

William Luis Alves Ferreira

**Educação e Difusão do Conhecimento em  
Ciências Através de Ferramentas Modernas de  
Aprendizagem: Desenvolvimento de aplicativos  
Interativos**

Brasil

2021, Agosto

William Luis Alves Ferreira

**Educação e Difusão do Conhecimento em Ciências  
Através de Ferramentas Modernas de Aprendizagem:  
Desenvolvimento de aplicativos Interativos**

Relatório final em cumprimento ao Programa  
Unificado de Bolsas de Estudo para Apoio de  
Estudantes de Graduação (PUB-USP) promo-  
vida pela Universidade de São Paulo - USP

Universidade de São Paulo - USP

Instituto de Física de São Carlos – IFSC

Centro de Pesquisa e Inovação em Biodiversidade e Fármacos – CIBFar-CEPID

Espaço Interativo de Ciências – EIC

Brasil

2021, Agosto

# Resumo

Este relatório descreve as atividades realizadas para concluir o projeto de revitalização do jogo Cruzadinha: Jardim Medicinal, parcela do Programa Unificado de Bolsas de Estudo para Apoio de Estudantes de Graduação (PUB-USP) com o título “Educação e Difusão do Conhecimento em Ciências Através de Ferramentas Modernas de Aprendizagem: Desenvolvimento de aplicativos Interativos” vigente em 2020-2021 que, resumidamente, entre outros objetivos, busca desenvolver sistemas interativos para disseminação de ciência sobre os diversos temas na área das ciências biológicas. Esta revitalização faz parte do plano de ação do EIC em atualizar e maximizar o engajamento do repositório de jogos defasados, além de ampliá-lo.

O desenvolvimento deste projeto centrou-se na pesquisa de modelos e diretrizes para desenvolver Sistemas Educacionais Gamificados (GES). E com o planejamento e a implementação alcançamos como resultado a segunda versão do jogo, atualizado e responsivo, assim tornando-o multiplataforma. Além do produto final (o jogo), encontrou-se diretrizes e fluxo de desenvolvimento para otimizar as próximas revitalizações do repositório disponível pelo EIC.

**Palavras-chaves:** [removido para revisão]

# Lista de ilustrações

Figura 1 – Diagrama dos processos de levantamento de requisitos . . . . .	10
Figura 2 – Esquema de levantamento de requisitos . . . . .	11
Figura 3 – Captura da tela inicial do jogo - Versão 1.0 . . . . .	13
Figura 4 – Captura de tela dos principais elementos visuais do jogo - Versão 1.0 . . . . .	14
Figura 5 – Captura de falha na mecânica de avançar automático do jogo - Versão 1.0 . . . . .	15
Figura 6 – Captura de tela das instruções - Versão 1.0 . . . . .	15
Figura 7 – Captura de tela dos indicadores de acerto - Versão 1.0 . . . . .	16
Figura 8 – Captura de tela com falhas de demonstração de status - Versão 1.0 . . . . .	16
Figura 9 – Esquema de geração aleatória de palavras cruzadas NxM . . . . .	18
Figura 10 – Principais quadros ( <i>export Figma</i> ) - Versão Final <i>desktop</i> . . . . .	19
Figura 11 – Principais quadros ( <i>export Figma</i> ) - Versão Final <i>mobile</i> . . . . .	20
Figura 12 – Principais quadros ( <i>export Figma</i> ) - Versões Alternativas <i>mobile</i> . . . . .	21
Figura 13 – Captura de tela <i>desktop</i> Cruzadinha: Jardim Medicinal (Versão 2.0) . . . . .	23
Figura 14 – Captura de tela <i>mobile</i> Cruzadinha: Jardim Medicinal (Versão 2.0) . . . . .	24

# Lista de abreviaturas e siglas

CDCC	Centro de Divulgação Científica e Cultural
DOM	Modelo de Documento por Objetos, em inglês, <i>Document Object Model</i>
EIC	Espaço Interativo de Ciências
JS	Linguagem de programação <i>web</i> - JavaScript
JSON	Acrônimo de JavaScript <i>Object Notation</i>
PBL	Aprendizagem Baseada em Problemas, em inglês, <i>Problem-Based Learning</i>
PUB	Programa Unificado de Bolsas de Estudo para Apoio de Estudantes de Graduação - Universidade de São Paulo

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>6</b>
<b>2</b>	<b>OBJETIVOS</b>	<b>8</b>
<b>3</b>	<b>METODOLOGIA</b>	<b>9</b>
<b>3.1</b>	<b>Estrutura de Produção de <i>Software</i></b>	<b>9</b>
3.1.1	Produção de Requisitos	11
3.1.2	Levantamento	11
3.1.3	Registro	12
<b>3.2</b>	<b>Proposta de Revitalização</b>	<b>12</b>
3.2.1	Levantamento de requisitos	12
<b>4</b>	<b>RESULTADOS</b>	<b>13</b>
<b>4.1</b>	<b>Projeto do Jogo: Proposta de Revitalização</b>	<b>13</b>
4.1.1	Requisitos Estéticos	13
4.1.1.1	Conceito Artístico	14
4.1.1.2	Mecanismos	15
4.1.1.3	Progressão de Dificuldade	16
4.1.1.4	Interface do Usuário - UI	17
4.1.2	Requisitos Funcionais	18
<b>4.2</b>	<b>Protótipo de <i>Layout</i></b>	<b>19</b>
<b>4.3</b>	<b>Versão Final</b>	<b>21</b>
4.3.1	Layout Final	21
4.3.2	Descrição Complementar: Novidades da versão	21
<b>5</b>	<b>CONCLUSÃO</b>	<b>25</b>
<b>5.1</b>	<b>Comentários Complementares</b>	<b>25</b>
<b>5.2</b>	<b>Implementação Futura</b>	<b>25</b>
	<b>REFERÊNCIAS</b>	<b>26</b>

# 1 Introdução

Aprendizado criativo contempla a exploração didática das questões do cotidiano de modo a destrinchar problemas reais, agregando, então, experiência do mundo real aos estudantes. Em contraponto, o tratamento isolado dos assuntos didáticos apresentados no ensino regular, principalmente, através de aulas expositivas, não capacita os estudantes a utilizar da interdisciplinaridade para resolução de problemas como na educação criativa. Esta abordagem é amplamente disseminada em países como Portugal, Inglaterra e Holanda, com o cerne dessa metodologia de Aprendizagem Baseada em Problemas (i.e PBL, em inglês, Problem-Based Learning) como discutido por (CARDOSO, 2011).

