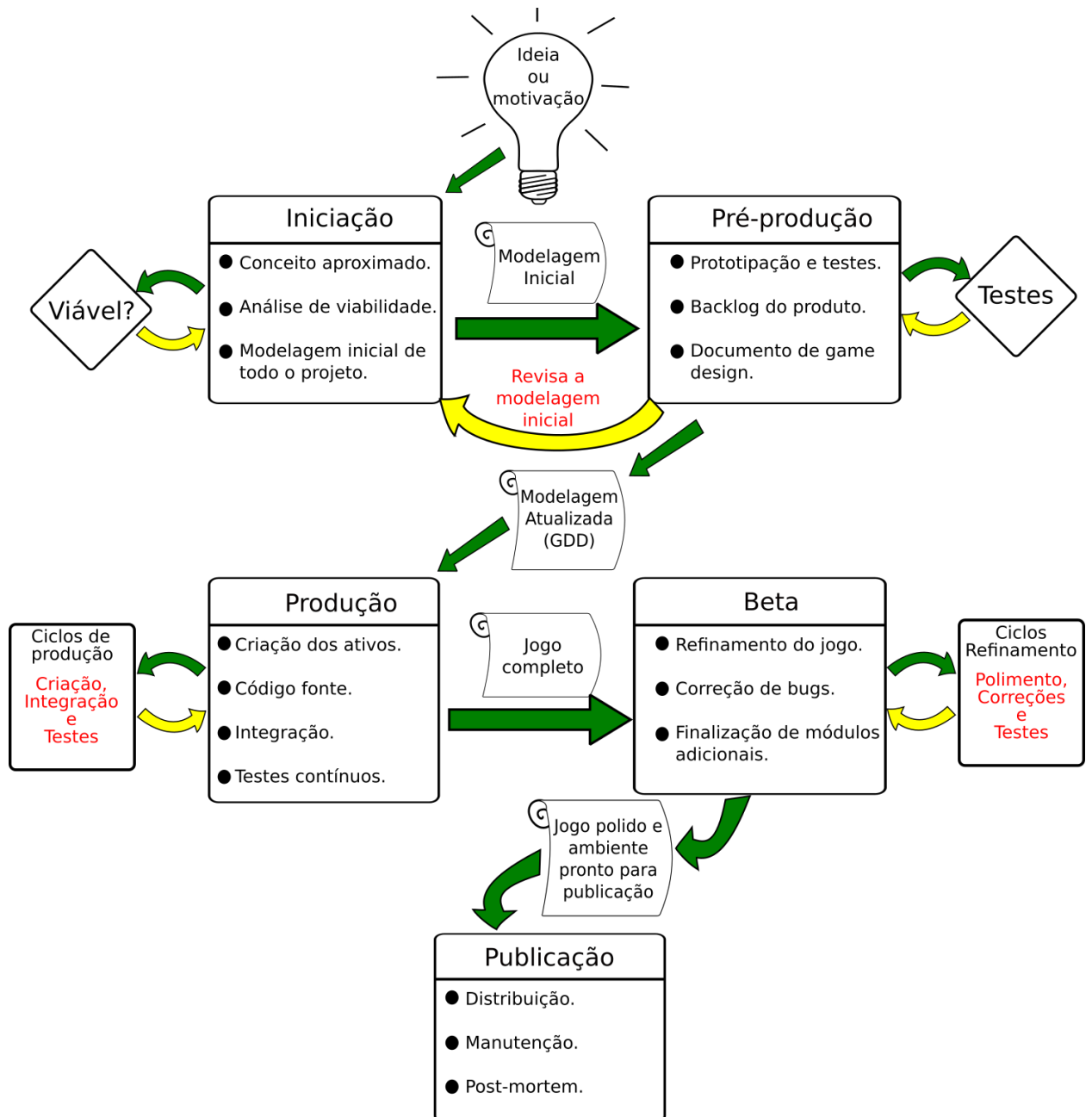


Figura 7.1 - Estrutura do processo JAI e os artefatos produzidos em cada fase.

Cada fase é constituída por um conjunto de atividades que orientam a sua aplicação no desenvolvimento de jogos acessíveis. Estas atividades são representadas através de *boards* criados para materializar tais orientações e facilitar a aplicação destas em cada fase do processo JAI, entregando ao desenvolvedor de jogos uma forma mais prática de aplicá-lo.

### 7.3. Iniciação

**O que entrega:** ao fim da fase de iniciação a equipe terá criado o conceito aproximado do jogo, a modelagem inicial do projeto, um cronograma de atividades, um esboço das estratégias de monetização, de publicação, de marketing, do sistema de atualização e de suporte ao cliente, tudo compilado na versão preliminar do Documento de Projeto do Jogo (GDD, na sigla em inglês).

**No que consiste:** a iniciação é o momento de elaborar um rascunho das ideias para o jogo, conhecer o público-alvo e o mercado, fazer análise de viabilidade, definir as ferramentas de trabalho necessárias e esboçar a modelagem com o cronograma das atividades.

**Como atende aos idosos:** durante esta fase, a equipe terá o primeiro contato com público e já deve enumerar as diretrizes de acessibilidade que se encaixam ao projeto e que podem ser atendidas.

O desenvolvimento da fase de iniciação segue uma sequência quase linear, tendo o resultado da **análise de viabilidade** como condicional para interrupção, redirecionamento ou prosseguimento com as atividades, conforme ilustrado na Figura 7.2.

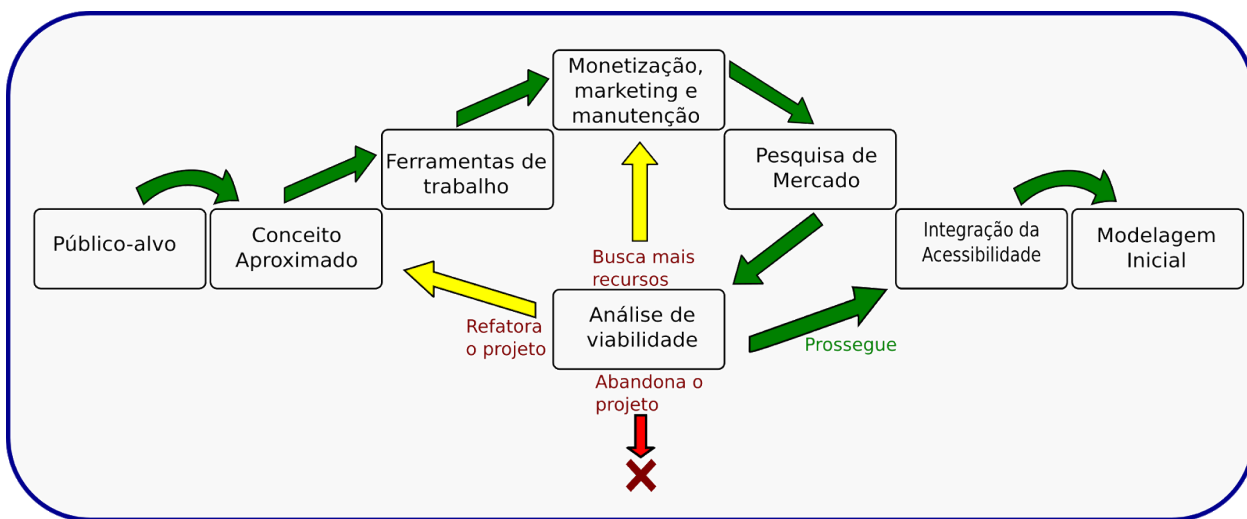


Figura 7.2 - Sequência de atividades sugerida para a fase de iniciação. Fonte: do autor.

A figura 7.3 apresenta o Board das Atividades da Fase de Iniciação, que serão descritas nas subseções que se seguem.



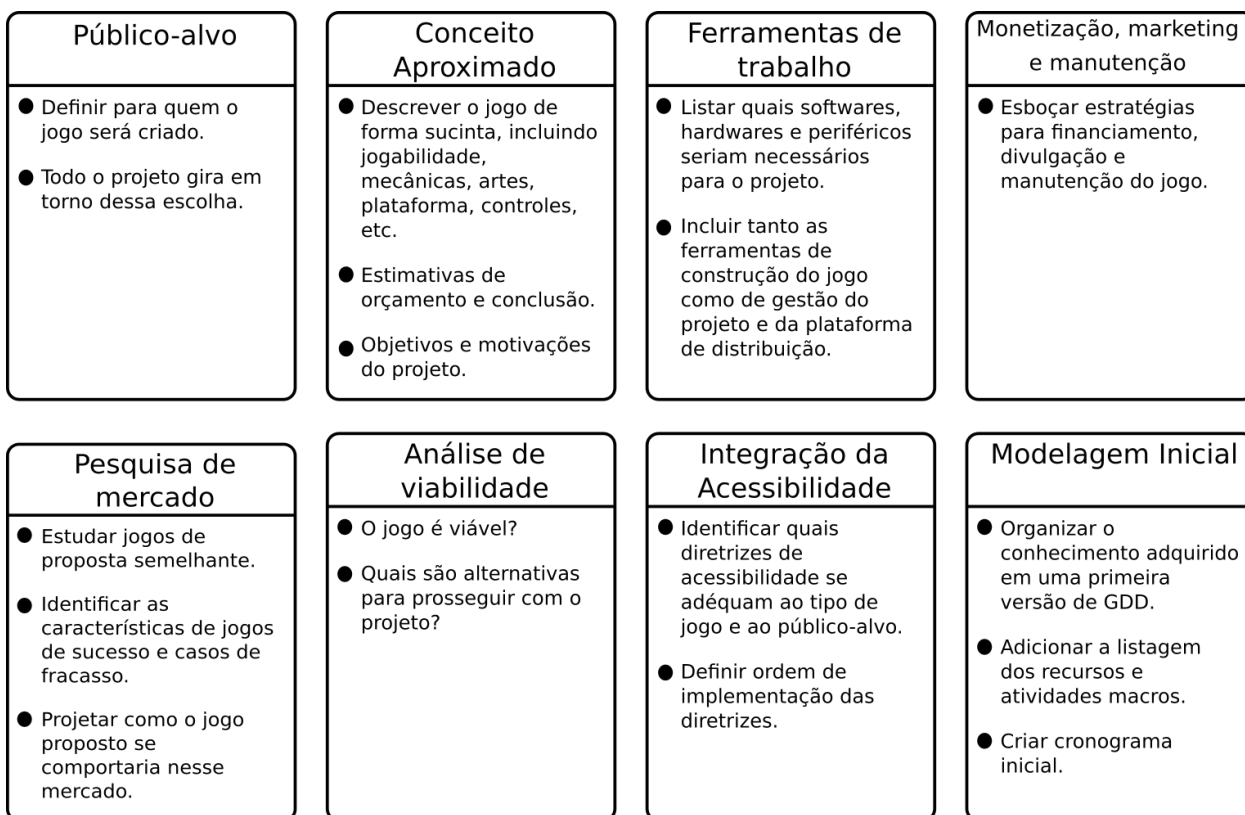


Figura 7.3 - Board das Atividades da fase de Iniciação. Fonte: do autor.

### 7.3.1 Público-alvo

Ao idealizar um jogo, é necessário considerar quem seriam os potenciais jogadores - para quem o jogo está sendo criado e que tipo de pessoa apreciaria jogá-lo. Um jogo é produzido com uma variedade de mecânicas, artes e efeitos audiovisuais que definem sua jogabilidade e ajudam a identificar em qual público despertaria maior interesse.

Apesar de Ferro (2018) concluir que personalidade de uma pessoa não é característica suficiente para prever sua preferência por tipos de mecânicas e elementos de jogos, Tondello (2019) identificou relações entre o sexo, a idade e característica de um jogador e sua possível predileção por determinados elementos e estilos de jogos. A pesquisa de Vahlo (2017) também ajuda criadores de jogos ao demonstrar a relação de preferência e rejeição de jogadores por determinadas dinâmicas.

Portanto, antes mesmo de descrever as características do jogo, a escolha do público-alvo é primordial. Todo produto é criado para alcançar alguém, para ser usado por algo ou alguém em determinada condição e situação. Com o público-alvo em mente, todo o decorrer do projeto orbita em torno de como criar algo de valor para estas pessoas.

### 7.3.2 Conceito aproximado

O conceito aproximado do jogo é um conjunto de informações básicas que descrevem o jogo a ser criado, mas que apesar de sucinto, é importante oferecer detalhes o suficiente para que qualquer pessoa possa entender a ideia geral do jogo através dessa leitura. O conceito aproximado deve conter ao menos as seguintes informações:

- Objetivos específicos e secundários por trás do projeto. Os objetivos devem ser claros e motivadores, também servem como guias e ajudam a manter o escopo e foco do projeto.
- O gênero e subgênero do jogo. Ex: ação de plataforma, casual de tabuleiro, casual de puzzle (jogos que desafiam o raciocínio), RPG de ação, etc.
- Mecânicas básicas de jogabilidade. Ex: controlar a mira de uma arma de bolinhas para acertar o alvo, usar o elementos do cenários para redirecionar a trajetória das bolinhas, derrotar os inimigos que aproximam da arma, etc.
- Arte conceitual. Ex: Desenhos em 3D *low poly* (modelagem 3D com polígonos de baixa complexidade), cores sólidas e sem degradês, contornos para maior contraste dos elementos interativos em relação ao cenário.
- Roteiro (quando aplicável). Mesmo pequenos jogos casuais podem oferecer uma simples narração que contextualizam seu próprio universo imaginário, como ocorre no jogo “Disney Emoji Blitz”, da Jam City, Inc (2020).
- As plataformas para publicação. Ex: PC, console, mobile e arcade.
- Controles. Ex: teclado e mouse, joystick, joystick virtual, tela de toque, etc.
- Estimativa de orçamento e conclusão. Quanto dinheiro e esforço a equipe pensa em investir para criação do jogo e qual é o tempo previsto para realização.
- Recursos específicos essenciais para o projeto. Alguns jogos requerem o uso de kits específicos para testes ou adaptação a plataforma de distribuição. Ex: kits de desenvolvedor, óculos 3D, placas de arcades, etc.

### 7.3.3 Ferramentas de trabalho

O conceito aproximado do jogo oferece as informações necessárias para que a equipe do projeto escolha quais ferramentas são necessárias para criação do jogo. Obter esta listagem dos softwares, hardwares e periféricos ajuda a mensurar o custo total do projeto, pois algumas

dessas ferramentas podem adicionar despesas de compra ou de licença de uso. Estas ferramentas incluem, entre outras:

- Game Engine;
- Software de Modelagem 3D;
- Softwares e equipamentos de desenho vetorial e pixelado;
- Softwares e equipamentos para edição de som;
- Ferramenta de controle de versão;
- Plataforma de gestão do projeto;
- Periféricos (óculos de realidade virtual, dispositivos de testes, etc)

#### **7.3.4 Estratégias de monetização, marketing, sistema de atualização e suporte ao cliente**

Além do núcleo composto pelas atividades intrínsecas à criação do jogo, existem atividades que correm em paralelo e que também suportam o sucesso do projeto. Esta terceira tarefa trata-se de estudar e adaptar estratégias para financiar o projeto, conhecer as mídias para divulgá-lo e torná-lo visível ao público-alvo, e por último, esboçar como será feita a manutenção e o suporte ao cliente após publicação do jogo. Portanto é necessário destacar e considerar ao longo do projeto os seguintes tópicos:

- Financiamento/Monetização - é necessário que projeto possua fundos para suprir os custos de produção, marketing, publicação e manutenção. Seja por meio de editais de financiamento ou através de técnicas de monetização de aplicativos, é importante fazer um levantamento das formas mais comuns e efetivas de monetizar jogos do gênero em questão ou arrecadar financiamentos para o projeto.
- Marketing - é interessante pesquisar as formas de propaganda que surtem melhor retorno para o tipo de jogo desejado e os custos para aplicá-las. Também deve-se considerar a possibilidade de terceirizar essa atividade.
- Manutenção - após a publicação, é comum que os jogos recebam algumas atualizações para correção de bugs e expansão das funcionalidades. Para obter *feedback* constante e oferecer um retorno claro e agradável aos jogadores, é importante ter um canal de comunicação estável, seja em uma página de rede social, um site próprio ou dentro da própria plataforma de distribuição.