Gamificação (ou, em inglês, *gamification*) tornou-se uma alternativa e complemento às aulas expositivas, o que significa utilizar elementos dos jogos de forma a engajar pessoas para atingir um objetivo. Na educação, como proposto por (KIRYAKOVA; ANGELOVA; YORDANOVA, 2014) o potencial da gamificação é imenso: ela funciona para despertar interesse, aumentar a participação, desenvolver criatividade e autonomia, promover diálogo e resoluções a situações-problema, tornando-se uma alternativa e complemento às aulas expositivas.

As diversas abordagens alternativas ao tradicional modo de educar, vêm crescendo com advento de novas tecnologias e formatos de mídias como proposto por (LITTO, 1996). Tratar o aprendizado como uma evolução contínua ao invés de módulos educacionais de conteúdo memorizados - fornecido pelo meio tradicional de aulas expositivas - demonstram ganhos significativos na consolidação do aprendizado a longo prazo, além de desenvolver a organização mental de informação pelo sistema cognitivo rápido (KAHNEMAN, 2011), associado muitas vezes de forma simplificada a intuição.

Este projeto trata-se da continuidade do desenvolvimento do plano de revitalização do acervo de jogos da modalidade aplicação *web* do EIC (2021), especificamente, este documento contempla a revitalização do jogo Cruzadinha: Jardim Medicinal EIC (2021) já existente no repertório do Espaço Interativo de Ciências (EIC). Devido a datação da linguagem de programação e *Game Design*, fez-se necessário aplicar o processo de modernização, além de ampliar o engajamento tornando-o *cross-platform* ao reestruturá-lo como uma aplicação *web* responsiva.

Este relatório contém as principais características sobre a produção do jogo em questão, abordando, em especial, os diversos aspectos que envolvem o planejamento, desenvolvimento e problemáticas educacionais envolvidas na gamificação da educação, em especial, estendendo as atividades desempenhadas e registradas no ***Relatório final em cumprimento ao Programa Unificado de Bolsas de Estudo para Apoio***

***de Estudantes de Graduação (PUB-USP) Edição 2019-2020*** . Sobre o caráter técnico, temos o uso de *frameworks* dinâmicas focadas em aplicações *single-view* possibilitando o desenvolvimento de sistemas com alto grau de interatividade, além da seleção da linguagem de programação *web* JavaScript, bibliotecas (REACTJS) e demais *frameworks* com o paradigma de componente para sistemas *Web*.



## 2 Objetivos

Sobre a instituição Espaço Interativo de Ciências (EIC), esta possui o objetivo de implementar o desenvolvimento de mídias interativas como novas ferramentas de aprendizagem para a difusão do conhecimento em ciências com foco na descoberta de novos medicamentos e produtos naturais dos ecossistemas de nosso Estado (Mata Atlântica, Cerrado e Mangue). Com isso, todos os projetos desenvolvidos pela instituição possuem as mesmas diretrizes alinhadas ao princípio de disseminar ciência sobre os diversos temas na área das ciências biológicas.

Este relatório contempla as pesquisas de ferramentas e método para revitalização do jogo interativo Cruzadinha: Jardim Medicinal [EIC \(2021\)](#), a respeito dos tópicos: contexto educacional, *game design*, performance e distribuição. Com isso, busca-se analisar esses tópicos a fim de modernizar a interface do usuário revitalizando o jogo já existente.

Um dos desafios mais importantes no design de jogos (especialmente ao projetar jogos educativos) é identificar a experiência de usabilidade do aluno, pois norteará as decisões do projeto, tanto em aspectos funcionais, quanto estéticos, maximizando, desta forma, o engajamento dos usuários finais (estudantes) com o jogo. As demais seções deste relatório descrevem a análise e desenvolvimento focando em maximizar o engajamento do jogo, assim como, transmitir as intenções didáticas atreladas a ele.

## 3 Metodologia

Neste tópico, serão apresentados o estudo e a revisão bibliográfica a cerca da produção e do desenvolvimento de aplicações. Para isso, fez-se uso do campo de estudo “Engenharia de *Software*”, interessado no gerenciamento de projeto, organização e produtividade. Resumidamente, todo o projeto foi baseado em **proposta** e **requisito**, mais especificamente, através da revisão bibliográfica, norteou-se a análise de defasagem técnica proporcionando a proposta de transmutação da versão 1.0 para 2.0, que por sua vez conduziu o desenvolvimento do jogo (objeto deste relatório), ou seja, a proposta de revitalização concedeu o modo de explorar os requisitos a fim de modernizar o jogo, Cruzadinha: Jardim Medicinal.

Segue, nesta seção, um breve resumo sobre engenharia de *software* (seção 3.1), e, em seguida, define-se o escopo do projeto (Proposta de revitalização, seção 3.2) com os tópicos a serem analisados.

### 3.1 Estrutura de Produção de *Software*

Engenharia de software é uma das área da computação voltada à especificação, desenvolvimento, manutenção e criação de software; com a aplicação de tecnologias e práticas de gerenciamento de projetos, visando organização, produtividade e qualidade. Atualmente, essas tecnologias e práticas englobam linguagens de programação, banco de dados, ferramentas, plataformas, bibliotecas, padrões de projeto de software, processamento e avaliação de qualidade como proposto por [Pressman e Maxim \(2016\)](#).

Para o projeto descrito neste relatório, as ferramentas de engenharia de software cumprem um papel fundamental devido a capacidade de maximizar a qualidade do projeto já existente a fim de revitaliza-lo.

Desta forma, os tópicos sumarizam os conceitos aplicados neste paradigma de concepção de projeto.

- a) Sommerville ([COSER; CARVALHO; KOVALESKI, 2006](#)) Estrutura básica para levantamento de requisitos de um sistema.
- b) Compreensão do domínio: Os desenvolvedores de aplicações devem elaborar sua compreensão do domínio do problema, seja através de reunião com especialistas ou estudo da área;
- c) Coleta de requisitos: É o processo de interagir com os *stakeholders*<sup>1</sup> do sistema

---

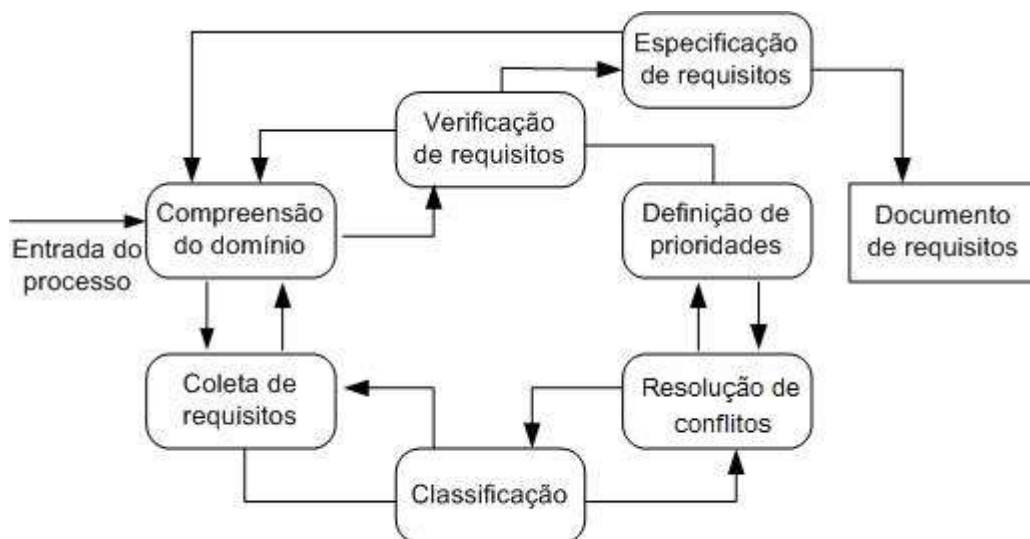
<sup>1</sup> stakeholders - O mesmo que: *view*, traduz-se como **hierarquia de visões**, comum a gerenciamento de banco de dados.

para descobrir seus requisitos. A compreensão do domínio se desenvolve mais durante essa atividade;

- d) Classificação: Essa atividade considera o conjunto não estruturado dos requisitos e os organiza em grupos coerentes;
- e) Resolução de conflitos: Quando múltiplos *stakeholders* estão envolvidos, os requisitos apresentaram conflitos. Essa atividade tem por objetivo solucionar esses conflitos;
- f) Definição das prioridades: Em qualquer conjunto de requisitos, alguns serão mais importantes do que outros. Esse estágio envolve interação com os *stakeholders* para a definição dos requisitos mais importantes;
- g) Verificação de requisitos: Os requisitos são verificados para descobrir se estão completos e consistentes e se estão em concordância com o que os *stakeholders* desejam do sistema.

A primeira etapa da análise de projeto é identificar os possíveis pontos de vista. Nessa etapa, os desenvolvedores se reúnem com os *stakeholders* e utilizam a abordagem de *brainstorming* para identificar os serviços em potencial e as entidades que interagem com o sistema.

Figura 1 – Diagrama dos processos de levantamento de requisitos

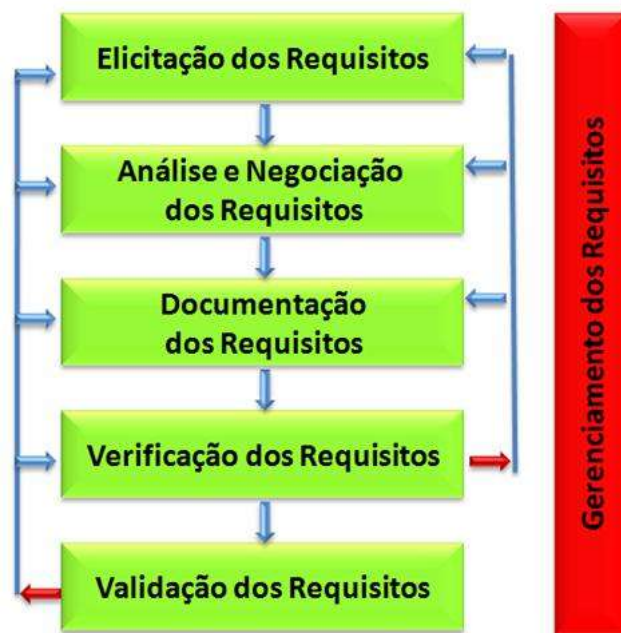


Fonte: [MORAES \(2020\)](#)

Pode-se definir uma estrutura básica para formular o desenvolvimento de aplicações, tanto sobre *game design* (estrutura e desenvolvimento do jogo em termos de engajamento e diversão), quando sistemas de *software* (requisitos funcionais).

Com o esquema presente na Figura 2, seguem os requisitos necessários para desenvolver o projeto de revitalização.

Figura 2 – Esquema de levantamento de requisitos



Fonte: Spínola (2020)

Conteúdo complementar deste tópico pode ser encontrado no artigo Engenharia de Software - Introdução, escrito por Spínola (2020) para o site DevMidia.

### 3.1.1 Produção de Requisitos

A cada fase do ciclo de vida do software, produzimos um documento contendo uma representação distinta do software a ser construído. Cada um desses documentos representa o software em um determinado nível de abstração. A tendência é diminuirmos o nível de abstração através da inclusão de mais e mais detalhes até que, sua última representação, seja o código fonte na linguagem escolhida.

### 3.1.2 Levantamento

Esta atividade relaciona-se à obtenção dos requisitos do software. Para isto, desenvolvedores e engenheiros de software trabalham com clientes e usuários finais para descobrir o problema a ser resolvido, os serviços do sistema, o desempenho necessário, restrições de hardware e outras informações.

### 3.1.3 Registro

Uma vez identificados e negociados, os requisitos devem ser documentados para que possam servir de base para o restante do processo de desenvolvimento.

## 3.2 Proposta de Revitalização

O ponto de partida para a revitalização é realizar a análise da atual versão do jogo, executando, portanto, as seguintes etapas em cada aspecto que contemple o sistema:

- a) Analisar defasagem do sistema;
- b) Propor mudança;
- c) Requisitos Levantados.

Desta forma, nas próximas subseções descreve-se com detalhes as etapas a acima, e , com isso, possibilitar-se-á a análise das principais características da versão atual do jogo, principalmente suas limitações.

### 3.2.1 Levantamento de requisitos

Com base na revisão bibliográfica apresentada na seção 3.1.2, buscou-se proporcionar os requisitos técnicos e funcionais mediante as necessidades de aprimoramento, além das características presentes no projeto em questão.

Sobre requisitos de ferramenta de produção, temos os tópicos:

- a) Linguagens de programação - JavaScript;
- b) *Frameworks*/Bibliotecas - ReactJS e Bootstrap;
- c) Serviços de homologação - Configuração do servidor de hospedagem;
- d) Prototipação - Figma.