### **7.3.5 Mercado**

O jogo a ser criado deve ser visto como um produto a ser vendido e consumido, portanto é importante conhecer o comportamento do mercado onde este jogo será anunciado, quem são os consumidores do produto, quem são os concorrentes, os casos de sucesso, as características de fracasso, os fatores de fidelização e o comportamento do jogador no pós-compra. As informações adquiridas após a pesquisa devem ser confrontadas com o conceito aproximado e demais subtópicos anteriores. Essa comparação ajudará a responder questões como:

- Existe público interessado no tipo de jogo pensado?
- Se há interesse no jogo, se trata de um nicho momentâneo ou o jogo tende a ser jogado por um longo período?
- Quais seriam os diferenciais deste novo jogo em relação aos já encontrados no mercado?
- Como se destacar diante da concorrência?
- É possível criar esse jogo com os recursos disponíveis no momento?
- Qual é o orçamento e prazo médio para produção desse tipo de jogo?
- Há capital de reserva para suportar até a publicação do jogo?
- A estratégia de monetização cobriria os custos de produção?
- O jogo seria rentável?

Para mais informações sobre análise de mercado e outras relações de negócios, o portal do SEBRAE (2020) oferece um guia prático e maiores direcionamentos para se aprofundar no assunto.

### **7.3.6 Análise de viabilidade**

Ao confrontar os resultados da análise de mercado com os objetivos do projeto descritos no conceito aproximado, a equipe é capaz de responder ao único questionamento desta tarefa: “O projeto é viável?”. Se sim, é possível seguir em frente executando as próximas tarefas. Caso contrário, três alternativas podem ser aplicadas:

1. O projeto pode ser interrompido neste ponto e evitando maiores perdas e prejuízos futuros.

2. O projeto pode ser refatorado a partir do conceito aproximado a fim de criar algo dentro de um escopo viável.
3. A equipe deve buscar os recursos humanos, tecnológicos e financeiros necessários para suportar a ideia proposta inicialmente.

### **7.3.7 Integração da acessibilidade**

O processo proposto pretende destacar-se por garantir que as limitações cognitivas e motoras (comum em idosos) dos usuários sejam consideradas durante todo o processo de desenvolvimento do jogo, pois acredita que esse diferencial também possa contribuir para inclusão digital, bem estar social e até mesmo angariar vantagens competitivas no mercado de jogos eletrônicos.

Após passar pela avaliação de viabilidade, o conceito aproximado e a escolha do público ajudarão a compreender as diretrizes de acessibilidade que o jogo deve atender. Cientes de que é virtualmente impossível aplicar todas as diretrizes encontradas na literatura, o *Game Accessibility Guidelines* (2020b) possui um esquema de organização das diretrizes baseadas por alcance (número de pessoas que se beneficiam), impacto (o benefício sentido pelo público-alvo) e custo de implementação.

Seguindo essa estratégia, o objetivo desta tarefa é elencar o máximo de diretrizes que se adaptam ao gênero do jogo proposto, à plataforma e ao público-alvo, e mantê-las organizadas em seis categorias: Motor, Cognitivo, Visual, Auditivo, Fala e Geral. E por fim, estabelecer uma ordem de prioridade de implementação das diretrizes:

- O que deve ser implementado já nos primeiros protótipos.
- O que deve ser implementado na fase de produção.
- O que pode ficar para a fase de manutenção do jogo.

Para ajudar nesta tarefa, o Apêndice B é um material de consulta que oferece um compilado de diretrizes de acessibilidade seguindo o mesmo padrão categorias do *Game Accessibility Guidelines* (2020b) e que foi organizado para trabalhar em conjunto com o processo JAI.

### 7.3.8 Modelagem inicial do projeto e cronograma de atividades

As informações do projeto devem estar conectadas, organizadas e de fácil acesso a toda equipe de forma que ajude no gerenciamento do projeto e seja mais prático para consulta e atualização.

Todos os estudos, pesquisas e resultados desenvolvidos durante a fase de iniciação devem ser compilados na forma de um documento de projeto chamado GDD. Esta versão inicial do GDD evoluirá e acompanhará todo o projeto até mesmo após a publicação do jogo. O Apêndice A oferece um modelo de GDD simplificado que pode ser usado como guia para documentação do jogo desde esta fase inicial do projeto.

Já nessa nova documentação deve ser adicionado um cronograma inicial contendo a listagem de todas as atividades macro (pouco detalhamento), recursos, módulos, ativos, integrações que devem ser realizadas para a conclusão do projeto e o tempo estimado para realização de cada tarefa.

O Apêndice A, na seção “Cronograma de Atividades e Marcos de Desenvolvimento”, mostra um modelo de como construir um cronograma com as atividades macros do projeto. Este modelo pode ser expandido para gerir as atividades mais detalhadamente após a criação do “backlog do produto” durante a fase de pré-produção.

## 7.4 Pré-produção

**O que entrega:** a pré-produção acaba quando toda a modelagem, cronograma do projeto e o backlog do produto estiverem revisados e documentados no GDD completo.

**No que consiste:** a fase de pré-produção usa os artefatos preliminares elaborados durante a fase de iniciação para criar os protótipos interno e estrutural, sendo o primeiro para alinhamento interno com a equipe, enquanto o segundo é mais aprimorado e já voltado para testes com o público-alvo. Os novos conhecimentos adquiridos são usados para revisar e completar o GDD.

**Como atende aos idosos:** durante a fase de pré-produção ocorre o primeiro contato do público idoso com o jogo (na versão de protótipo estrutural). De acordo com as orientações de acessibilidade, os testes devem incluir participantes com deficiências e, portanto algumas diretrizes de acessibilidade já são implementadas para serem avaliadas durante os testes.

Durante a fase de pré-produção ocorre o amadurecimento do projeto. Cada atividade dessa fase produz conhecimentos para aperfeiçoar o GDD e prepara o projeto para iniciar os ciclos de produção.

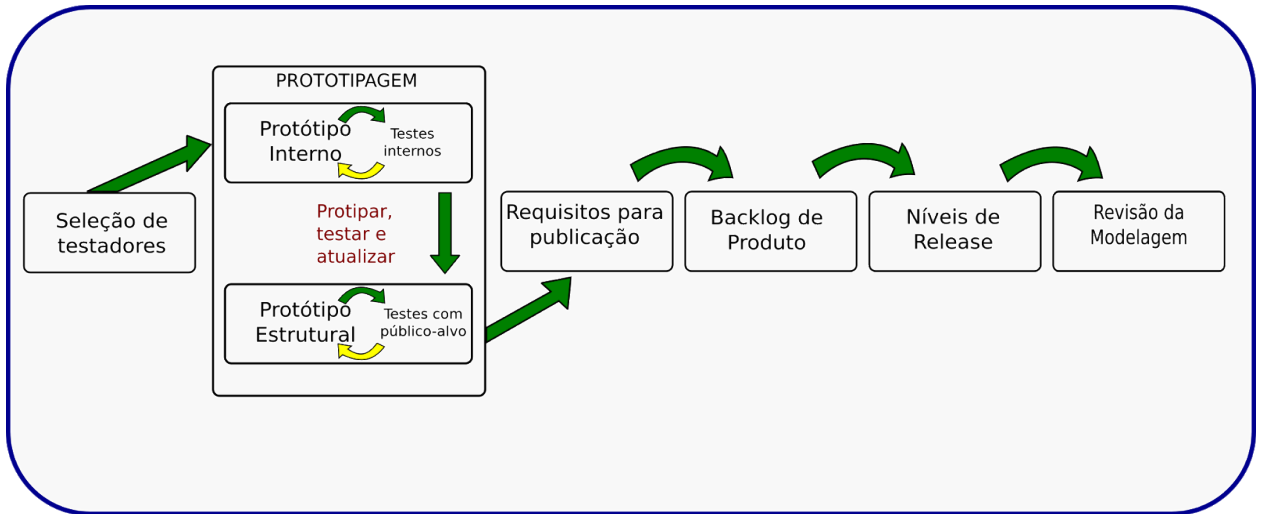


Figura 7.4 - Sequência de atividades sugerida para a fase de Pré-produção. Fonte: do autor.

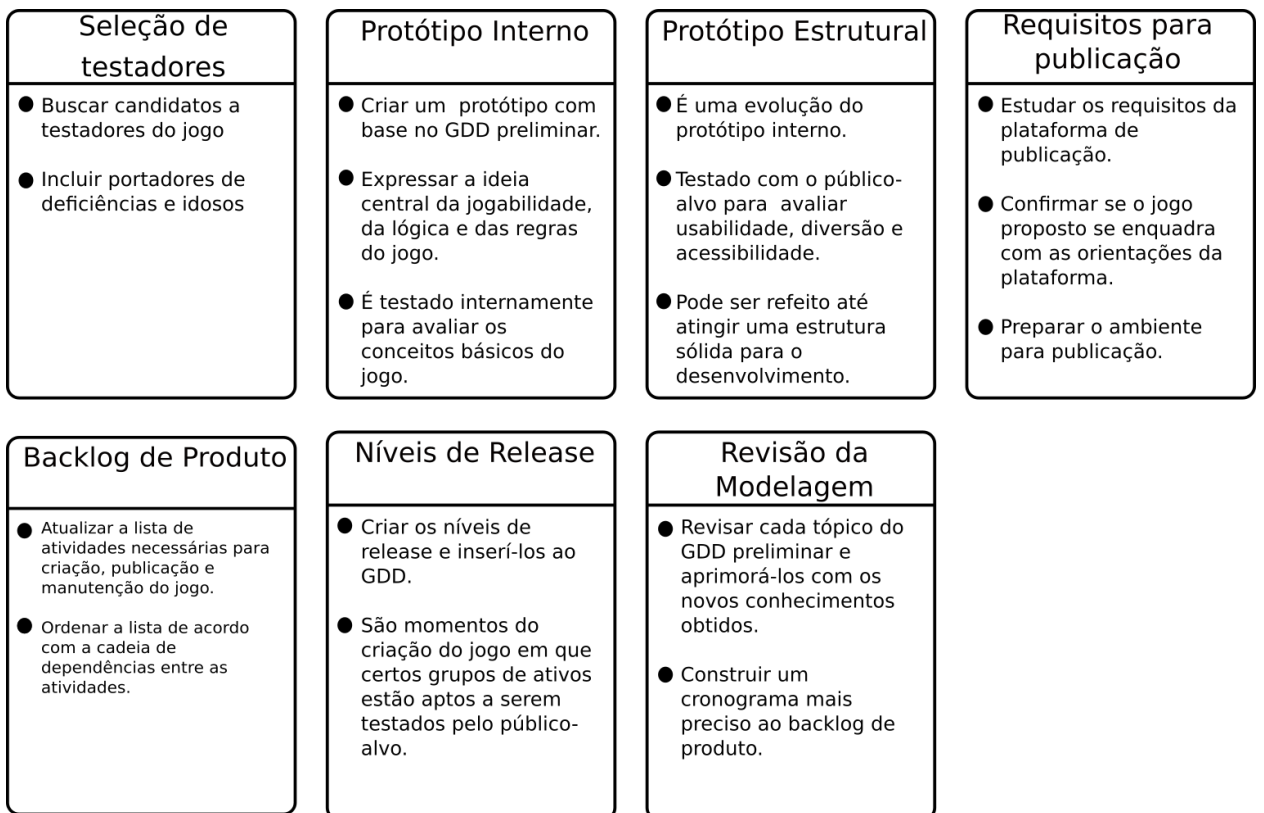


Figura 7.5 - Board das Atividades da fase de Pré-produção. Fonte: do autor.

#### **7.4.1 Diretrizes de acessibilidade para fase de pré-produção**

São diretrizes que podem ser implementadas já para os primeiros protótipos:

- Adaptações mais genéricas, mas que contemplam um grande público. A exemplo: aplicação de alto contraste entre os elementos da interface de usuário; uso de uma fonte padrão simples e legível; e garantir que os controles sejam o mais simples possível.
- Diretrizes que afetam diretamente os candidatos dos primeiros testes. A exemplo: se é de conhecimento da equipe que algum candidato utiliza aparelho auditivo, é recomendado implementar o controle de volumes separadamente para sons de efeito, música de fundo e falas, pois alguns sons mal compreendidos pelo jogador podem acabar sendo captados como ruídos incômodos.

#### **7.4.2 Seleção dos participantes dos testes**

O *Game Accessibility Guidelines* (2020a), no campo de diretrizes gerais, recomenda a inclusão de pessoas portadoras de deficiências entre os participantes dos testes do jogo. Em complemento, esta proposta aconselha participação de adultos mais velhos e idosos ao grupo de testadores para se obter um panorama mais realista do grau de acessibilidade e ludicidade do jogo.

Os primeiros protótipos e versões do jogo não possuem uma variedade de recursos de acessibilidade para atender todos as categorias de deficiência, portanto é importante conhecer os limites de acessibilidade do jogo e evitar frustrações desnecessárias. Dentre os candidatos a testadores do jogo, apenas os contemplados pelas adaptações de acessibilidade devem ser selecionados.

#### **7.4.3 Protótipo Interno**

O protótipo interno é criado com base no GDD preliminar produzido durante a fase de iniciação e deve exemplificar a ideia central da jogabilidade, da lógica e das regras do jogo. Essa atividade ajudará a equipe a alinhar sua compreensão sobre o jogo a ser criado e a praticar com ferramentas de trabalho com as quais podem não estar familiarizados. Para essa fase do projeto, este processo sugere a realização de um teste fumaça seguido de um teste de jogabilidade.



- Teste Fumaça: é um teste rápido que informa se o programa está compilando e se suas funcionalidades gerais estão funcionando.
- Teste de Jogabilidade: para este protótipo interno, o teste de jogabilidade pode ser aplicado com os próprios membros da equipe ou convidados. A ideia é obter um panorama geral de critérios como diversão e progressão de dificuldade.

Após a realização dos testes , a equipe deve responder às perguntas a seguir:

- Os desafios do jogo são divertidos e passíveis de balanceamento?
- A jogabilidade proposta parece simples e intuitiva (visando público idoso)?
- A forma de entrada dos comandos é clara e eficiente?
- O dispositivo escolhido (PC, celular, tablet, embarcado, etc) é confortável para o jogo e oferece os recursos de hardware necessários para jogá-lo?

Se a resposta for “**não**” para alguma destas perguntas, o protótipo deve ser refeito com os ajustes necessários para satisfazer as questões acima. É comum que o **conceito aproximado** do jogo seja alterado durante a fase prototipação. O processo de prototipar, testar e ajustar deve se repetir até que a equipe sinta consistência no jogo proposto.

Quando o protótipo interno estiver concluído e aprovado pela equipe, o conceito aproximado (da fase de iniciação) deve ser atualizado e os tópicos em sequência devem ser verificados para avaliar se as mudanças implicam em impactos nas demais decisões do projeto.

#### **7.4.4 Protótipo Estrutural**

O protótipo estrutural é uma evolução do protótipo interno e tem por objetivo ser testado com o público-alvo para avaliar critérios de usabilidade, diversão e acessibilidade. Para tal, é sugerido a realização dos seguintes testes:

- Teste Fumaça: é importante verificar se o protótipo estrutural é estável o suficiente antes de ser testado pelo público.
- Teste de Jogabilidade: tal como no protótipo interno, mas a ser aplicado com o público-alvo.

- Teste de Usabilidade: para obter dados mais realista do sentimento do jogador enquanto joga, identificar desconfortos e erros cometidos pelo jogador que não foram planejados pelo projeto.

Durante a tarefa de criação do protótipo estrutural é necessário que algumas diretrizes de acessibilidade já sejam implementadas a fim de tornar o jogo viável para testes com o público-alvo. Para oferecer uma experiência agradável investindo pouco tempo de trabalho, é incentivado o uso de ativos audiovisuais já prontos (gratuitos ou comprados), pois apesar da baixa fidelidade com o produto final, o foco é a avaliar a reação dos jogadores diante do jogo proposto.

Esta versão do protótipo deve conter 2 ou 3 níveis do jogo que ofereçam uma experiência completa ao jogador. A criação desses níveis ajudará a equipe a mensurar o esforço necessário para criação do jogo completo.