Com relação ao *Game Design*, os estudos de referência necessários são:

- a) *Layout* de telas;
- b) *Design* de elementos visuais;
- c) Identidade visual;
- d) Experiência em *Game* (Progressão).

Apresenta-se, nesta seção (3.2), um modelo das principais etapas necessárias para execução do projeto descrito neste documento.

## 4 Resultados

Para apresentação deste capítulo é necessário distinguir os resultados em três seções; a primeira, seção 4.1, apresenta o projeto resultante do modelo definido na seção 3.2; e posteriormente, na seção 4.2, apresenta-se a versão intermediária do jogo, em que descreve-se os principais elementos já definidos antes do *Layout* final, e, por fim, na seção 4.3 apresenta-se o jogo Cruzadinha: Jardim Medicinal (Versão 2.0) que estará disponível na íntegra no site do EIC na segunda quinzena de agosto de 2021.

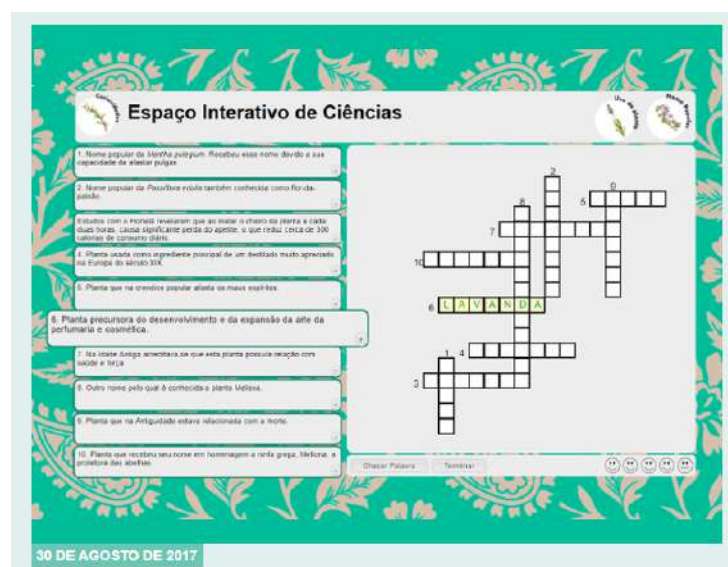
### 4.1 Projeto do Jogo: Proposta de Revitalização

Nesta seção, serão apresentados os resultados em relação ao projeto de software, como: levantamento de requisitos, *design* de *layout* de baixa complexidade e definição dos requisitos funcionais e seus algoritmos.

#### 4.1.1 Requisitos Estéticos

Analisando a defasagem do jogo atual, realizam-se as **propostas** de mudanças correspondentes aos subtópicos listados abaixo, obtendo como resultado o levantamento de **requisitos**.

Figura 3 – Captura da tela inicial do jogo - Versão 1.0



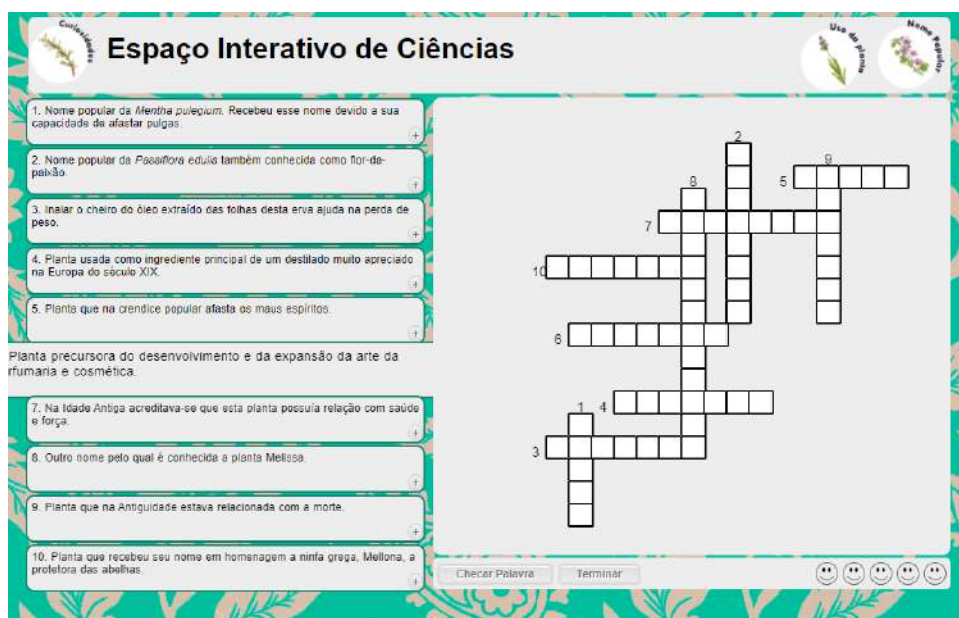
Fonte: EIC (2021)

Ressaltando que esse projeto trata-se da revitalização do jogo já existente no repertório do EIC, Cruzadinha: Jardim Medicinal [EIC \(2021\)](#); foram avaliados os aspectos dispostos nas subsecções: Conceito artístico, Mecanismos, Progressão e Interface.

#### 4.1.1.1 Conceito Artístico

Características como paleta de cor, layout e os componentes de ornamento dos indicadores caracterizam os marcadores que fatalmente deixam datadas as questões estéticas do jogo, isso se deve ao uso do ‘Adobe Flash’ [Adobe \(2020\)](#) como ferramenta de construção do jogo na versão 1.0. Outro elemento visual incoerente é o dimensionamento da fonte e sobre posição das dicas.

Figura 4 – Captura de tela dos principais elementos visuais do jogo - Versão 1.0



Fonte: [EIC \(2021\)](#)

Proposta: Ferramentas como o Flash trabalham com elementos estruturados e estáticos em tempo de execução, o que acarreta na incapacidade de manutenção e evolução de qualquer sistema. Em relação ao *layout* dos elementos visuais, redimensionar fontes (texto) e redesenhar a distribuição de informação através de *switchs* para aproveitar o espaço disponível em múltiplas telas mantendo a coerência entre elas.

Requisitos: Linguagem e ferramenta de produção que possibilitem a renderização em tempo de execução com uma distinção de conteúdo e *assets*, possibilitando, desta forma, a manutenção e atualização da estética.