A equipe deve avaliar os resultados dos testes com o protótipo estrutural, aplicar correções e realizar os testes novamente, repetindo esse processo até que a estrutura base do jogo esteja consolidada para seguir com desenvolvimento. Nesse contexto é possível que surjam descobertas mais radicais como: os participantes não se adaptaram à plataforma escolhida (PC, mobile ou outros); a jogabilidade proposta é desconfortável, estressante ou cansativa aos jogadores; os controles são difíceis de manusear e geram muitos erros de entrada de comando; ou outros. A equipe deve estar preparada para realizar grandes mudanças na estrutura do jogo durante esta fase do projeto.

#### **7.4.5 Levantamento de requisitos da plataforma de distribuição**

As plataformas de publicação de jogos possuem políticas e mecanismos próprios que podem incluir taxas por publicação, custo de kit de desenvolvimento para plataforma, padrões e normas a serem atendidas e prazos para avaliação do jogo pela gestora da plataforma antes de liberar a publicação. A exemplo, as plataformas de publicação Steam e Google Play publicam regulamentos, diretrizes, boas práticas e guias sobre como se tornar um criador de aplicativos na plataforma. Tais informações podem ser encontradas através da página inicial da *Steam Direct* (Steamworks, 2020a), na documentação oficial da Steamworks (2020b), na lista de verificação do lançamento do Google Play (2020a) e através da central de políticas do desenvolvedor do Google Play (2020b). Portanto, a equipe deve preparar o ambiente para

publicação do jogo e verificar se seu projeto segue as orientações da plataforma desejada. Resumidamente, é necessário realizar os seguintes passos para publicar em plataformas mais comuns como Steam e Google Play:

- Criar conta na plataforma.
- Pagar taxas para uso da conta de desenvolvedor na plataforma.
- Completar informações sobre a identidade do desenvolvedor (e-mail, endereço, contatos, etc).
- Criar perfil do jogo a ser publicado na plataforma.
- Inserir informações de classificação indicativa.
- Conferir se a versão do SDK (do inglês, Kit de Desenvolvimento Padrão) usado para criação do jogo condiz com as versões mínimas aceitas pela plataforma.
- Checar se o conteúdo inserido no jogo é permitido pelas regras da plataforma e dos países para os quais se planeja alcançar com o jogo.
- Se o jogo será vendido ou se possui transações comerciais no contexto do jogo, é necessário configurar a conta como comerciante e oferecer dados bancários e outras exigências da plataforma.

Se o projeto visa publicar o jogo em forma de software embarcado, deve ser evitado que conteúdos sexuais ou violentos sejam dispostos em ambientes públicos, ao alcance de pessoas menores de 18 anos. De forma geral, para trabalhar fora de plataformas de publicação de jogos convencionais, a equipe deve obedecer regras e leis locais como as classificações indicativas de conteúdo audiovisuais (ROMÃO, 2006).

#### **7.4.6 Backlog do produto**

Os resultados dos testes com os protótipos interno e estrutural e o levantamento dos requisitos para publicação fornecem os dados necessários para mensurar a real complexidade do jogo e identificar outros requisitos que antes não foram concebidos.

A lista de atividades criada durante a modelagem inicial deve ser atualizada para contemplar tudo que for considerado necessário para a realização do jogo e, em seguida, organizada demonstrando as cadeias de dependências que há entre as atividades. Tal como o *backlog* do produto na metodologia Scrum (SCHWABER, 2017), esta lista ordenada é um

documento vivo e deve ser atualizada à medida que o projeto evolui e novos requisitos são descobertos. Para fins de facilitar o entendimento, o termo “backlog do produto” será aproveitado para referenciar esta lista durante todo o trabalho.

#### **7.4.7 Nível de release**

O backlog do produto é a base para a próxima tarefa da fase de pré-produção: inserir uma seção no GDD para definir os “níveis de release”. Os níveis de release são momentos em que certos grupos de ativos, recursos e mecânicas de jogos listadas no backlog do produto estarão desenvolvidas e compiladas em uma versão jogável e passível de testes com usuários. Por exemplo, para um jogo do tipo “*match 3*” (do inglês, combine 3) os níveis de release podem ser distribuídos como:

- Release 1: Funcionamento básico do jogo, o que inclui: mecanismo de entrada dos comandos, a movimentação das peças, os critérios de combinação, as condições de vitória e derrota. Além de ativos incompletos como os efeitos sonoros e imagens dos elementos.
- Release 2: Sistema de cores para daltônicos, caixas de diálogo, novos efeitos sonoros e visuais, adição de trilha sonora de fundo, novos cenários e novas interações especiais dentro do jogo.
- Release 3: Integração com recursos da plataforma de distribuição, opção de configuração de perfis de jogador, sistema de premiação do jogador.
- Release 4: Inclusão de novos elementos no jogo, novos desafios, protótipo da loja de produtos *In-App*, como explicado pela página para desenvolvedores da Apple Store (Apple Developer, 2020).
- Release 5: Sistema de direção assistida, ativos audiovisuais completos, balanceamento de dificuldade durante a partida.
- Release 6: Novos feedbacks para as ações do jogador, controles para ligar e desligar efeitos audiovisuais
- Release 7: Polimentos e ajustes à plataforma de distribuição para iniciar a fase beta.

#### **7.4.8 Revisão da modelagem**

A fase de pré-produção começa usando os recursos elaborados durante a fase de iniciação e percorre uma sequência de tarefas que auxiliam na compreensão total do projeto.

Da prototipação à definição dos níveis de release, cada tarefa oferece mais conhecimento para projetar o jogo preenchendo as incertezas com planos e atividades precisas. Esta jornada, por vezes negligenciada, é essencial para reduzir os riscos de falhas catastróficas no decorrer da produção do jogo e, portanto, vale evidenciar que todos os novos conhecimentos adquiridos devem ser usados para revisar e atualizar o GDD antes de avançar para a próxima fase.

A última tarefa da fase de pré-produção é revisar cada tópico do GDD preliminar criado na fase de iniciação e aprimorá-lo com os novos dados obtidos. Entre as informações a serem atualizadas estão:

- Todos os dados do conceito aproximado: o gênero e subgênero do jogo; mecânicas básicas de jogabilidade; estilo gráfico; arte conceitual; roteiro; plataformas para publicação; controles; estimativa de orçamento e cronograma; recursos específicos essenciais; objetivos específicos.
- Público-alvo: a equipe deve sempre considerar para quem estão criando o jogo.
- Escopo da acessibilidade: as deficiências passíveis de suporte via ajustes de acessibilidade no jogo.
- Diretrizes de acessibilidade: quais os novos desafios acessibilidade e usabilidade foram narrados pelos jogadores durante os testes e como poderiam ser contornados?

## 7.5 Produção

**O que entrega:** a fase de produção acaba quando todos os ativos, telas e mecânicas do jogo estiverem concluídas e integradas com a plataforma de distribuição. O jogo deve estar pronto para ser testado abertamente durante a fase Beta.

**No que consiste:** durante esta fase são realizados os ciclos de produção para criação de todos os níveis, ativos, telas e mecânicas, além da implementação de todas as diretrizes de acessibilidades planejadas. É feito a integração do jogo com a plataforma de distribuição, a criação do material de marketing e prototipação dos módulos de monetização, atualização e suporte com o cliente. O jogo deve passar por baterias de testes internos semanalmente e testes com o público-alvo a cada **nível de release**. Os ciclos de implementação, teste, reimplementação devem continuar até satisfazer as necessidades para a fase beta.

**Como atende aos idosos:** os idosos participam ativamente durante a realização dos testes do jogo e contribuem para a melhoria do produto a cada novo “nível de release”.

Diferente das fases anteriores, a fase de produção não possui uma sequência de tarefas lineares do começo ao fim. Os backlogs do produto vão sendo consumidos à medida que a produção avança através de sucessões de desenvolvimento, teste e atualização: o que é chamado de ciclo de produção.

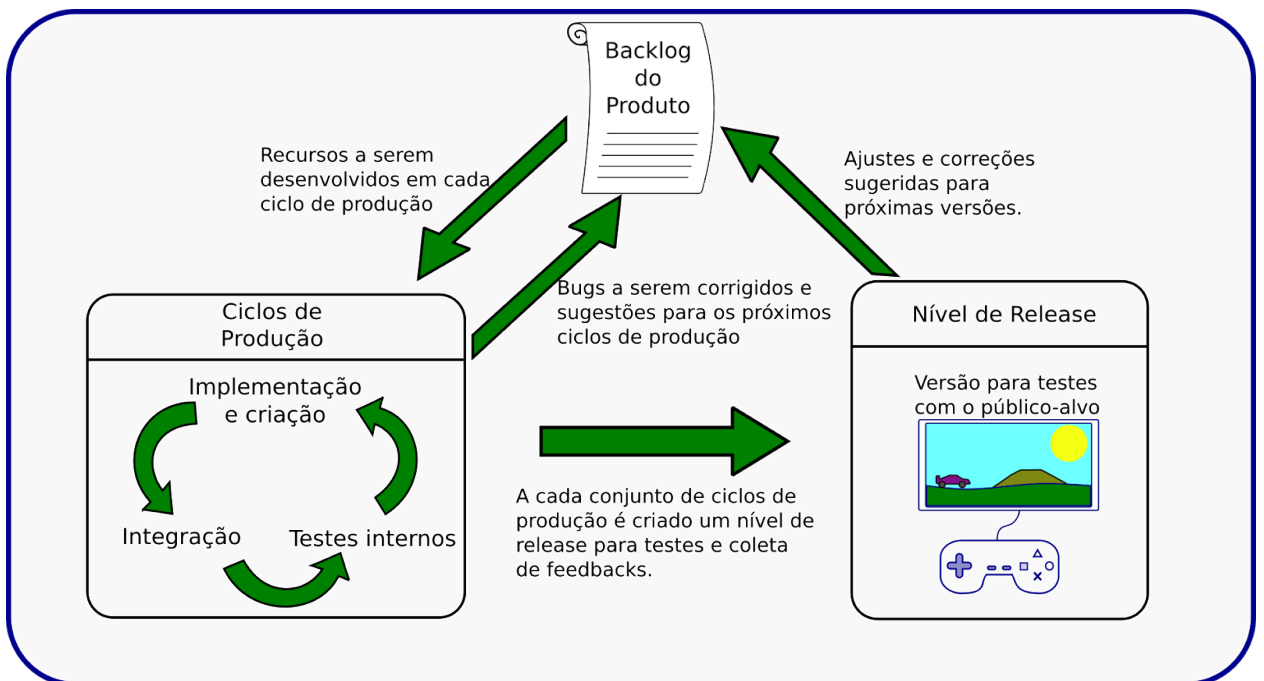


Figura 7.6 - Sequência de atividades sugerida para a fase de Produção. Fonte: do autor.

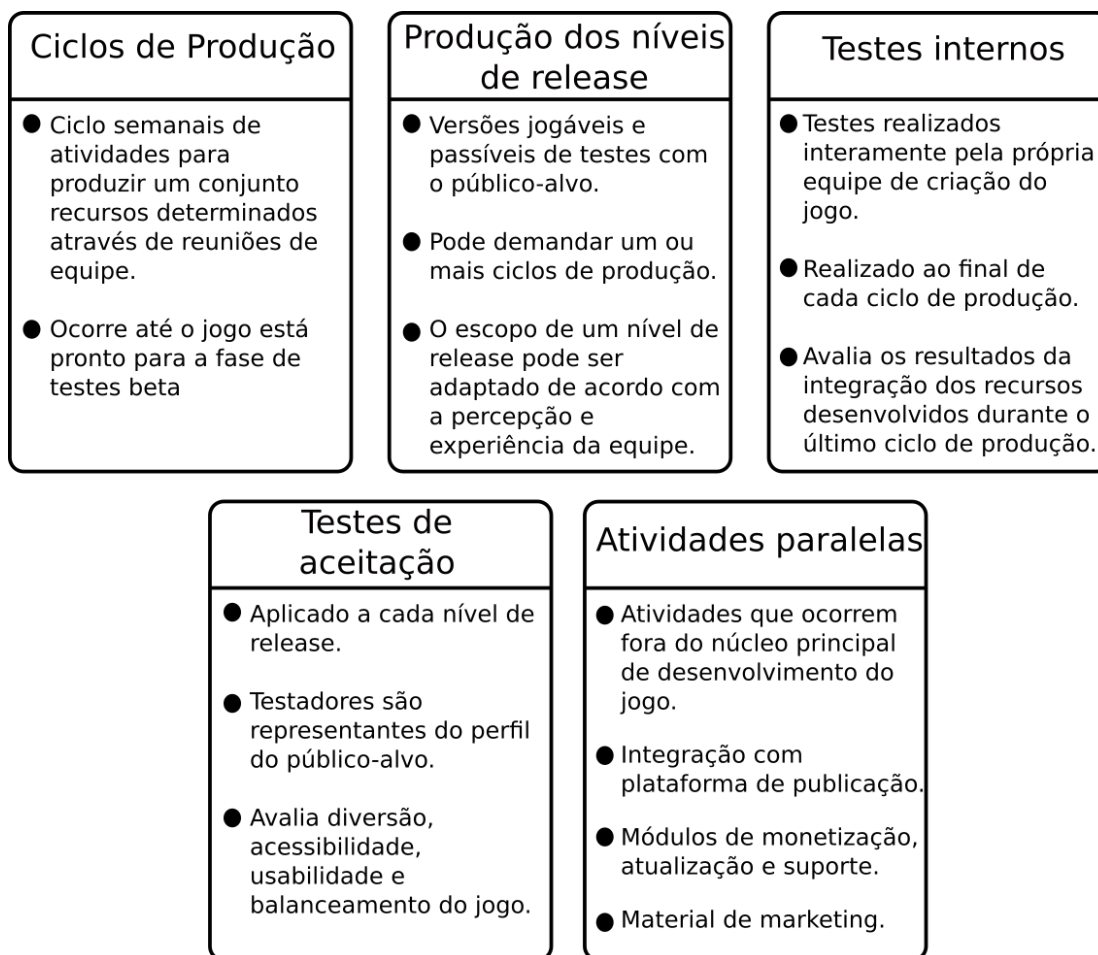


Figura 7.7 - Board das Atividades da fase de Produção. Fonte: do autor.

### 7.5.1 Os ciclos de produção

Um ciclo de produção tem como objetivo alcançar uma meta semanal e cada meta é constituída de um grupo de recursos que a equipe planeja entregar ao final da semana. Ao começo de cada ciclo de produção a equipe deve se reunir para atualizar as metas e o plano de testes que irá avaliá-las. A criação de um jogo trata-se de um processo que é, acima de tudo, criativo, portanto deve-se evitar sobrecargas de trabalho. Neste contexto, é recomendado ciclos de 5 ou 6 dias de produção, sendo o último dia reservado para integração dos novos recursos e testes internos. Bugs encontrados devem ser tratados como recursos a serem trabalhados no ciclo de produção a seguir.

Quando todas as mecânicas, ativos, interfaces e configurações estiverem prontas e trabalhando em sinergia e com todos os bugs corrigidos (num cenário hipotético ideal), a fase de produção estará concluída e a equipe deve se preparar para a disponibilizar o jogo para testes na fase beta.

### **7.5.2 Diretrizes de acessibilidade para fase de produção**

Durante a fase de produção existe a oportunidade de implementar a maioria das diretrizes de acessibilidade propostas pelo projeto. Assim, esta proposta recomenda que sejam implementadas:

- Todas as diretrizes de baixo custo de implementação e alto alcance que condizem com o tipo de jogo sendo criado.
- Diretrizes envolvidas diretamente com a jogabilidade e balanceamento de dificuldades.
- O máximo das recomendações sobre como evitar desconfortos ao jogador, por exemplo: evitar excesso de animações nos menus e oferecer uma opção de desabilitá-las no menu de acessibilidades; atividades com tempo limitado; e fundos com muitos elementos de distração.
- Diretrizes mais complexas devem ser implementadas de acordo com o cronograma do projeto. Se a equipe considerar uma diretriz importante, mas não conseguir implementá-la a tempo, pode agendar este recurso para ser entregue após a publicação do jogo, em forma de uma atualização.

### **7.5.3 Produção dos níveis de release**

Um nível de release pode demandar um ou mais ciclos de produção e é concluído após a realização dos testes de aceitação e coleta dos feedbacks que contribuirão para os ajustes dos próximos ciclos de produção, possivelmente em forma de novas funcionalidades para o backlog do produto.

A equipe define quantos ciclos de produção serão necessários para alcançar cada nível de release. Os níveis de release estabelecidos durante a fase de pré-produção são referências para orientar o andamento da produção, mas podem ser ajustados de acordo com a percepção e experiência da equipe.

Por se tratar de uma versão do jogo que será testada com jogadores do público-alvo é importante que novos recursos de acessibilidade sejam adicionados ou ajustados a cada nível de release.



#### 7.5.4 Testes internos e testes de aceitação

Durante a fase de produção é possível diferenciar dois tipos de atividade de testes. O primeiro são os **testes internos** que são realizados ao fim de cada ciclo de produção por membros da própria equipe, cujo principais objetivos e métodos sugeridos são:

- Testar o funcionamento de novos recursos integrados ao jogo.
  - Teste Fumaça: por ser um teste rápido e oferecer informações importantes sobre o funcionamento geral do jogo, é conveniente iniciar a bateria de teste usando o método fumaça.
  - Teste de Regressão: é um meio efetivo de avaliar se a integração do conteúdo produzido durante o ciclo de produção atual não adicionou erros no funcionamento do conteúdo já existente anteriormente.
- Avaliar a performance e compatibilidade de dispositivos.
  - Teste de Performance: é importante verificar se os novos recursos produzidos não comprometeram a performance do jogo.
  - Teste de compatibilidade: no contexto de jogos para dispositivos móveis, o jogo deve ser testado com a maior variedade possível dispositivos para se identificar e corrigir incompatibilidades ou mal funcionamentos.
- Avaliar o balanceamento de dificuldade do jogo.
  - Teste de Jogabilidade: a melhor forma de testar o balanceamento do jogo é jogando o jogo, por isso os testes de jogabilidade são tão importantes e realizados em várias fases do projeto.
- Revisar as diretrizes de acessibilidade e conformidades com padrões.
  - Teste de Regressão: é a melhor maneira de verificar se as mudanças realizadas no jogo causaram algum impacto negativo sobre as implementações das adaptações de acessibilidade.
  - Teste de Conformidade: é o momento de verificar se mudanças realizadas no jogo afetaram negativamente alguma implementação de acessibilidade e usabilidade.
- Localizar bugs.
  - Teste Ad Hoc: a ideia é forçar situações que poderiam quebrar o funcionamento do jogo.

- Teste Funcional: por demandar mais tempo e esforço, o teste funcional pode ser agendado para os ciclos de produção da semana de entrega de um nível de release.

O segundo tipo são os **testes de aceitação** aplicados a cada nível de release. Estes requerem a participação de pessoas selecionadas previamente e que se encaixam no público-alvo do projeto. Assim como no teste com protótipo estrutural, estes testes devem incluir pessoas agravadas de limitações e que possam avaliar os recursos de acessibilidade implementados no jogo. No caso das pessoas idosas, que normalmente são acometidas de uma ou mais limitações, é interessante envolver participantes de diferentes faixas etárias e diferentes perfis socioculturais. Os métodos de testes sugeridos nessa etapa são o teste de jogabilidade e o teste de usabilidade, tal como no protótipo estrutural. Os dados dos testes de aceitação devem ser obtidos através da observação do comportamento dos jogadores e por coleta de feedbacks com questionários e conversa direta com os jogadores. Esses dados devem ajudar a avaliar pontos como:

- O potencial de diversão do jogo.
- A receptividade e o impacto das diretrizes de acessibilidade implementadas.
- A compreensão da interface e das mecânicas do jogo.
- A progressão da dificuldade.

Os resultados dos testes internos e testes de aceitação devem ser documentados e analisados por toda equipe. Correção de bugs, inconformidade e “desconfortos” identificados devem ser tratados como recursos a serem trabalhados nos próximos ciclos de produção. A tabela 1 mostra um esquema de como representar os problemas identificados durante os testes.

<Nome do teste aplicado>	Versão do jogo: <exemplo: dev-0.03>
Problema:	Classificação: (Gravíssimo/Grave/Intermediário/Médio/ Menor)
Descrição:	<IMAGEM>
Solução:	

Comentário do desenvolvedor:
------------------------------

Tabela 1: Modelo para documentação dos problemas identificados durante os testes.

### 7.5.5 Atividade paralelas

Dentre as atividades do backlog do produto do projeto, existem aquelas que tratam de necessidade fora do núcleo do jogo e que valem serem lembradas aqui.

- **Integração com a plataforma de distribuição.**

É importante destacar alguns ciclos de produção para tratar da integração do jogo com a plataforma de distribuição. Esse processo pode ser demorado e requerer séries de correções e adaptações, portanto não deve ser deixado para o final da fase de produção.

- **Protótipo dos módulos de monetização, atualização e suporte com o cliente.**

Alguns jogos são **monetizados** através de vendas *In-App* e esses itens virtuais precisam ser registrados com a plataforma de distribuição ou outro provedor, os preços precisam ser ajustados e a loja virtual preparada para essas transações.

A forma como o jogo será **atualizado** após a publicação depende do seu gênero, da plataforma onde foi publicado e das tecnologias envolvidas na sua criação. Seja qual for, o mecanismo de atualização precisa estar funcionando antes de iniciar a fase beta.

Visto as dificuldades envolvidas no processo de criação de um jogo acessível, manter um canal para comunicação e **suporte ao público** pode ser uma boa alternativa para receber feedbacks constantes durante a fase beta e após a publicação. Além disso é importante manter as mídias como site, rede social e página de publicação do jogo sempre atualizadas e com informações sincronizadas, de forma que o jogador escolha a mídia que mais lhe for acessível para buscar informações ou publicar suas avaliações.

É aconselhado que os recursos descritos acima sejam implementados (quando aplicável ao tipo de jogo) em forma de protótipo estrutural (funcional, mas sem grandes acabamentos), pois o importante neste momento é concluir a fase de produção e iniciar os testes beta.

- **Criação do material de marketing.**

A criação da maioria do conteúdo audiovisual do jogo ocorre durante a fase de produção. Este conteúdo pode ser aproveitado para elaboração de materiais de divulgação do jogo, demonstrar o andamento do projeto para a comunidade que o segue, fazer chamadas para os testes betas e atrair a atenção do público-alvo.

## 7.6 Beta

**O que entrega:** um jogo concluído e polido no seu estágio final. Todas os módulos e integrações com as plataformas de publicação devem estar prontas e aguardando apenas o lançamento oficial.

**No que consiste:** o jogo é apresentado ao público para testes e os módulos prototipados durante a fase de produção devem ser concluídos paralelamente à coleta dos feedbacks e correções de bugs. Não se espera grandes mudanças no comportamento do jogo nesta fase, apenas ajustes e melhorias no que já está pronto. O processo de testes e polimentos deve continuar até chegar ao ponto de publicação.

**Como atende aos idosos:** como todos os demais recursos do jogo, as diretrizes de acessibilidades são revisadas e polidas para atingir um elevado grau de usabilidade ao público-alvo. O jogo deve ser divulgado como acessível e citando quais adaptações foram feitas para atendê-los.

Um jogo chega a fase beta quando o conteúdo está completo. Isso significa que todo o conteúdo foi criado e implementado, restando apenas o trabalho de correção de bugs e polimentos. No processo JAI foi adicionado a isso a tarefa de finalizar os módulos adicionais, visto que muito esforço já foi investido em testes durante a criação dos níveis de release da fase de produção.

A fase beta mantém o sistema de trabalho em ciclos de produção semanais, porém se difere da fase de produção pelo tipo de conteúdo a ser produzido. Enquanto a fase de produção cria novos conteúdos e os níveis de release para demonstração com os testadores selecionados, a fase beta cria versões melhoradas do mesmo conteúdo e usa o sistema de atualizações para apresentar estas novas versões ao fim de cada ciclo de produção.

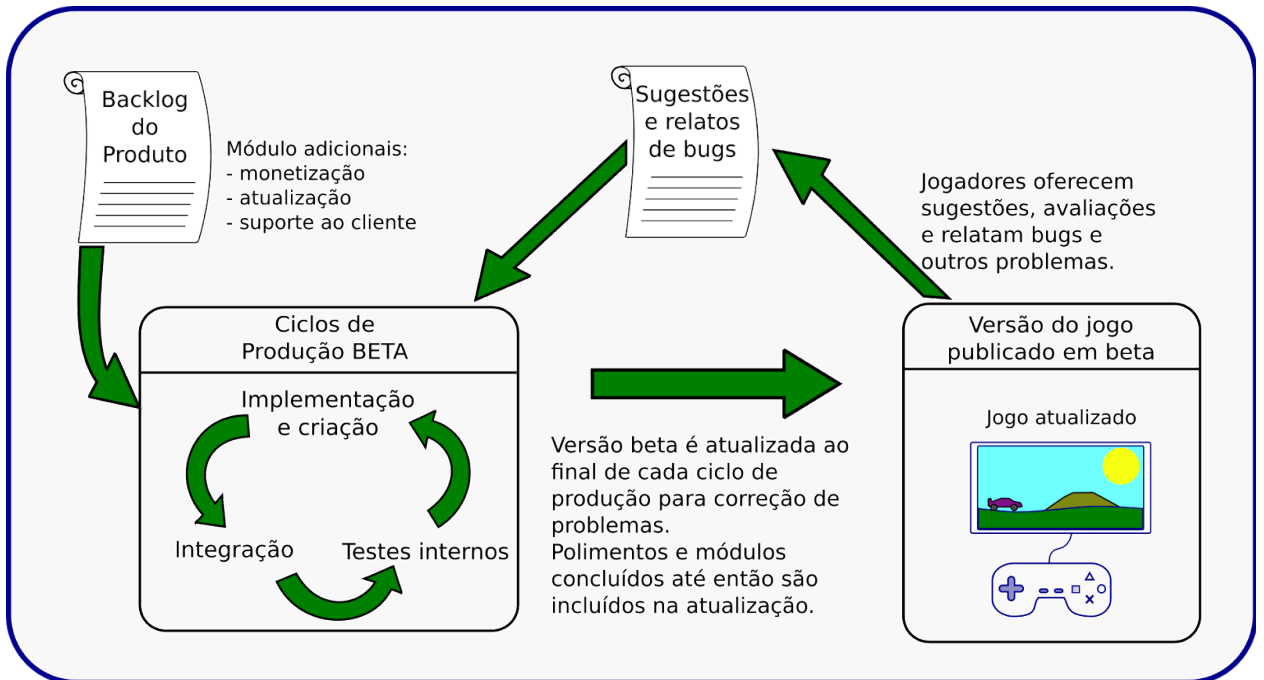


Figura 7.8 - Sequência de atividades sugerida para a fase Beta. Fonte: do autor.

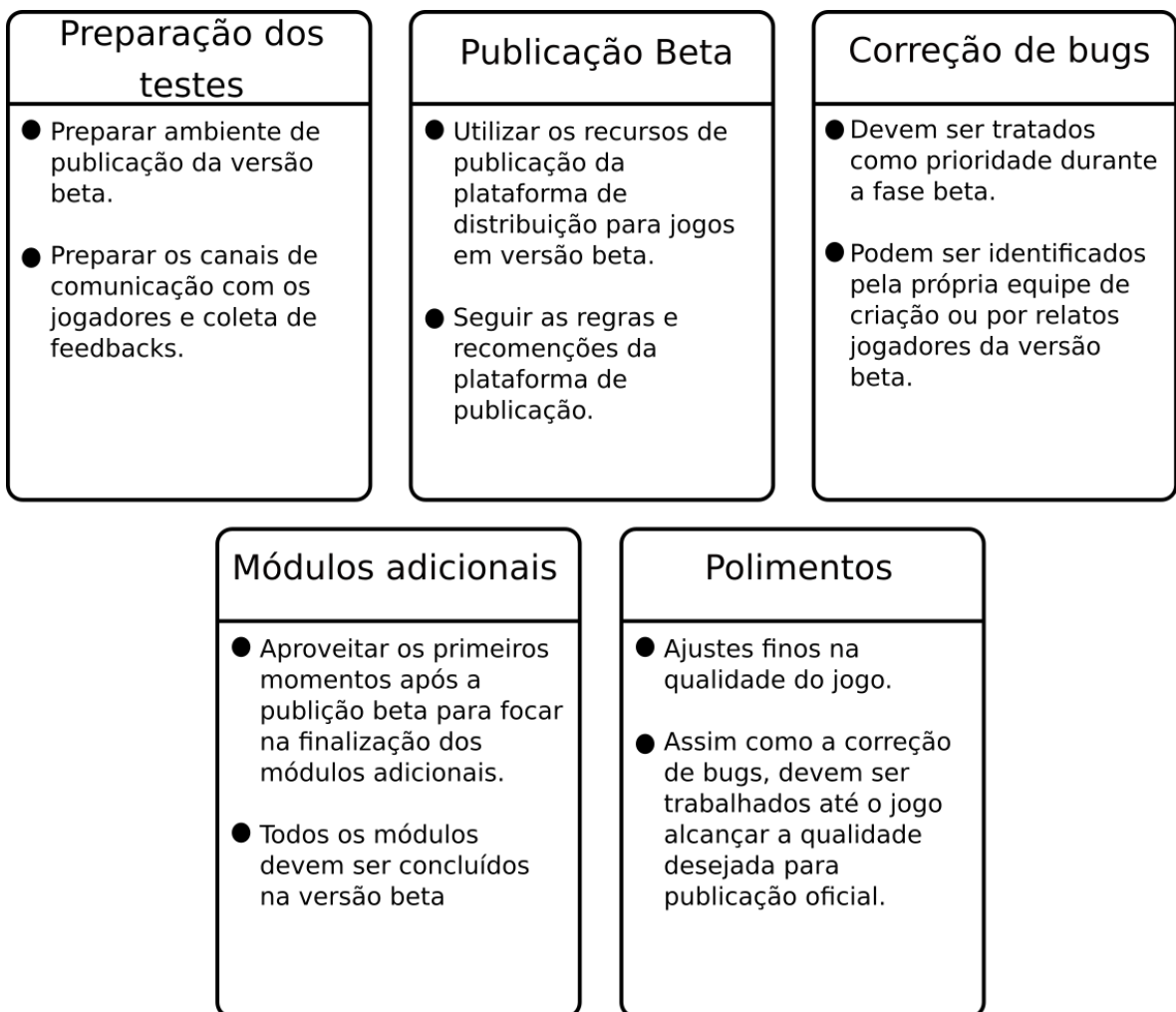


Figura 7.9 - Board das Atividades da fase Beta. Fonte: do autor.

### **7.6.1 Preparação dos testes**

Ao chegar na fase beta, o jogo deve possuir a qualidade e todos os recursos necessários para ser testado por ampla diversidade de pessoas, sendo assim, é recomendado realizar campanhas de divulgação e angariar um grande número de pessoas para testarem o jogo. Como falado anteriormente, deve-se evidenciar que o jogo foi construído com objetivo de atender às necessidades especiais de idosos e portadores de deficiências, citar quais recursos foram implementados para atender ao público e como usá-los.

É importante manter canais de comunicação com os testadores sempre atualizados e, se possível, oferecer um endereço de página web ou fórum específico para reportar bugs, oferecer sugestões ou realizar avaliações. Se o projeto não possuir um site ou fórum, é sugerido a divulgação de questionários online junto à divulgação do jogo. Os questionários devem conter campos que permitam aos testadores avaliarem a jogabilidade, diversão, usabilidade, a progressão da dificuldade, além de poderem descrever bugs e desconforto percebidos durante o jogo.

### **7.6.2 Publicação Beta**

A forma como o jogo será publicado em versão beta depende da plataforma de distribuição escolhida. A exemplo, a Google Play Store oferece a opção de publicar o jogo para teste interno, teste fechado e teste aberto, antes de publicá-lo em produção. O teste aberto refere-se à versão beta, disponibilizando o jogo para um grupo maior de possíveis testadores. A plataforma Steam também oferece recursos para publicar o jogo em forma de “beta fechado” e “acesso antecipado”. O acesso antecipado equivale ao teste aberto, onde pessoas interessadas podem experimentar o jogo e ajudar reportando bugs e oferecendo sugestões. Para mais informações: Google Play Console (2020c), STEAMWORKS (2020c), STEAMWORKS (2020d).