Solução Adotada: Utilizar paradigmas para desenvolvimento responsivo através de *frameworks single-view* deve ser o suficiente para um processamento de *assets* em tempo de

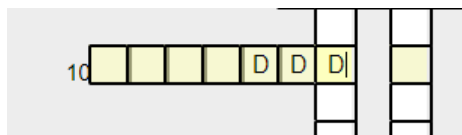


execução, por exemplo, aplicando Angular2, ReactJs, ou, até mesmo, apenas desenvolver em JavaScript. Como nenhum trecho de código será reaproveitado, um protótipo de *layout* será definido para acrescentar identidade visual no jogo, já que é inexistente na versão em análise (1.0) qualquer paralelo claro ao EIC e seus patrocinadores.

#### 4.1.1.2 Mecanismos

A aba “instruções” e o consagrado histórico do Cruzadinhas “O Coquetel” [Coquetel \(2021\)](#) não deixa ambiguidade nas mecânicas presentes no jogo, porém, durante a execução, é perceptível a ausência de um indicativo de acerto parcial distinto da conclusão do jogo - significando que, ao acertar uma das palavras do conjunto disponível, uma resposta visual deve ser emitida (circundar as letras ou talvez um contador das palavras) - e uma mecânica de avanço automático entre as lacunas pode melhorar a experiência do usuário.

Figura 5 – Captura de falha na mecânica de avançar automático do jogo - Versão 1.0



Fonte: [EIC \(2021\)](#)

Figura 6 – Captura de tela das instruções - Versão 1.0



Fonte: [EIC \(2021\)](#)

Proposta: Buscar *framework*/biblioteca ou desenvolver um *script* que possibilite o avanço automático entre as lacunas, além de acrescentar resposta visual/sonora dos acertos parciais mais intuitiva que os indicadores presentes.

Requisitos: Uma ferramenta que permita um controle refinado de eventos, em particular da seleção das letras do tabuleiro, será necessário para implementar a mecânica



Figura 7 – Captura de tela dos indicadores de acerto - Versão 1.0



Fonte: [EIC \(2021\)](#)

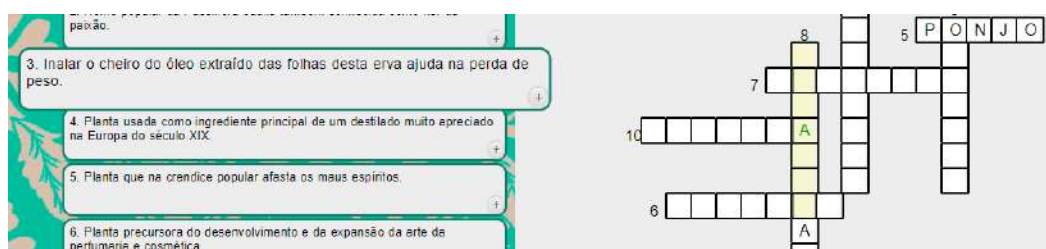
de avanço automático entre lacunas, através de *framework* orientada a componentes e suas posições, juntamente, a componentes com controle de estados e resposta bem definida a eventos.

Solução Adotada: Com a seleção de bibliotecas como ReactJS (paradigma orientado a componentes) possui-se o processamento e controle mais amplo de eventos em comparado a outros paradigmas como DOM pré-processado, isto é, podemos gerenciar os eventos de acerto parcial e finalização do jogo. Também, faz-se necessário a busca por biblioteca ou desenvolvimento de *script* que possibilite a mecânica de avanço automático de lacunas integrada ao ReactJS.

#### 4.1.1.3 Progressão de Dificuldade

A progressão do desafio, em uma primeira impressão, está desbalanceada, já as dicas estão equilibradas na perspectiva da proposta didática do jogo, apresentando os sinônimos e descrição dos tópicos envolvidos. A mecânica de acertos parciais não é evidente durante o jogo, assim como, não é notório as palavras e suas associadas dicas em relação ao *status* de acerto; com isso, a progressão do jogo como um todo é falha por não proporcionar escalabilidade e adaptabilidade para usuário. Mas, vale a ressalva que devido o histórico do Cruzadinha disponível em revistas como “O Coquetel” [Coquetel \(2021\)](#) não agregam competitividade, como tempo de resolução ou pontuação de qualquer tipo, por esse motivo, progressão de dificuldade neste jogo perde o sentido.

Figura 8 – Captura de tela com falhas de demonstração de status - Versão 1.0



Fonte: [EIC \(2021\)](#)

Proposta: Considerando a natureza do cruzadinhas comercial, tanto um sistema

de pontuação, quanto estabelecer um cronômetro não aplica-se na dinâmica de tornar o jogo fluído. Portanto, faz-se necessário estabelecer a o gerenciamento de dicas, ou seja, ofertar uma quantia limitada de dicas para que o usuário tenha autonomia de controlar a dificuldade do jogo, além de tornar randômico a jogabilidade conciliado a geração de tabuleiros aleatórios, sobre o último será explorado ao longo do texto.

Requisitos: Elementos visuais dinâmicos em tempo de execução renderizados conforme as ações durante o jogo.

Solução Adotada: Assim como na subseção anterior, as bibliotecas com paradigmas orientados a componentes possuem o tratamento mais amplo de eventos em comparado a outros paradigmas, desta forma, as respostas do jogo ao jogador são mais complexas e diversas comparada à ferramenta Flash. Contudo, limitando ao uso de ReactJs o uso de telas complementares anterior a tela principal jogo podem cumprir o papel de introduzir os jogadores as mecânicas básicas do jogo e seus objetivos, além de possibilitar o gerenciamento de dicas, mecânica esta capaz de tornar a progressão do jogo mais fluído e divertido.

#### 4.1.1.4 Interface do Usuário - UI

O jogo a ser revitalizado possui um único *layout* - o tabuleiro do Cruzadinha: Jardim Medicinal, as dicas e os indicadores de acertos - atende as necessidades básicas do jogo, porém este *layout* não situa o progresso do jogador ou uma integração clara com o Facebook. Uma alternativa seria apresentar uma classificação dos demais jogadores. A experiência em jogo não deve interferir na dinâmica base, encontrar as 10 palavras, mas desfavorece o recurso de integração com o Facebook.

Proposta: A Interface do Usuário (UI) deve ser aprimorada para maximizar a leitura de informação sobre as palavras que compõem o jogo, simultaneamente, mantendo o jogador ciente do progresso em tempo de execução, e, além disso, introduzi-lo de forma natural e fluida a iniciar o jogo seja por “fase introdutória” ou instruções textuais mais claras semanticamente e apresentadas de forma adequada.