### **7.6.3 Correção de bugs**

O jogo está completo e com todos ativos prontos, porém ainda há bugs a serem identificados e corrigidos. A correção de bugs e outros problemas relatados pelos testadores deve ser prioridade e a equipe deve se organizar para tratá-los com rapidez, enquanto que

sugestões devem ser documentadas e avaliadas se serão implementadas durante a fase beta ou se ficarão no plano de manutenção e expansão do jogo.

#### **7.6.4 Módulos adicionais**

Enquanto ocorrem os testes betas e os feedbacks são enviados, a equipe deve aproveitar para finalizar os módulos (monetização, atualização e suporte ao cliente) prototipados durante a fase de produção.

#### **7.6.5 Polimentos**

Os polimentos são modificações sutis realizadas para o jogo oferecer melhor imersão e explorar a sensação de prazer nos jogadores. Eles costumam ser ajustes finos nas mecânicas, inclusão ou alteração de efeitos audiovisuais, correções nas paletas de cores, balanceamento nos desafios do jogo, inclusão de feedbacks nas ações do jogador e melhorias na performance do jogo.

À medida que as sugestões e os relatos de bugs forem reportados, a equipe deve se organizar e dividir os esforços entre a conclusão dos módulos, a realização dos polimentos e correção dos bugs. Todos os módulos devem estar completos ao fim da fase beta e os polimentos e correções de bugs devem continuar até o jogo alcançar o nível de publicação oficial.

### **7.7 Publicação**

**O que entrega:** o jogo publicado na loja de aplicativo ou outra plataforma de distribuição de jogos, a documentação do jogo atualizada com o adicional dos planos de manutenção e expansão.

**No que consiste:** publicação do jogo na plataforma de venda e distribuição, reuniões para compartilhamento de conhecimento e alinhamento dos planos de manutenção e expansão.

**Como atende aos idosos:** os idosos terão acesso a um produto que literalmente ajudaram a construir e poderão desfrutar de um lazer que respeitam as suas condições. O site e/ou embalagem do jogo deve fornecer detalhes dos recursos de acessibilidade disponíveis no jogo e como habilitá-los (quando se aplica).

A publicação, ou também chamado de lançamento, é o marco mais desejado para qualquer projeto de jogo digital. Após a modelagem do projeto, prototipagem, criação de conteúdos, testes, correções e polimentos, o jogo finalmente é oficialmente lançado.

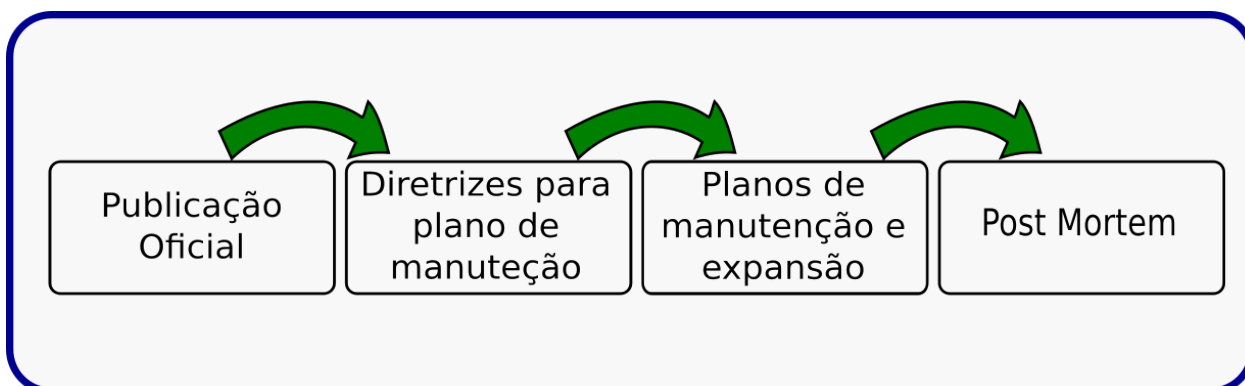


Figura 7.10 - Sequência de atividades sugerida para a fase de Publicação. Fonte: do autor.

Publicação Oficial	Diretrizes para o plano de manutenção	Planos de manutenção e expansão	Post Mortem
<ul style="list-style-type: none"> <li>● O projeto alcança seu ápice de qualidade e o jogo é lançado oficialmente.</li> <li>● Foco no diálogo com a comunidade do jogo e na produção de propaganda.</li> </ul>	<ul style="list-style-type: none"> <li>● Identificar a diretrizes que não foram atendidas durante a fase de produção</li> <li>● Revisar a diretrizes que podem ser implementas após a publicação do jogo.</li> </ul>	<ul style="list-style-type: none"> <li>● Atualizar o GDD com dados mais recentes.</li> <li>● Criar a lista de recursos a serem produzidos após a publicação do jogo e estabelecer um plano para realização das atualizações</li> </ul>	<ul style="list-style-type: none"> <li>● Compartilhamento de conhecimento.</li> <li>● Definir o ponto de encerramento do projeto.</li> </ul>

Figura 7.11 - Board das Atividades da fase de Publicação. Fonte: do autor.

### 7.7.1 Publicação oficial

Se o jogo encontra-se finalizado, polido, testado, todos os módulos adicionais funcionando, todos os requisitos da plataforma de distribuição já satisfeitos e o marketing do jogo em execução, então está tudo pronto para lançar o jogo oficialmente. Após publicá-lo, a equipe deve acompanhar as vendas (ou simplesmente downloads), manter a comunicação constante com a comunidade ao redor do jogo e dedicar-se intensamente ao marketing.



### **7.7.2 Diretrizes de acessibilidade para o plano de manutenção**

Desde a fase de pré-projeto, esta proposta priorizou as diretrizes de acessibilidade com menor complexidade, deixando as diretrizes mais complexas para serem implementadas de acordo com o espaço no cronograma. Com o fim dos ciclos de produção e o jogo já publicado, é recomendado que o plano de manutenção inclua a implementação de:

- Diretrizes de alto impacto, mas de alto custo de implementação. Por exemplo: usar som e música distintos para todos os objetos e eventos do jogo; fornecer uma opção para desativar ou ocultar todos os elementos não interativos; incluir tolerância a erros como clicks involuntários; e outros.
- Diretrizes importantes que não foram implementadas a tempo durante a fase de produção.

### **7.7.3 Planos de manutenção e expansão**

Além de acompanhar o andamento do jogo publicado, a equipe ainda deve atualizar a documentação com os dados mais recentes e criar o plano de manutenção e expansão. Quando o GDD foi criado nas fases de iniciação e pré-produção, foram definidos alguns recursos que possivelmente só seriam implementados após o lançamento do jogo. Estes recursos entram como conteúdo de atualização e expansão para o jogo e podem incluir características como: novos níveis, novos itens, recursos de acessibilidade, sugestões de jogadores de teste beta e até mesmo correções de bugs pontuais que não foram solucionados a tempo. É interessante manter a comunidade atualizada sobre os recursos e mudanças que estão por vir em novas atualizações do jogo. Além disso, os jogadores costumam ser fontes de sugestões e relatos de bugs que podem ser agregados às atividades de manutenção e expansão.

A produção desse plano pode seguir o modelo de criação do backlog do produto - uma lista de atividades ordenadas por prioridades e dependências - seguido de um cronograma para execução desta lista e incluindo previsões de datas para lançamento das atualizações.

### **7.7.4 Compartilhamento de conhecimento (post mortem)**

As atividades de desenvolvimento são desaceleradas e a equipe pode se reunir para discutir assuntos como: o conhecimento adquirido durante o projeto, o que deu certo, as escolhas erradas e suas consequências, como devem se comportar em um próximo projeto e o

ponto de encerramento do projeto - estado em que o jogo será dado como finalizados e não receberá mais atualizações.

## **8. Sequência Simplificada do Processo JAI**

Esta seção apresenta um passo-a-passo simplificado da aplicação do JAI na criação dos principais artefatos do processo, ou ainda para colocá-lo em prática em projetos de baixa complexidade. Esta sequência simplificada não substitui a leitura do processo JAI, mas é uma abstração que facilita a compreensão do processo como um todo.

### **8.1 FASE DE INICIAÇÃO: Criando documentação inicial do jogo**

Um criador de jogos digitais teve uma ideia que acha interessante transformá-la em jogo - é um jogo casual de montagem e construção de objetos semelhante a um quebra-cabeça 3D, mas o jogo deve ser acessível a todas as idades (incluindo adultos mais velhos e idosos) e fomentar a criatividade dos jogadores. O modelo de GDD no Apêndice A é uma instância da modelagem desse jogo.

Partindo dessa motivação, o primeiro passo é delimitar o público-alvo para qual o jogo deve ter maior aceitação. Jogos casuais costumam ter boa recepção com o público feminino maior de 35 anos, somando a isso o interesse em criá-lo como acessível, a definição do público-alvo pode ser descrita como:

- Sexo: unisex, maioria feminina.
- Idade: 35~70 anos.
- Interesses: Jogos casuais, desafios relaxantes, não competitivo, jogos de passatempo.

Desta forma, a modelagem inicial pode ser construída na seguinte ordem:

1. Delimitar o público-alvo;
2. Escolher as mecânicas, o estilo de sons e artes, os controles, as plataformas de publicação, o enredo e as demais características que definem o conceito aproximado do jogo com as características que melhor se enquadram ao público;
3. Definir os periféricos, dispositivos e softwares necessários para produção do jogo e gerenciamento do projeto;

4. Esboçar as estratégias de monetização, marketing e manutenção de acordo com o prazo, quantidade de membros na equipe, especialidades dos membros e o orçamento inicial do projeto;
5. Realizar uma análise de mercado e identificar as características que o jogo proposto tem em relação aos já disponíveis no mercado;
6. Analisar a viabilidade de prosseguir com a criação do jogo com base nos objetivos do projeto e a análise de mercado;
7. Definir as metas de acessibilidade tal como no exemplo da seção “Metas de Acessibilidade” do Apêndice A. O Apêndice B oferece uma listagem de várias orientações de acessibilidade que podem ser consultadas neste momento;
8. Por fim, criar uma lista de atividades macros que são necessárias para produção do jogo e organizar todas as informações no formato de um GDD;

## **8.2 FASE DE PRÉ-PRODUÇÃO: Prototipagem e revisão da modelagem**

Após criar a modelagem inicial, é importante dedicar um tempo para conversar com pessoas com perfil do público-alvo e cadastrar uma lista de candidatos a testadores do jogo. Quanto maior a diversidade dos testadores, melhor será a qualidade do retorno de informação obtido nos testes. Portanto a fase de pré-produção do JAI segue esses passos:

1. Selecionar testadores para o jogo;
2. Criar um protótipo interno baseado na modelagem inicial e realizar testes com os membros da equipe. Se necessário, repetir esta tarefa e atualizar o conceito aproximado com os aprendizados;
3. Criar o protótipo estrutural (seguindo as metas de acessibilidade) e testá-lo com o público-alvo. Repetir essa tarefa até atingir um protótipo com estrutura sólida para o desenvolvimento;
4. Estudar os requisitos para uso da plataforma de publicação e , se possível, realizar uma publicação de teste;
5. Criar o backlog do produto com todas as atividades necessárias para produção do jogo;
6. Definir os níveis de release (marcos de desenvolvimento) do jogo;
7. Revisar a modelagem e construir o cronograma detalhado para criação do jogo;

### **8.3 FASE DE PRODUÇÃO: Construindo o jogo através de ciclos iterativos**

A criação do jogo deve ser dividida em ciclos de produção para executar as atividades descritas no backlog do produto. O backlog do produto deve ser completamente consumido ao fim da fase de produção, incluindo a prototipação dos módulos adicionais (cuja a finalização se dá na fase beta). As atividades simplificadas dessa fase são:

1. Definir o nível de release e selecionar as atividades e recursos a serem desenvolvidas para alcançá-lo. É importante que cada níveis de release inclua ao menos uma nova adaptação de acessibilidade ou melhoria em uma já implementada;
2. Dividir as atividades em ciclos de produção semanais contando com o último dia sempre para testes e integrações;
3. Executar os ciclos de produção até alcançar o nível de release e levar a versão produzida a testes com o público-alvo de testadores pré-selecionados (incluindo idosos e portadores de deficiências);
4. Analisar como os testadores avaliaram a acessibilidade, usabilidade, balanceamento e diversão do jogo. Transformar as recomendações e reclamações em novas atividades no backlog do produto;
5. Repetir a sequência de 1 a 4 até o jogo alcançar um nível de qualidade para publicação beta e os módulos adicionais estejam prototipados e funcionais.

### **8.4 FASE BETA: Trabalhando com a versão beta do jogo**

Durante a fase beta é mantida a estrutura de trabalho em ciclos de produção, se diferenciando apenas pelos objetivos - polimento, correção de bugs e finalização de módulos adicionais do projeto, vez criação de conteúdos exclusivamente novos. O backlog do produto ainda será usado nesta fase apenas para guiar a finalização dos módulos de monetização, atualização e suporte ao cliente. A fase beta realiza os seguintes passos:

1. Publicar o jogo em versão beta para coleta de *feedbacks*;
2. Organizar os ciclos de produção para atender às seguintes atividades:
  - a. finalizar os módulos adicionais;
  - b. realizar polimentos na jogo (audiovisuais, jogabilidade e acessibilidade);
  - c. correção de bugs relatados durante os testes betas;
3. Manter o jogo beta sempre atualizado a cada fim de ciclo de produção semanal;
4. Repetir esse ciclo até o jogo alcançar o nível desejado para a publicação.

## 8.5 FASE DE PUBLICAÇÃO: Lançamento oficial

Etapa final na criação do jogo.

1. O jogo é publicado oficialmente na plataforma de distribuição escolhida;
2. Deve ser criado um plano de manutenção e expansão do jogo incluindo atualizações nas metas de acessibilidade;
3. As documentações do jogo podem ser atualizadas e a equipe deve se reunir para compartilhar conhecimento e definir o ponto de encerramento do projeto.

## 9. Resultados e Discussões

O processo JAI, proposto para orientar criadores de jogos interessados em alcançar ao público idosos e portadores de deficiências, foi formulado a partir das experiências do autor com o apoio de pesquisas e trabalhos propostos por outros criadores de jogos. O produto desse esforço foi a proposta de um processo desprendido dos modelos tradicionais de criação de software, mas que não negligencia a importância do planejamento e ordem nas operações para assegurar maior qualidade do jogo no momento de sua publicação.

### 9.1 Análise Comparativa

Os processos de desenvolvimento de jogos propostos por Ramadan (2013), Hendrick (2009) e a versão atualizada apresentada pela empresa Blitz Game Studio (2020), todos em algum momento citados pela revisão sistemática de Aleem (2016), demonstram perspectivas diferentes do ciclo de vida do desenvolvimento de jogos e colaboraram com suas ideias e peculiaridades na elaboração do processo JAI.

As tabelas 2, 3 e 4 sintetizam como o processo JAI se comporta em relação a algumas das referências que contribuíram para sua criação, de acordo com as fases.