Requisitos: Uma linguagem que possua paradigma de estrutura de elementos padronizada e limpa deve ser o suficiente para proporcionar uma boa experiência de usabilidade.

Solução Adotada: A princípio, tomaremos o tratamento de UI baseado na bibliotecas/*frameworks* de *front-end* **Bootstrap 5 team** (2021), devido à baixa complexidade das telas e volume limitado, o que permite mais personalização do *Design Layout*, alinhado à subseção 4.1.1.1, e, desta forma, exigir-se-á mais controle de padrão/coerência do desenvolvedor.

### 4.1.2 Requisitos Funcionais

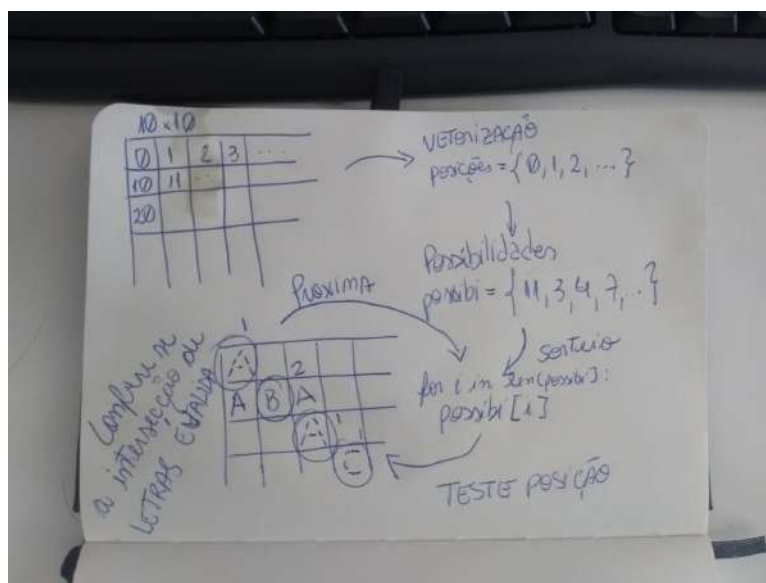
Com a análise realizada anteriormente em Requisitos Estéticos (subseção 4.1.1), extraímos os tópicos e diretrizes para esboçar o projeto, e, deste ponto, planejar o desenvolvimento, andamento e produção da revitalização em termos funcionais.

A seguir, será mostrada a lista de requisitos funcionais levantados:

- a) **Matriz de células dinâmica** - gerar matriz NxM, viabilizando-se:
  - Escalabilidade de dificuldade;
  - Adequação do encaixe das palavras;
  - Referenciar as células para distribuição automática;
- b) **JSON** - Para controle e gerenciamento do conteúdo textual do jogo.
- c) **Distribuição dinâmica das palavras** - mantém-se as disposições possíveis das palavras (vertical e horizontal), assim como, o efeito de cruzamento das palavras e reconhecimento dos espaços nulos, a fim de posicionar de forma automática as palavras, no qual foram definidos o tamanho da matriz que compõem o tabuleiro do jogo e o banco de palavras a serem posicionadas.

A principal funcionalidade do jogo é a característica de gerar automaticamente palavras cruzadas aleatórias e para qualquer matriz NxM. Este componente é definido no arquivo **projeto/src/Componentes/Construtor.js** e foi concebido segundo o rascunho apresentado na figura 9.

Figura 9 – Esquema de geração aleatória de palavras cruzadas NxM



Fonte: Próprio autor

## 4.2 Protótipo de *Layout*

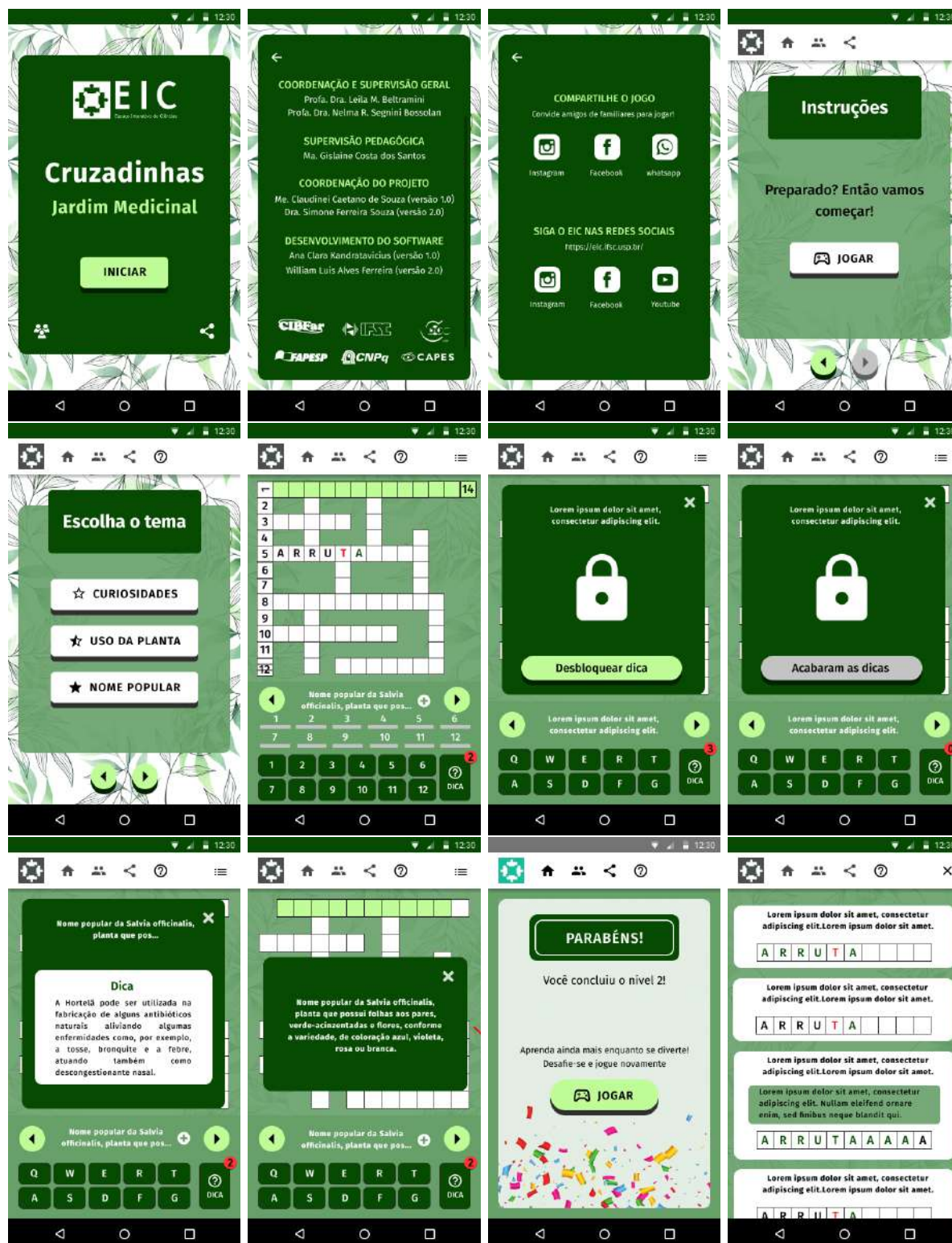
Utilizando a ferramenta *Figma* [Figma \(2021\)](#), concebeu-se os protótipos que nortearam as telas produzidas em *React Js* e *CSS* para estilização, e, desta forma, utilizando os requisitos levantados na seção 4.1 foram desenvolvidos os protótipos presentes nas Figuras 10 e 11.