<b>Ramadan (2013)</b>	<b>Em relação ao JAI</b>
<b>Iniciação:</b> É a criação do conceito inicial do tipo de jogo que será produzido.	<b>Iniciação:</b> O JAI busca ir além do conceito inicial do jogo, mas um conceito inicial para o projeto como um todo, incluindo pontos como análise de mercado, contato com o público e

	levantamento das ferramentas necessárias. Tudo compilado no GDD preliminar.
<p><b>Pré-produção:</b> Envolve a criação e revisão da modelagem (GDD), junto à prototipação do jogo. São definidos 2 fases de protótipos:</p> <ol style="list-style-type: none"> <li>1. Fundação - é uma versão usada para demonstrar a ideia central de gameplay e avaliar o critério de diversão</li> <li>2. Estrutural - é uma versão refinada da fundação e, demonstra e testa critérios funcionais lógica, aritmética e regras do jogo além da diversão.</li> </ol> <p>A pré-produção acaba quando as revisões e mudanças na modelagem são aprovadas e documentadas no GDD.</p>	<p><b>Pré-produção:</b> Inspirados pelas ideias de Hamadan, a proposta deste trabalho visa a criação de 2 protótipos (interno e estrutural) durante a pré-produção, porém diferenciando-se no objetivo por trás destes protótipos, sendo o primeiro para alinhamento interno com a equipe, enquanto o segundo é mais aprimorado e já voltado para testar com o público-alvo.</p>
<p><b>Produção:</b> É o processo central que envolve a criação e integração de códigos e assets. São definidos dois níveis de protótipos nessa fase.</p> <ol style="list-style-type: none"> <li>1. Detalhes formais - é uma versão melhorada do protótipo estrutural e contém mais mecânicas e assets completos. Trata da correção de bugs, inclusão de novos recursos, aprimoramento de performance e balanceamento do jogo.</li> <li>2. Refinamento: é um protótipo completo que representa a versão polida do jogo. Focado em melhorar a diversão e acessibilidade. Apenas mudanças sutis nessa fase do processo.</li> </ol>	<p><b>Produção:</b> Ramadan mantém o esquema de dois níveis de protótipos na fase de produção e usa esse níveis de protótipos como marcos do estado do jogo, determinando até mesmo a forma de aplicação dos testes e os critérios de avaliação.</p> <p>A proposta do processo JAI divide a produção por ciclos de produção e “níveis de release”. O conteúdo produzido a cada ciclo de produção e o que será apresentado em cada “nível de release” é algo determinado pela equipe durante a fase de pré-projeto (no GDD) e adaptado durante a fase de produção.</p> <p>O JAI também considera que a fase de produção deve criar ativos que estão fora do núcleo de desenvolvimento do jogo, mas que devem ser considerados quando se trata de</p>

	<p>um projeto como um todo: material de marketing e a prototipação dos módulos de monetização, atualização e suporte com o cliente.</p>
<p><b>Testes:</b> São testes internos feitos para avaliar a usabilidade e jogabilidade, e se aplicam de forma específica para cada nível do protótipo de produção.</p> <p>1. Testes dos detalhes formais - conduzido por testes jogando (playstest) e usado para testar a funcionalidade de recursos e o balanceamento de dificuldades. Os bugs identificados durante os testes de funcionalidade devem ser documentados e analisados. Os testes de dificuldade são feitos com várias modificações durante o jogo para checar se a mudança torna o jogo muito difícil, muito fácil, ou bem equilibrado.</p> <p>2. Testes no Refinamento - avalia a diversão e acessibilidade. É feito através de playtest com coleta de feedbacks para identificar se o jogo é chato, desafiador, frustrante, etc. A acessibilidade pode ser avaliada observando o comportamento do jogador em teste. Se o jogador tiver dificuldade em entender o jogo, quer dizer que o jogo não é acessível.</p> <p>Os resultados obtidos durante os testes indicam se os desenvolvedores passam para fase Beta, ou se retomam novos ciclos de produção.</p>	<p><b>Testes:</b> Ramadan aplica os testes e os avalia de acordo com o nível do protótipo (detalhes formais ou refinamento).</p> <p>O JAI também determina dois tipos de testes. O primeiro são os <b>testes internos</b> que são realizados ao fim de cada ciclo de produção por membros da própria equipe. O segundo tipo são os <b>testes de aceitação</b> aplicados a cada “nível de release” e demandam a participação de pessoas que se encaixam no público-alvo do projeto. Assim como Ramadan avalia critérios de usabilidade e acessibilidade durante os testes no <b>protótipo de refinamento</b>, o JAI avalia estes critérios durante os <b>testes de aceitação</b>.</p> <p>Diferenciando-se pela exigência de participantes dos grupos de acessibilidade (idosos e portadores de deficiências) já que visa julgar as adaptações implementadas para atendê-los.</p>
<p><b>Beta:</b> A fase Beta é definida pelos testes externos e utilizam os mesmo métodos de testes da fase de produção. Os testes podem ser aplicados</p>	<p><b>Beta:</b> A proposta JAI não foca em “closed beta” durante esta fase, pois já realizou sessões de testes com grupos pré-selecionados durante</p>

<p>a um grupo pré-selecionado (closed beta) ou podem ser abertos ao público que se cadastram para participar (open beta).</p> <p>Os feedbacks obtidos nos testes podem guiar novos ciclos de produção ou avançar o projeto para o lançamento.</p>	<p>a fase de produção. O foco do JAI nessa fase é finalizar os módulos adicionais, corrigir bugs e realizar polimentos.</p>
<p><b>Lançamento:</b> publicação do jogo, documentação do projeto, compartilhamento de conhecimentos, planos de manutenção e expansão.</p>	<p><b>Publicação:</b> Segue o mesmo padrão de atividades da proposta de Ramadan.</p>

Tabela 2: Comparativos entre a proposta de Ramadan e o JAI.

<b>Hendrick (2009)</b>	<b>Em relação ao JAI</b>
<p><b>Protótipo:</b> Selecione as ferramentas de trabalho necessárias para criação do jogo. Defina o estilo de arte, crie a modelagem inicial e o primeiro esboço de GDD. Crie protótipos de animação e gameplay.</p>	<p><b>Iniciação:</b> Hendrick demonstra uma abordagem extremamente prática durante toda sua fase inicial (protótipo), desconsidera questões como análise de mercado, conhecimento do público-alvo e acessibilidade e parte direto para documentação e prototipação do jogo.</p>
<p><b>Pré-produção:</b> Criar duas ou três níveis completos do jogo, incluindo as recompensas e os “avanços”.  Criar um nível do jogo completamente usando o mesmo método de produção para mensurar os custos de desenvolvimento do jogo completo.  Completar o GDD.</p>	<p><b>Pré-produção:</b> A ideia de Hendrick em mensurar o esforço para criar todo o projeto a partir da criação completa de poucos níveis do jogo foi levada em consideração pelo JAI e foi aplicada para construção do protótipo estrutural. Além disso, foram adicionados detalhes sobre como desenvolver estes níveis de jogo de forma rápida e sem onerar os prazos do projeto.</p>
<p><b>Produção:</b> Construir todos os níveis, com as mecânicas, scripts de IA, missões, assets, etc.  Criar especificações e protótipos de cobranças, suporte ao cliente e sistema de</p>	<p><b>Produção:</b> Hendrick pontua a necessidade de dedicar esforço para atender às necessidades fora do núcleo do jogo. Esta foi outra sugestão de Hendrick que foi aproveitada pelo JAI.</p>



atualização.	Mais uma vez, a proposta deste trabalho buscou absorver boas reflexões e incrementá-las com detalhamento e direcionamento sobre como proceder durante a execução das sugestões.
<p><b>Beta:</b> Ajustes no gameplay, identificação e correção de bugs e outros refinamentos para alcançar uma versão estável do jogo.</p> <p>Finalizar sistemas como o de cobranças, atendimento a clientes.</p> <p>A abertura do “open beta” marca o início desta etapa.</p>	<p><b>Beta:</b> Muito semelhante ao apresentado por Hendrick, o processo proposto usa a fase beta para correção de bugs, realização de ajustes e refinamentos, além de finalizar os módulos adicionais.</p>
<p><b>Live:</b> O jogo rodando 24x7 a partir do “open beta”. Manutenções semanais, mensais e programas de expansão no jogo.</p>	<p><b>Publicação:</b> Além dos planos de manutenção e expansão, a proposta do JAI visa a documentação e compartilhamento de conhecimento.</p>

Tabela 3: Comparativos entre a proposta de Hendrick e o JAI.

<b>Blitz Game Studio (2020)</b>	<b>Em relação ao JAI</b>
<p><b>Pitch:</b> Fase de negociação com a publisher. É modelagem inicial e conceito do jogo, são discutidos o orçamentos, o cronograma e qualquer outro recurso específico necessário para o projeto.</p> <p>Apenas a alta hierarquia da equipe participa dessa etapa.</p>	<p><b>Iniciação:</b> A Blitz é uma empresa consolidada, com equipes estruturadas e capazes de trabalhar em mais de um projeto.</p> <p>Seu nível de maturidade permite manejar membros da equipe entre projetos e distribuir a gestão sob uma cadeia de comando hierárquica.</p> <p>A Blitz antecipa o cronograma para a fase inicial, pois precisa lidar com isso durante as</p>

	<p>negociações com a publisher.</p> <p>Para adaptar a proposta JAI a uma realidade semelhante, foi incluída a tarefa de criação de uma listagem dos esforços para realização do projeto junto a um cronograma preliminar.</p> <p>E com isso a fase de iniciação passa produzir conhecimento e documentação necessária para montagem de um <i>pitch</i> de apresentação do projeto a um investidor.</p>
<p><b>Pré-produção:</b></p> <p>A modelagem do jogo é cuidadosamente planejada. São definidos pontos importantes como a game engine e outras ferramentas de trabalho para os artistas e designers. Se houver alguma característica nova no projeto, serão criados protótipos para que ninguém na equipe seja surpreendido por algum recurso inesperado.</p>	<p><b>Pré-produção:</b></p> <p>Ambos focam na modelagem do jogo e se atentam em identificar características e ferramentas essenciais para criação do jogo.</p> <p>A prototipagem da Blitz nessa fase é para uso exclusivamente interno e serve para ajudar a própria equipe a entender melhor sobre o projeto. Esta é outra propriedade que foi absorvida pelo JAI e adaptada para atender aos objetivos da proposta.</p>
<p><b>Produção principal:</b> (implementação do conceito do jogo):</p> <p>A cada mês ou 6 semanas a equipe apresenta uma versão do estágio atual do jogo ao publisher. O que será entregue em cada estágio está definido na documentação de game design (são os milestones).</p>	<p><b>Produção:</b></p> <p>O processo proposto define os milestones como “níveis de release”, mas não define quantas semanas aproximadamente serão necessárias para entregá-los, pois esta decisão fica a cargo da equipe envolvida.</p> <p>Como este trabalho propõe testes internos</p>

<p><b>Alpha:</b> (testes internos):</p> <p>Neste ponto, o jogo está jogável, entretanto incompleto. Podem estar faltando texturas, efeitos, áudios, etc.</p>	<p>semanais, as atividades da fase alpha definida pela Blitz Game Studio foram incorporadas na fase de produção.</p>
<p><b>Beta</b> (testadores terceirizados):</p> <p>O jogo está completo e com todos os assets, porém ainda há bugs para serem identificados e corrigidos. Está tudo polido a medida do possível e o processo de playtest continuará até chegar ao ponto de publicação.</p>	<p><b>Beta:</b></p> <p>A fase beta do JAI se comporta como um <i>open beta</i>, quando o jogo é publicado abertamente para testes.</p> <p>Os objetivos são semelhantes ao processo da Blitz Game Studio, porém com adicional da atividade de conclusão dos módulos adicionais.</p>
<p><b>Master</b> (lançamento do jogo):</p> <p>Com o jogo completo e o foco voltado para correção de bugs e polimentos, a equipe é reduzida e alguns membros são remanejados para outros projetos. Normalmente os programadores permanecem do começo ao fim do projeto.</p>	<p><b>Publicação:</b></p> <p>Esta proposta não vislumbra trabalhar em mais de um projeto por vez, nem segmentação da equipe. Em vez disso, é sugerido o compartilhamento de conhecimentos e o planejamento do futuro do jogo.</p>

Tabela 4: Comparativos entre processo da Blitz Game Studio e o JAI

É possível destacar a forte influência dos processos analisados sobre a criação do JAI. Mas para alcançar os objetivos deste trabalho, muitas ideias foram ajustadas e muito conteúdo foi acrescentado para suprir lacunas de incertezas que surgiram durante a sua formação.

Dentre os critérios formulados para orientar a criação do JAI (“Design Participativo e Design Centrado no Usuário”, “Metodologia Ágil”, “Acessibilidade em todo processo”,

“Adequação a equipes pequenas”), o mais difícil de atender foi adequá-lo para pequenas equipes de trabalho, pois foi necessário introduzir certo nível de complexidade em cada fase do processo para garantir a manutenção do foco na criação da acessibilidade e para elevar o processo a um nível de aplicação profissional ao requisitar esforços adicionais nas fases de **iniciação** e **pré-produção**. A inclusão de tal complexidade compilada ao processo o tornou capaz orientar tanto a criação do jogo, como a gestão de todo o projeto. E como alternativa, equipes trabalhando em projetos de baixa complexidade podem fazer uso da sequência simplificada do processo JAI descrita na seção 8 em conjunto com os *boards* de cada fase do processo disponíveis na seção 7.

## 10. Considerações Finais

Existe uma preocupação global quanto a geração de bem-estar e inclusão digital da população idosa. Entretanto, mesmo as maiores lojas virtuais de softwares disponibilizam poucos aplicativos adaptados às necessidades especiais desta grande parcela da população. Durante o levantamento de dados bibliográficos e trabalhos relacionados, foi possível identificar várias iniciativas quanto a propostas de diretrizes pontuais para tornar os jogos acessíveis, mas nenhum modelo que abordasse todo o processo em si, da concepção da ideia à publicação. Portanto, o JAI se vê na responsabilidade de propor soluções técnicas avançadas e documentá-las através dos artefatos gerados no processo, bem como, apresenta a compilação de diretrizes de acessibilidade (descritas no Apêndice B) integradas em cada fase do mesmo, para que sirvam de apoio a futuros trabalhos e estudos que abordem o bem-estar e a inclusão digital da população idosa.

Em questões de desafios, o autor reconhece que pode haver dificuldade na avaliação dos resultados práticos deste trabalho tanto quando aplicado pelos desenvolvedores de jogos, como no cotidiano do usuário final. No que se refere ao uso do jogo acessível pelo idoso, talvez seja necessário a contribuição de geriatras ou psicólogos na fase de mensuração destes resultados, pois a mudança comportamental positiva esperada, como consequência do lazer e da inclusão digital, pode ser atingida em tempos e intensidades diferentes. Por isso, é importante apressar as etapas de prototipação e testes para que os resultados venham a ser colhidos e analisados no mais curto prazo possível.

No que se refere ao uso pelos desenvolvedores, do processo JAI e de seu conjunto de orientações - caracterizadas pelos boards apresentados - os resultados poderão ser validados a

medida que estes incorporarem a proposta no desenvolvimento de seus jogos, possibilitando que a acessibilidade seja intrínseca em todas as etapas do processo. Tais desafios norteiam os trabalhos futuros desta pesquisa.

O processo JAI e as orientações que o acompanham estruturam um arcabouço para a concepção de um modelo de desenvolvimento de jogos acessíveis que contemplará não apenas as etapas de um processo de desenvolvimento, mas também a própria gestão do projeto, integrando as diretrizes de acessibilidades naturalmente ao processo.

Embora o JAI contemple a acessibilidade durante todo o processo de criação do jogo, este trabalho carece por validação em ao menos dois pontos específicos, que não puderam ser realizados devido ao isolamento social imposto pela pandemia do COVID 19 que teve início em março deste ano até a presente data. O primeiro é validá-lo como processo viável, para tal é necessário colocá-lo em prática diante de uma demanda real e analisar os pontos positivos e negativos de trabalhar seguindo as orientações do JAI. O segundo quesito de validação é o resultado do processo. Um jogo produzido com o JAI deve ser avaliado quanto a usabilidade, jogabilidade e acessibilidade para então mensurar a capacidade do processo JAI em guiar o desenvolvimento de um jogo acessível e prazeroso.

Como possibilidade de melhoria deste trabalho é sugerido a criação de uma ferramenta que automatize a construção da documentação inicial do projeto e auxilie na criação do GDD. Tal ferramenta poderia ainda recomendar um conjunto de diretrizes de acessibilidades para um determinado jogo a partir de informações como o público-alvo, dispositivo e o gênero do jogo.

## REFERÊNCIAS

ALEEM, S., Capretz, L. F., & Ahmed, F. **Game development software engineering process life cycle: a systematic review**. Journal of Software Engineering Research and Development, 4(1). doi:10.1186/s40411-016-0032-7. 2016.

Apple Developer. **In-App Purchase**. Disponível em:  
<<https://developer.apple.com/in-app-purchase>>. Acessado em 09 jul. 2020.

Blitz game studio. **Project Lifecycle**. Disponível em:  
<[http://www.blitzgamesstudios.com/blitz\\_academy/game\\_dev](http://www.blitzgamesstudios.com/blitz_academy/game_dev)>. Acessado em: 20 set. 2020.

British Council. **How can technology help us meet the needs of our ageing society?**. Disponível em:  
<<https://www.britishcouncil.org/anyone-anywhere/explore/communities-connections/technology-ageing-society>>. Acessado em: 20 set. 2020.

DUQUE, Ezequiel. et al. **Uma análise sobre o desenvolvimento participativo de jogos educacionais voltados para a terceira idade**. SBIE. 2018.

DUQUE, Ezequiel; ISHITANI, Lucila. **Uma análise da adequação de desafios em jogos móveis considerando o público idoso**. SBGAMES. 2016.

FULLERTON, Tracy. **GAME DESIGN WORKSHOP: A Playcentric Approach to Creating Innovative Games**. CRC Press. 2014.

Ferro, L. S. (2018). **An analysis of players' personality type and preferences for game elements and mechanics**. Entertainment Computing, 27, 73–81.  
doi:10.1016/j.entcom.2018.03.003

GAMBERINI, L; et al. **Cognition, technology and games for the elderly: An introduction to ElderGames Project**. PsychNologyJournal 4. 2006.

Game Accessibility Guidelines. **A straightforward reference for inclusive game design Full List**. Disponível em: <<http://gameaccessibilityguidelines.com/full-list/>>. Acessado em 06 jul. 2020.

Game Accessibility Guidelines. **A straightforward reference for inclusive game design Why and How**. Disponível em: <<http://gameaccessibilityguidelines.com/why-and-how/>>. Acessado em 06 jul. 2020.

Google Play. **Central de políticas do desenvolvedor**. Disponível em:  
<<https://play.google.com/about/developer-content-policy>>. Acessado em 08 jul. 2020.

Google Play. **Configurar um teste aberto, fechado ou interno**. Disponível em:  
<<https://support.google.com/googleplay/android-developer/answer/3131213>>. Acessado em: 21 set 2020.

Google Play. **Lista de verificação do lançamento**. Disponível em: <<https://developer.android.com/distribute/best-practices/launch/launch-checklist>>. Acessado em 08 jul. 2020.

Guru99. **What is Compatibility Testing?: Forward & Backward Testing**. Disponível em: <<https://www.guru99.com/compatibility-testing.html>>. Acessado em: 21 set 2020.

HELPPPI, Ville-Veikko. **The Importance and Difference from App Testing**. Disponível em: <<https://bitbar.com/blog/mobile-game-testing-part-1-the-importance-and-difference-from-app-testing>>. Acessado em: 21 set 2020.

HENDRICK, A. **Project Management for Game Development**. 2014. Disponível em: <<http://mmotidbits.com/2009/06/>>. Acessado 20 set. 2020.

HOLLAND, Taylor Mallory. **4 Healthy Aging Technology Solutions for Seniors**. Samsung Insights. 2018. Disponível em: <<https://insights.samsung.com/2018/02/22/4-healthy-aging-technology-solutions-for-seniors>>. Acessado em: 20 set. 2020.

IBGE. Instituto Brasileiro de Geografia e Estatística. **Pesquisa Nacional por Amostra de Domicílios Contínua**. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/23445-pnad-continua-tic-2017-internet-chega-a-tres-em-cada-quatro-domicilios-do-pais>>. Acessado em: 20 set. 2020.

IJSSELSTEIJN, Wijnand et al. **Digital Game Design for Elderly Users**. 2007. Disponível em: <<http://alexandria.tue.nl/openaccess/Metis215142.pdf>>. Acessado em: 20 set. 2020.

Jam City Inc. **Disney Emoji Blitz**. Disponível em: <[https://play.google.com/store/apps/details?id=com.disney.emojimatch\\_goo&hl=pt\\_BR](https://play.google.com/store/apps/details?id=com.disney.emojimatch_goo&hl=pt_BR)>. Acessado em 08 set. 2020.

KEITH, Clinton. **Agile game development with Scrum**. RR Donnelley. Estados Unidos, Indiana. 2010

KRUCHTEN, P. **The Rational Unified Process: An Introduction**, 2nd edn. Addison Wesley Longman, Reading, 2000.

LAITINEN, Sauli. **Better Games Through Usability Evaluation and Testing**. Disponível em: <[https://www.gamasutra.com/view/feature/130745/better\\_games\\_through\\_usability\\_.php?print=1](https://www.gamasutra.com/view/feature/130745/better_games_through_usability_.php?print=1)>. Acessado em: 21 set 2020.

LANGLEY, Hugh. **Samsung blinded me at CES, then the Relúmino glasses helped me see**. Disponível em: <<https://www.wearable.com/samsung/samsung-relumino-glasses-review>>. Acessado em: 20 set. 2020.

MACHADO, Mônica Consolação & ISHITANI, Lucina. **Heurísticas para avaliação de gameplay direcionadas a idosos**. SBGAMES. 2015. Disponível em: <<http://www.sbgames.org/sbgames2015/anaispdf/artesedesign-full/147318.pdf>>. Acessado em: 20 set. 2020.

MCGRATH, J. **The game development lifecycle**: A theory for the extension of the agile project methodology. 2014. Disponível em: <<https://blog.dopplerinteractive.com/post/112172271166/the-game-development-lifecycle-a-theory-for-the>>. Acessado em 20 set. 2020.

NAP, H.H.; KORT, Y.A.W.; IJSSELSTEIJN, W.A. **Senior gamers**: Preferences, motivations and needs. *Gerontechnology* 2009; 8(4):247-262; doi: 10.4017/gt.2009.08.04.003.00

NETO, Hélio C. Silva; ROQUE, Licínio. **Experiência dos Jogos Digitais aplicados ao Envelhecimento Ativo**. SBGAMES. 2014.

OECD (2020), **Elderly population (indicator)**. doi: 10.1787/8d805ea1-en. Disponível em: <<https://data.oecd.org/pop/elderly-population.htm>>. Acessado em: 24 set 2020.

OMS. Organização Mundial da Saúde. **Ageing and Health**. Disponível em: <<https://www.who.int/news-room/fact-sheets/detail/ageing-and-health>>. 2018. Acessado em: 21 set. 2020.

OMS. Organização Mundial da Saúde. **Ageing and life-course**: What is “Healthy Ageing?”. Disponível em: <<https://www.who.int/ageing/healthy-ageing/en/>>. 2018. Acessado em: 20 set. 2020.

OMS. Organização Mundial da Saúde. **Envelhecimento Ativo**: Uma Política de Saúde. Brasil, Brasília/DF. 2005.

ONU. **World Population Ageing 2019**. Disponível em: <<https://www.un.org/en/development/desa/population/publications/pdf/ageing/WorldPopulationAgeing2019-Highlights.pdf>>. Acessado em: 20 set 2020.

PRESSMAN, Roger S. **Software Engineering**: A practitioner's Approach. 8. ed. Estados Unidos, Nova York: McGraw-Hill Education, 2015.

RAMADAN, R.; Widyani, Y. **Game development life cycle guidelines**. 2013 International Conference on Advanced Computer Science and Information Systems (ICACISIS). doi:10.1109/icacsis.2013.6761558

ROMÃO, José Eduardo, CANELA, Guilherme. ALARCON, Anderson. **Manual da Nova Classificação Indicativa**. Brasília. Ministério da Justiça. Secretaria Nacional de Justiça. Departamento de Justiça, Classificação, Títulos e Qualificação, 2006.

SAKUDA, Luiz Ojima; FORTIM, Ivelise. **II Censo da Indústria Brasileira de Jogos Digitais**. Homo Ludens. 2018.



SCHWABER, Ken; SUTHERLAND, Jeff. Guia do Scrum. **Um guia definitivo para o Scrum**: As regras do Jogo. Versão em português do Brasil. Outubro de 2017. Disponível em: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Portuguese-Brazilian.pdf>. Acessado em: 20 set. 2020.

SEBRAE. **7 passos para analisar o seu mercado**. Disponível em: <https://m.sebrae.com.br/sites/PortalSebrae/artigos/7-passos-para-analisar-o-seu-mercado,602a4d4efe960610VgnVCM1000004c00210aRCRD>>. Acessado em: 07 set. 2020.

SHNEIDERMAN, B.; C. Plaisant. **Designing the User Interface**, 4. ed., Addison-Wesley, 2004.

STEAMWORKS. **Documentação do Steamworks**. Disponível em: <https://partner.steamgames.com/doc/home>>. Acessado em 08 jul. 2020.

STEAMWORKS. **Participando do programa de distribuição Steamworks**. Disponível em: <https://partner.steamgames.com/steamdirect>>. Acessado em 08 jul. 2020.

Software Testing Fundamental. **Compliance Testing**. 2020. Disponível em: <https://softwaretestingfundamentals.com/compliance-testing>>. Acessado em: 20 set 2020.

Software Testing Fundamental. **Performance Testing**. 2020. Disponível em: <https://softwaretestingfundamentals.com/performance-testing/>>. Acessado em: 20 set 2020.

Software Testing Fundamental. **Regression Testing**. 2020. Disponível em: <https://softwaretestingfundamentals.com/regression-testing>>. Acessado em: 20 set 2020.

Software Testing Fundamental. **Smoke Testing**. 2020. Disponível em: <https://softwaretestingfundamentals.com/smoke-testing/>>. Acessado em: 20 set 2020.

Software Testing Fundamental. **Usability Testing**. 2020. Disponível em: <https://softwaretestingfundamentals.com/usability-testing/>>. Acessado em: 20 set 2020.

Software Testing Help. **Ad-Hoc Testing: How To Find Defects Without A Formal Testing Process**. Disponível em: <https://www.softwaretestinghelp.com/ad-hoc-testing>>. Atualizado em: 1 set 2020. Acessado em: 21 set 2020.

Software Testing Help. **What Is Regression Testing?: Definition, Tools, Method, And Example**. Disponível em: <https://www.softwaretestinghelp.com/regression-testing-tools-and-methods/>>. Atualizado em: 1 set 2020. Acessado em: 21 set 2020.

SOMMERVILLE, Ian. **Software Engineering**: 10. ed. Boston: Pearson, 2016.

Steamworks. **Acesso Antecipado**. Disponível: <https://partner.steamgames.com/doc/store/earlyaccess>. Acessado em: 21 set 2020.

Steamworks. **Testes no Steam**. Disponível: <https://partner.steamgames.com/doc/store/testing>>. Acessado em: 21 set 2020.

TAROUCO, Liane Margarida et al. **Jogos Educacionais**. 2014. Disponível em: <<http://www.cinted.ufrgs.br/ciclo3/af/30-jogoseducacionais.pdf>>. Acessado em: 20 set. 2020.

Test Bytes. **9 Different Types of Game Testing Techniques**. 2017. Disponível em: <<https://www.testbytes.net/blog/types-of-game-testing/>>. Acessado em: 21 set 2020.

Test Bytes. **A Guide on Game Testing Methodology**. 2017. Disponível em: <<https://www.testbytes.net/blog/game-testing-methodology/>>. Acessado em: 21 set 2020.

Test Bytes. **Game Testing Tutorial: All The Information is Here!**. 2018. Disponível em: <<https://www.testbytes.net/blog/game-testing-tutorial>>. Acessado em: 21 set 2020.

TONDELLO, G. F., & Nacke, L. E. **Player Characteristics and Video Game Preferences**. Proceedings of the Annual Symposium on Computer-Human Interaction in Play - CHI PLAY '19, 2019.

VAHLO, J., Kaakinen, J. K., Holm, S. K., & Koponen, A. (2017). **Digital Game Dynamics Preferences and Player Types**. Journal of Computer-Mediated Communication, 22(2), 88–103. doi:10.1111/jcc4.12181

W3C. **Web Content Accessibility Guidelines 2.1**. Disponível em: <<https://www.w3.org/TR/WCAG21>>. Acessado em: 20 set. 2020.

# Nome do Jogo

*DOCUMENTO DE PROJETO DO JOGO*

## Público-Alvo:

Definir qual é o perfil de pessoas que o jogo deve agradar.

Exemplo:

Sexo: unisex, maioria feminina.

Idade: 35~70 anos.

Interesses: Jogos casuais, desafios relaxantes, não competitivo, jogos de passatempo.

## Identidade do Jogo:

Definir o jogo em uma frase que guiará as decisões de projeto.

Exemplo:

**Puzzle de montagem (jigsaw) em 3D** que simula a construção de máquinas, móveis, cômodos, ferramentas, veículos, pontes e muito mais.

## Percepção Instantânea do Jogo:

Defina a impressão que o jogo deve transmitir em até 5 palavras ou frases curtas.

Exemplo:

Puzzle relaxante de construção de objetos e montagem de quebra-cabeça 3D.

## Mecânicas Básicas de Gameplay:

Descrever as ações típicas da jogabilidade.

Exemplo:

- Mover objetos livremente em 3D;
- Conectar peças;
- Criar as peças;
- Criar ferramentas usando peças menores;
- Criar objetos inovadores e compartilhar com outras pessoas o desafio de montá-los.

### Enredo e Gameplay:

Breve descrição de como a jogabilidade interage com o enredo do jogo:

Exemplo:

Num mundo completamente destruído, o jogador monta, peça por peça, cada objeto, máquina e instrumento que vai ajudá-lo a continuar sua missão de reconstruir o mundo.

### Recursos Atrativos:

Lista de recursos ou características únicas que você pensa em incluir ao jogo.

Exemplo:

- Relaxante
- Desafiador
- Jogador cria ferramentas que o ajudarão a construir outras máquinas e objetos
- Aprendizado sobre montagem de máquinas e móveis.

### Diferencial:

Lista características que podem fazer o jogo se destacar dentre outros.

Exemplo:

- Mistura o desafio de brincadeira com o aprendizado real de quem está montando objetos reais.
- Potencial ilimitado de montagens e criação de objetos.

### Fator de Replay:

O que o jogo oferece para motivar o jogador a continuar jogando.

Exemplo:

- Poder inventar seus próprios objetos e compartilhar os desafios.
- Recompensas diárias e semanais (skins e novos desafios).
- Ranking de menor tempo de montagem.

### Interface e Controle:

Listar os métodos de entradas do jogador, os controles e como o jogador interage com o jogo.

Exemplo:

- Movimentação e ações usando a tela de toque

### Estilo de Arte:

Descrever a característica gráfica que o jogo deve apresentar. Incluir várias referências com aspecto similar.

Exemplo:

Arte 3D low poly, cores e texturas que representam as características de uma oficina mecânica ou ateliê de artes. Degradê suave nas imagens de background. A cor dos elementos interativos deve ter um contraste que facilite a identificação.

- Figura 1.
- Figura 2.
- Figura 3.
- ...

### Música e Efeitos Sonoros:

Incluir links para músicas e efeitos sonoros similares aos desejados para o jogo. Descrever a sensação que a trilha sonora deve passar ao jogador.

Exemplo:

Os efeitos sonoros dos elementos do jogo devem ter correspondência com o mundo real, mas com sensação mais cômica/cartunesca.

As músicas devem ser suaves e contínuas.

Links:

- Link 1;
- Link 2;
- Link 3;
- ...

### Ferramentas de Trabalho

Periféricos, dispositivos e softwares necessários para produção do jogo e gerenciamento do projeto.

Exemplo:

- Smartphone e tablet para testes;

- Computador e sistema operacional compatível com OpenGL 3 para desenvolvedores;
- Game Engine: Godot;
- Modelagem 3D em Blender;
- Manipulação de imagens com Krita ou similar.

### Plataforma de Publicação e Serviço de Distribuição:

Listar os dispositivos para o qual o jogo será publicado e qual será o meio de distribuição.

Exemplo:

Jogo criado para dispositivos móveis (a princípio, apenas Android) e deve ser distribuído através do Google Play.

### Monetização:

Descrever a estratégia de monetização do jogo.

Ex:

1. Jogo feito sob encomenda. Venda direta.
2. Venda de produtos In-App.

### Marketing:

Estratégias de marketing do jogo.