Figura 10 – Principais quadros (*export Figma*) - Versão Final *desktop*



Fonte: Próprios autores



Figura 11 – Principais quadros (*export Figma*) - Versão Final *mobile*

Fonte: Próprios autores

Ressalva-se o uso do figma para destrinchar alternativas tanto de layout como de funcionalidades para teste de conceito da proposta 3.2, como visto na figura 12.

Figura 12 – Principais quadros (*export Figma*) - Versões Alternativas *mobile*

Fonte: Próprios autores

## 4.3 Versão Final

Por fim, a versão final da revitalização do jogo, Cruzadinha: Jardim Medicinal (Versão 2), estará disponível para o público final no site do EIC na segunda quinzena de agosto de 2020. A fim de serem executados testes de qualidade, o código foi disponibilizado para o acesso da equipe no link [eic.ifsc.usp.br/app/cruzadinhas](http://eic.ifsc.usp.br/app/cruzadinhas). Segue nas figuras 13 e 14 as capturas de tela desta versão (2.0).

### 4.3.1 Layout Final

O *layout* desenvolvido pela consultora técnica de informação Dra. Simone Ferreira Souza juntamente com o bolsista autor deste documento são apresentados na figura 10 e 11 representa a revisão e aprimoramento apostos explorar as alternativas presente na figura 12.

### 4.3.2 Descrição Complementar: Novidades da versão

Este projeto foi desenvolvido em Javascript utilizando a biblioteca *open-source* ReactJS, concebido e fomentado pelo Facebook. Para *front-end* foi desenvolvido com base no *Layout* Final a folha de estilo (CSS) diretamente estruturada em flex-box com uso da *framework*/biblioteca **Bootstrap 5** simplificando a generalização dos assets.

A seguir todas as funcionalidades e diretrizes implementadas foram:

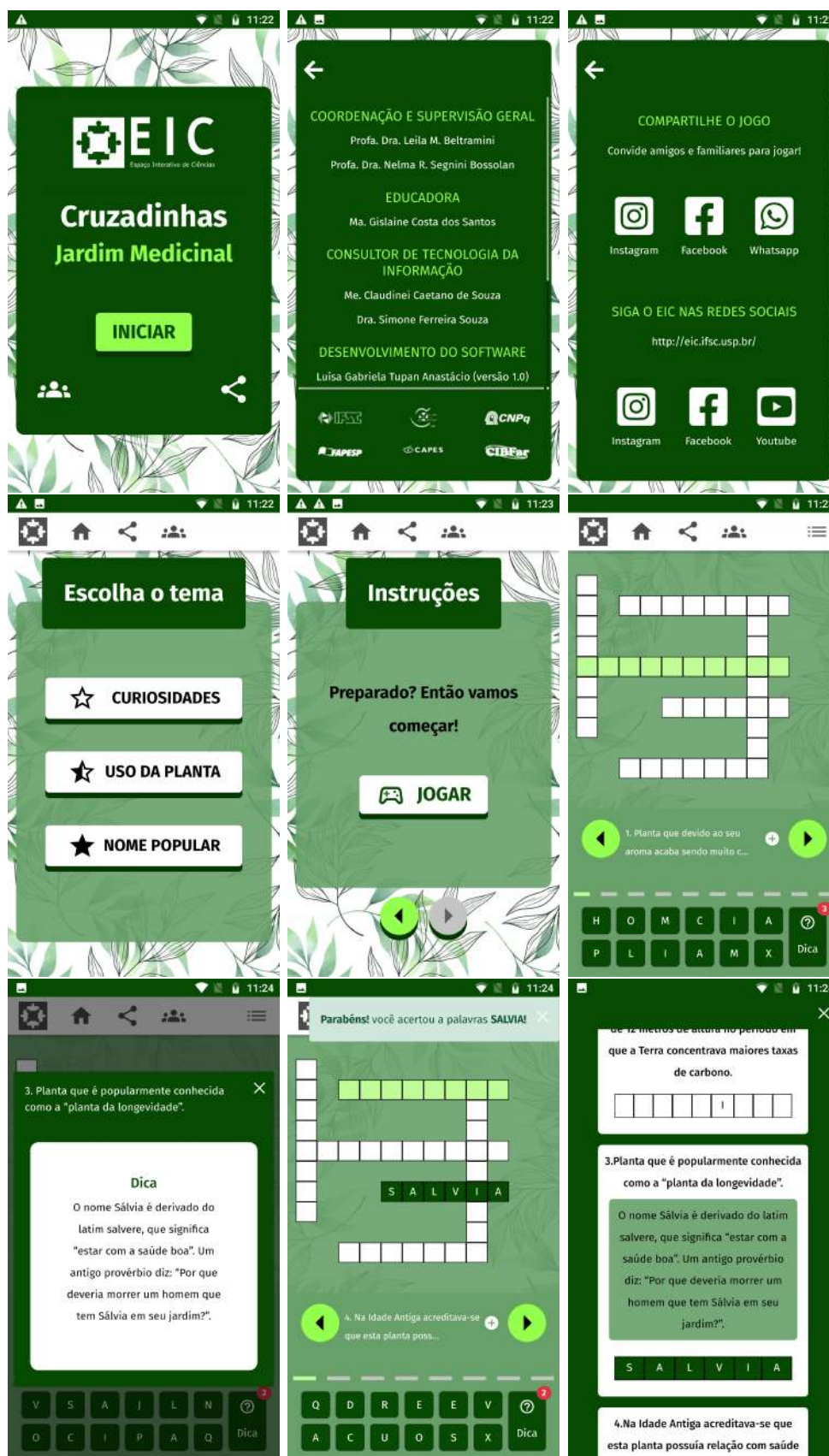
- Distribuição aleatória de palavras a qualquer dimensão do tabuleiro de letras;
- Temas distintos;