Exemplo:

- Criar páginas nas redes sociais;
- Participação no evento X e Y;
- Compartilhar conteúdos videolog do desenvolvimento do jogo;
- Criar eventos de para testes do jogo com distribuição de brindes (camiseta, broches, etc)

### Suporte:

Qual será o canal de comunicação com a comunidade para tirar dúvidas, reportar bugs, receber feedbacks?

Exemplo:

- Fórum do site próprio;
- Páginas oficiais nas redes sociais;

- Página do jogo no serviço de distribuição (Google Play)

### Atualização do Jogo:

Como as atualizações do jogo serão enviadas aos jogadores?

Exemplo:

O jogo será atualizado através das ferramentas do serviço de distribuição.

### Metas de Acessibilidade:

#### **Para etapa de protótipo:**

- Orientações básicas de contraste, fonte e métodos de entrada.
- Fornecer controles de volume separados para efeitos, fala e fundo/música.
- Permitir que o jogo possa ser iniciado sem precisar passar por vários níveis de menu.
- Usar um tamanho de fonte padrão de fácil leitura

#### **Para fase de produção:**

- Mecanismos de configuração de cores, contrastes, métodos de entrada e sons.
- Adaptações das legendas: tamanho da fonte e velocidade dos textos configuráveis.
- Adição de “delays” entre a ação do jogador e o efeito. (botões de interface).
- Tutorial intuitivo.
- Dificuldade adaptativa.
- Modo de assistência.
- Permitir que elementos não interativos sejam removidos ou menos evidenciados.
- Salvar perfil de configuração do jogador.
- Permitir que os jogadores progridam através de prompts de texto no seu próprio ritmo.
- Fornecer uma opção para desativar / ocultar todos os elementos não interativos.

#### **Na publicação e marketing:**

- Informar as adaptações de acessibilidades implantadas.

#### **Para o plano de manutenção do jogo:**

- Sistema de tolerância a erros.

### Cronograma de Atividades e Marcos de Desenvolvimento:

Dica: Usar o Trello ou outro aplicativo de gerenciamento de projeto web.

Várias atividades podem ser realizada em paralelo

<b>MARCO/PREVISÃO (tempo total estimado: 18 semanas)</b>	<b>INICIADO</b>	<b>CONCLUÍDO</b>	<b>REVISADO</b>
Mecânicas do jogo (3 semanas)	00/00/0000	00/00/0000	00/00/0000
Assets gráficos (3 semanas)	00/00/0000	00/00/0000	00/00/0000
Assets áudio - compra e adaptação - (1 semanas)	00/00/0000	00/00/0000	00/00/0000
Prototipação com público (até 2 semanas)	00/00/0000	00/00/0000	00/00/0000
Primeiro arco do jogo com montagem de um cenário completo - (4 semanas)	00/00/0000	00/00/0000	00/00/0000
Conta online, vendas In-App, sistema de recompensas, testado na plataforma de distribuição. (2 semana)	00/00/0000	00/00/0000	00/00/0000
Criação do site com material de divulgação e Fórum (3 semanas)	00/00/0000	00/00/0000	00/00/0000
Bateria de Testes Beta e Polimentos (3 semanas)	00/00/0000	00/00/0000	00/00/0000
Publicação da 1º versão oficial na loja (1 semana)	00/00/0000	00/00/0000	00/00/0000

Tabela de cronograma de atividades.



### Estimativa de Orçamento:

Com base no conhecimento produzido até o momento, quanto dinheiro e esforço o projeto requer para ser desenvolvido.

Exemplo:

- 1 programador: R\$9.000,00
- 1 artista: R\$4.500,00
- 1 pacote de efeitos sonoros: R\$100,00
- 1 pacote de músicas: R\$450,00
- Taxas do serviço de distribuição: R\$150,00
- Criação e manutenção do site: R\$600,00
- Despesas não descritivas: R\$400,00
- Despesas com equipamentos: R\$0,00 (já disponíveis)
- Despesas com softwares: R\$0,00 (open source)
- Total estimado: R\$14.600,00

### Backlog de Produto:

Listar todas as atividades necessárias para o desenvolvimento do projeto.

Dica: Usar o Trello ou outro aplicativo de gerenciamento de projeto web. Use o cronograma de atividades macro para organizar o tempo total do projeto e faça cronogramas mensais ou semanais incluindo apenas as atividades e recursos que serão desenvolvidos nesse período.

- Recurso A1;
- Recurso A2;
- Recurso B1;
- Recurso B2;
- Recurso C1;
- Recurso D1;
- Recurso NN;

## **APÊNDICE B <INDICAÇÕES DE DIRETRIZES DE ACESSIBILIDADE PARA IDOSOS>**

### **1. DIRETRIZES PARA FASE DE PROTOTIPAGEM**

As diretrizes para a fase de prototipagem são recomendações genéricas, mas que devem ser levadas em consideração durante toda a criação do jogo. Ao finalizar a fase de pré-produção com um protótipo atendendo a todas essas diretrizes (ou todas que se apliquem ao jogo em questão), os criadores terão uma estrutura sólida de acessibilidade para manter e evoluir durante todo o desenvolvimento do jogo.

#### **1.1 Motor**

- Garantir que os controles sejam o mais simples possível ou fornecer uma alternativa mais simples.
- Garantir que os elementos interativos/controles virtuais sejam grandes e bem espaçados, principalmente em telas pequenas ou sensíveis ao toque.
- Suporte para mais de um dispositivo de entrada.

#### **1.2 Cognitivo**

- Permitir que o jogo possa ser iniciado sem precisar passar por vários níveis de menu.
- Usar um tamanho de fonte padrão de fácil leitura.
- Usar uma linguagem simples e clara.
- Usar formatação de texto simples e clara.
- Evitar imagens tremidas e padrões repetitivos
- Evitar abreviações: requerem uma maior carga do cérebro para processá-las.

#### **1.3 Visão**

- Usar um tamanho de fonte padrão de fácil leitura.
- Usar formatação de texto simples e clara.
- Nenhum texto sobreposto em imagens ou gráficos.
- Fundos simples são melhores para evitar distrações.
- Evitar VR. O uso de óculos de realidade virtual podem causar tontura em pessoas mais velhas.

#### **1.4 Audição**

- Fornecer controles de volume separados para efeitos, fala e fundo/música.

#### **1.5 Fala**

- Garantir que a entrada de fala não seja indispensável e incluída apenas como um método de entrada suplementar/alternativo.

#### **1.6 Geral**

- Garantir que todas as configurações sejam salvas e mantidas.
- Incluir algumas pessoas com deficiências entre os participantes do play-testing.
- Evitar dispositivos de tela pequena, as pessoas mais velhas são os maiores usuários de tablets.
- Idosos têm melhor desempenho usando interfaces de toque (o toque no dedo degenera mais tarde do que em outras habilidades motoras)

### **2. DIRETRIZES PARA FASE DE PRODUÇÃO**

A fase de produção do jogo deve usufruir da estrutura de acessibilidade criada durante a prototipação e evoluí-la de acordo com as metas estabelecidas para o jogo. Isso quer dizer que, apesar da enorme quantidade de recomendações citadas nesta seção, a equipe deve manter o foco em suas metas de acessibilidade e aprimorá-las com as sugestões a seguir. As diretrizes que não forem implementadas a tempo durante a fase de produção (e que fazem parte das metas de acessibilidade) podem ser remanejadas para o plano de manutenção e expansão do jogo.

#### **2.1 Motor**

- Incluir uma opção para ajustar a sensibilidade dos controles.
- Permitir desligar ou calibrar finamente a vibração (quando houver) do controle ou outro dispositivo de contato.
- Evitar entradas repetidas de alta frequência (pressionar o botão rapidamente).
- Se estiver produzindo um jogo para PC, ofereça o modo de janela para compatibilidade com teclados virtuais sobrepostos.

- Evitar ou fornecer alternativas para exigências de manter botões sejam pressionados.
- Permitir que as interfaces sejam reorganizadas.
- Permitir que as interfaces sejam redimensionadas.
- Certifique-se de que várias ações simultâneas (por exemplo, clicar e arrastar) não sejam necessárias e incluídas apenas como método de entrada suplementar ou alternativo.

## 2.2 Cognitivo

- Reconsidere tudo o que você já sabia sobre o design da interface do usuário.
- Os idosos podem não entender coisas como rolagem ou funcionalidade de pesquisa.
- Os idosos podem não reconhecer abreviações e acrônimos comuns (especialmente aqueles com demência).
- Os idosos têm muito mais tempo de atenção do que as pessoas mais jovens; portanto, textos longos e conteúdo profundo são aceitáveis, desde que não precisem dividir sua atenção.
- Incluir tutoriais interativos.
- Permitir que os jogadores progridam através de prompts de texto no seu próprio ritmo.
- Incluir ajuda, orientação e dicas contextuais no jogo.
- Indicar ou permitir lembrete dos objetivos atuais durante o jogo.
- Indicar ou permitir lembrete de controles durante o jogo.
- Inclua um meio de praticar sem falhas, como nível de prática ou modo sandbox.
- Garantir que nenhuma informação essencial seja transmitida apenas por texto ou fala.
- Fornecer opções de cor do texto, ou no mínimo opções de contraste.
- Suporte a bate-papo por voz e texto para jogos multiplayer.
- Garantir que as opções de som / música para cada um dos principais objetos / eventos sejam distintas entre si.
- Destacar palavras importantes.
- Encontrar analogias com processos em um mundo físico usando símbolos que sejam facilmente compreendidos pelo público alvo.
- É altamente recomendável evitar atividades com tempo limitado em aplicativos dedicados a um público mais velho.

- Evitar excesso de animações nos menus e oferecer uma opção de desabilitá-las no menu de acessibilidades.
- Usar títulos claros ou indicadores de navegação para o usuário compreender o contexto da tela com a qual está interagindo.
- Indicadores de interatividade: Adicionar dicas visuais a elementos clicáveis, nos estados ativo e estático.
- A memória de curto prazo e a memória episódica são particularmente vulneráveis; portanto, quando possível, introduza os recursos do jogo gradualmente ao longo do tempo, em vez de todos de uma vez.
- Não dividir as telas em várias ações - um foco de cada vez terá melhores resultados.

### 2.3 Visão

- Garantir que nenhuma informação essencial seja transmitida apenas por uma cor.
- Fornecer alto contraste entre texto / interface do usuário e plano de fundo.
- Permitir desligar ou calibrar finamente a vibração (quando houver) do controle ou outro dispositivo de contato.
- Dar uma indicação clara de que os elementos interativos são interativos.
- Fornecer uma opção para desativar / ocultar a animação de fundo.
- Fornecer uma opção para ajustar o contraste.
- Evitar colocar informações temporárias essenciais fora dos olhos do jogador.
- É recomendável manter uma taxa de contraste de pelo menos 4,5:1, embora seja melhor permanecer acima de 7,0:1.
- Site para teste de contraste:
  - <https://webaim.org/resources/contrastchecker/>
- Site sobre sistema de cores:
  - <https://material.io/design/color/the-color-system.html#color-usage-and-palettes>
- Usar fontes com 16px no mínimo.
- Sans Serifs são bons tipos de letra para usar.
- Evite usar várias fontes e fontes condensadas.
- Use o peso do tipo para criar uma hierarquia clara (negrito > regular).
- Usar texto complementar para informações baseadas em cores (como erros ou interações).

- O texto precisa ser grande o suficiente para facilitar a leitura e alinhado.
- Evitar justificação de texto.
- Usar espaçamento entre linhas de pelo menos 1.5.
- Evite o azul para elementos importantes da interface.
- Não use cores para transmitir uma mensagem.
- Vermelho e verde são as cores mais difíceis de diferenciar para daltonismo.
- Alto contraste é o melhor, especialmente dentro do valor da cor (azul marinho e azul claro ou preto e branco).
- Verificar o jogo usando simuladores on-line de deficiência visual e convertendo projetos em escala de cinza para garantir que eles ainda estejam legíveis.

## **2.4 Audição**

- Garantir que nenhuma informação essencial seja transmitida apenas por sons.
- Manter o ruído de fundo no mínimo durante a fala.
- Fornecer legendas para discursos complementares.
- Suportar para bate-papo por texto e voz para multijogador.

## **2.5 Fala**

- Suportar para bate-papo por texto e voz para multijogador.
- Basear o reconhecimento de fala em palavras individuais de um pequeno vocabulário (por exemplo, "sim", "não", "abra") em vez de frases longas ou palavras com várias sílabas.

## **2.6 Geral**

- Oferecer uma ampla variedade de níveis de dificuldade.
- Permitir que o nível de dificuldade seja alterado durante o jogo, seja através de configurações ou dificuldade adaptativa.
- Oferecer um meio de ignorar elementos de jogabilidade que não fazem parte do mecanismo principal, por meio de configurações ou opção de pular no jogo.
- Incluir modos de assistência, como mira automática e dicas durante o jogo (como no jogo Candy Crush) para evitar que o jogador fique preso em um desafio.
- Fornecer um recurso de salvamento manual.

- Fornecer um recurso de salvamento automático.
- Incluir todas as categorias relevantes de comprometimento (motor, cognitivo etc.) entre os participantes do play-testing, em números representativos com base na idade / demografia do público-alvo.
- Permitir que os usuários ajustem o tamanho da interface, especialmente o texto.
- Dê um feedback claro sobre o progresso e a conclusão
- Fornecer lembretes e alertas como dicas para ações habituais

### **3. DIRETRIZES PARA PUBLICAÇÃO E MARKETING**

Estas são sugestões gerais, mas que são de suma importância para alcançar o público assistido pelas adaptações de acessibilidade implementadas ao jogo.

#### **3.1 Geral**

- Informar as adaptações de acessibilidades implantadas.
- Fornecer detalhes dos recursos de acessibilidade na embalagem e/ou site.
- Fornecer detalhes dos recursos de acessibilidade no jogo.
- Seja explícito: deixe claro o objetivo do produto.

### **4. DIRETRIZES PARA O PLANO DE MANUTENÇÃO DO JOGO**

Além de continuar aprimorando a qualidade da acessibilidade implementada ao jogo, a equipe pode expandir o leque de possibilidades e trabalhar para atualizar o jogo com versões que atendam as necessidades de cada vez mais pessoas.

#### **4.1 Motor**

- Permitir jogar com a tela em modo retrato e paisagem.
- Inclua um delay de 0,5 segundos entre as entradas.
- Fornecer esquemas de controle muito simples, compatíveis com dispositivos de tecnologia assistiva, como interruptores ou rastreamento ocular.

#### **4.2 Cognitivo**

- Fornecer uma opção para desativar sangue e outros efeitos violentos.
- Fornecer narrações pré-gravadas para todo o texto, incluindo menus e instaladores.

- Evitar movimentos ou eventos inesperados repentinos.
- Permitir que todas as narrativas e instruções sejam repetidas.
- Oferecer recurso de bate-papo com símbolos de expressão (emoticons).
- Fornecer uma opção para desativar ou ocultar todos os elementos não interativos do jogo.
- Adicionar textos informativos, além de cores e símbolos, como meios de obter informação em objetos interativos.

#### **4.3 Visão**

- Permitir que o tamanho da fonte seja ajustado.
- Fornecer narrações pré-gravadas para todo o texto, incluindo menus e instaladores.
- Garantir suporte ao leitor de tela, incluindo menus e instaladores (ou oferecer narrações dentro do próprio jogo).
- Usar design de som / música distinto para todos os objetos e eventos.

#### **4.4 Audição**

- Garantir que as legendas sejam reduzidas e que as palavras sejam apresentadas numa frequência (palavras por minuto) apropriadas para a faixa etária alvo.
- Permitir o bate-papo baseado em símbolos (emoticons).
- Use legendas para qualquer voz/vídeo.

#### **4.5 Fala**

- Basear o reconhecimento de fala em atingir um limite de volume (por exemplo, 50%) em vez de palavras.
- Permitir o bate-papo baseado em símbolos (emoticons).

#### **4.6 Geral**

- Permitir que as configurações sejam salvas em perfis diferentes, no nível do jogo ou da plataforma.
- Sistema de tolerância a erros.
- Considerar possibilidade de clicks incorretos e como facilitar o usuário recuperar ou desfazer a jogada.



- Pense em como informações importantes podem ser trazidas à tona novamente, se forem ignoradas muito rapidamente.