- c) Roteamento de telas;
- d) Estrutura de balanceamento, ou seja, flexibilidade em configurar o tabuleiro, por exemplo, definir conjunto de X palavras, Y linhas e W colunas, para qualquer X, Y e W;
- e) Múltiplas visões do conteúdo do jogo (tabuleiro , dicas e lista de palavras)
- f) Estrutura de *Layout Front-end Flexbox-CSS Bootstrap 5*;
- g) Tela de finalização do jogo;
- h) Compartilhamento nas redes sociais;
- i) Apresentação adequada dos patrocinadores e participantes no desenvolvimento do projeto;
- j) Identidade Visual alinhada as diretrizes do EIC.







Figura 14 – Captura de tela *mobile* Cruzadinha: Jardim Medicinal (Versão 2.0)

Fonte: Próprios autores

## 5 Conclusão

As pesquisas e estudos realizadas sobre os tópicos: organização de projeto (Engenharia de *Software*), tecnologias (bibliotecas e *frameworks* JavaScript) e *Game Design* desempenharam suas funções como esperado, ou seja, o conteúdo disposto na seção 3 proporcionou os fundamentos que nortearam a execução do projeto, além de mantê-lo consistente e generalista.

Ressalva-se a capacidade de explorar o jogo desenvolvido como base para diversos temas, não limitando-se ao implementado neste projeto, ou seja, a estrutura básica do jogo é palavras cruzadas genérico, logo sua aplicação não se restringe ao tema Jardim Medicinal, podendo ser redesenhado para outros temas frequentemente alinhados aos objetivos do EIC, através de alterações em *assets*, banco de palavras/dicas (JSON) e na folha de estilo intitulada no projeto como “diretrizes”.

### 5.1 Comentários Complementares

O entendimento sobre os aspectos didáticos incluídos no jogo ainda faz-se necessário a revisão dos conceitos a cerca das dicas e suas palavras, além do aprimoramento do engajamento de novos usuários, sendo este, feito através de aspectos visuais atraentes e agradáveis, isto é, obter o *Layout* necessário para maximizar o aproveitamento do jogo ao longo do tempo, pois devido ao caráter dinâmico proporcionado pela programação orientada a componentes, e , além da fácil modularidade de *assets*, é viável a manutenção e modernização sem que se percam os algoritmos já desenvolvidos.

### 5.2 Implementação Futura

A integração com os micro-serviços do Facebook se limitou a continuar o compartilhamento de placar já possibilitado na Versão 1 do jogo, porém, existe a oportunidade de aprofundar a integração ao proporcionar classificação de jogadores, ou disputas jogador contra jogador sobre a mesma configuração de tabuleiro, e, conseqüentemente, tal integração agregará mais engajamento orgânico ao jogo, desta forma, concedendo-lhe sobrevida.

Na última semana de Agosto de 2021 temos um conjunto de palavras sobre doenças relacionadas ao vetor **Aedes aegypti**, este conjunto é candidato para utilização do *template*, no qual requer estudo de paleta de cores e *assets* para concluir a implementação.

# Referências

- ADOBE, G. *Adobe Flash Player*. 2020. Adobe Flash Player. Disponível em: <<https://www.adobe.com/products/flashplayer.html>>. Acesso em: 09 jan 2020. Citado na página 14.
- CARDOSO, I. D. M. Métodos ativos de aprendizagem: o uso do aprendizado baseado em problemas no ensino de logística e transportes. 2011. Citado na página 6.
- COQUETEL, O. *O Coquetel*. 2021. O Coquetel. Disponível em: <<http://coquetel.com.br/>>. Acesso em: 09 jan 2020. Citado 2 vezes nas páginas 15 e 16.
- COSER, M. A.; CARVALHO, H. G. de; KOVALESKI, J. L. A gestão do conhecimento no apoio à gestão de requisitos em software. *XIII SIMPEP-Bauru, SP*, 2006. Citado na página 9.
- EIC. *Caça Palavras: Doenças Tropicais*. 2021. EIC. Disponível em: <<http://eic.ifsc.usp.br/caca-palavra/>>. Acesso em: 09 jan 2021. Citado 6 vezes nas páginas 6, 8, 13, 14, 15 e 16.
- FIGMA, G. *Figma*. 2021. Figma. Disponível em: <<https://www.figma.com/about/>>. Acesso em: 09 jun 2020. Citado na página 19.
- KAHNEMAN, D. *Thinking, Fast and Slow*. [S.l.]: Saraiva, 2011. Citado na página 6.
- KIRYAKOVA, G.; ANGELOVA, N.; YORDANOVA, L. Gamification in education. In: PROCEEDINGS OF 9TH INTERNATIONAL BALKAN EDUCATION AND SCIENCE CONFERENCE. [S.l.], 2014. Citado na página 6.
- LITTO, F. M. Repensando a educação em função de mudanças sociais e tecnológicas e o advento de novas formas de comunicação. In: *Litto, FM CONGRESSO IBEROAMERICANO DE INFORMÁTICA EDUCATIVA*. [S.l.: s.n.], 1996. v. 3. Citado na página 6.
- MORAES, J. B. D. *Técnicas para levantamento de Requisitos - DevMedia*. 2020. DevMedia. Disponível em: <<https://www.devmedia.com.br/tecnicas-para-levantamento-de-requisitos/9151>>. Acesso em: 09 jan 2020. Citado na página 10.
- PRESSMAN, R.; MAXIM, B. *Engenharia de Software-8ª Edição*. [S.l.]: McGraw Hill Brasil, 2016. Citado na página 9.
- SPÍNOLA, R. O. *Engenharia de Software - Engenharia de Requisitos - DevMedia*. 2020. DevMedia. Disponível em: <<https://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-engenharia-de-requisitos/8034>>. Acesso em: 09 jan 2020. Citado na página 11.
- TEAM, B. *Framework Front-end Bootstrap*. 2021. EIC. Disponível em: <<https://getbootstrap.com/>>. Acesso em: 21 Mai 2021. Citado na página 17.