

William Luis Alves Ferreira

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

Brasil

2020, 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

2020, Agosto

Resumo

Este relatório descreve as atividades realizadas para concluir o projeto de revitalização do jogo Caça Palavras: Doenças Tropicais, 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 2019-2020 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: Paradigma Orientado à Componentes. bibliotecas JavaScript. ReactJS. AngularJs. Engenharia de *Software*. *Frameworks Front-end*. *Material Design*. *Game Design*. Interface do Usuário (UI). Experiência do usuário (UX).

Lista de ilustrações

Figura 1 – Diagrama dos processos de levantamento de requisitos	9
Figura 2 – Esquema de levantamento de requisitos	10
Figura 3 – Captura da tela inicial do jogo - Versão 1.0	12
Figura 4 – Captura de tela dos principais elementos visuais do jogo - Versão 1.0 .	13
Figura 5 – Captura de tela dos elementos de mecânica - Versão 1.0	14
Figura 6 – Esquema de geração aleatória de caça palavras NxM	16
Figura 7 – Quadros exportados do Figma - Versão mobile Beta	17
Figura 8 – Quadros exportados do Figma - Versão <i>desktop</i> Beta	18
Figura 9 – Captura das telas implementadas - Versão Beta	19
Figura 10 – Principais quadros - Versão Final <i>desktop</i>	21
Figura 11 – Principais quadros - Versão Final <i>mobile</i>	22
Figura 12 – Captura de tela <i>desktop</i> Caça Palavras: Doenças tropicais (Versão 2.0)	23
Figura 13 – Captura de tela <i>mobile</i> Caça Palavras: Doenças tropicais (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

1	INTRODUÇÃO	6
2	OBJETIVOS	7
3	METODOLOGIA	8
3.1	Estrutura de Produção de <i>Software</i>	8
3.1.1	Produção de Requisitos	10
3.1.2	Levantamento	10
3.1.3	Registro	11
3.2	Proposta de Revitalização	11
3.2.1	Levantamento de requisitos	11
4	RESULTADOS	12
4.1	Projeto do Jogo: Proposta de Revitalização	12
4.1.1	Requisitos Estéticos	12
4.1.1.1	Conceito Artístico	13
4.1.1.2	Mecanismos	13
4.1.1.3	Progressão de Dificuldade	14
4.1.1.4	Interface do Usuário - UI	15
4.1.2	Requisitos Funcionais	16
4.1.3	Protótipo de <i>Layout</i>	17
4.2	Versão Beta	18
4.3	Versão Final	20
4.3.1	Layout Final	20
4.3.2	Descrição Complementar: Novidades da versão	20
5	CONCLUSÃO	25
5.1	Comentários Complementares	25
5.2	Implementação Futura	25
	REFERÊNCIAS	26

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 contempla a revitalização do jogo Caça Palavras - Doenças Tropicais EIC (2020) 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. 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 Caça Palavras - Doenças Tropicais [EIC \(2020\)](#), 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, Caça Palavras: Doenças Tropicais.

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*¹ do sistema

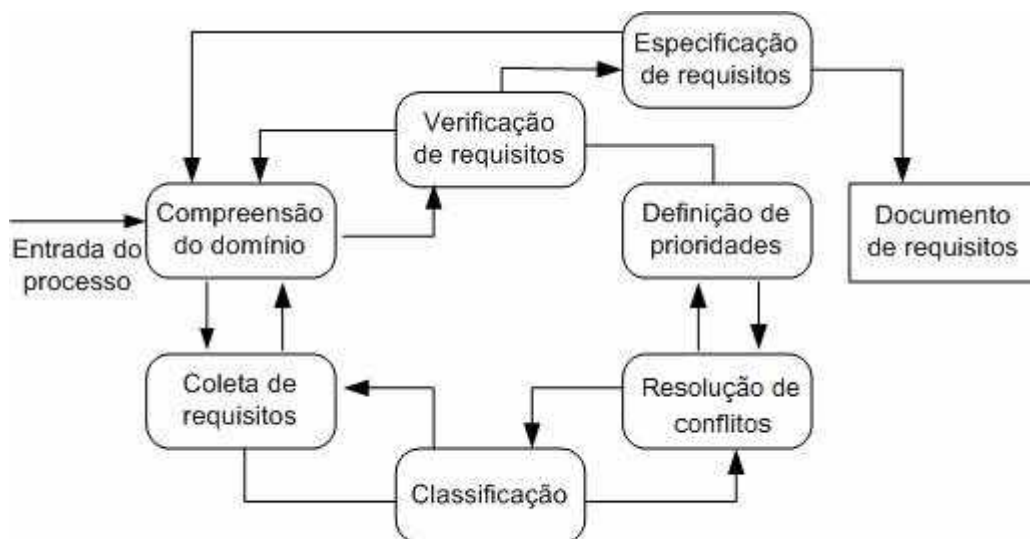
¹ stakeholders - O mesmo que: *view*, traduz-se como **hierarquia de visões**, comum a gerenciamento de banco de dados.

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.

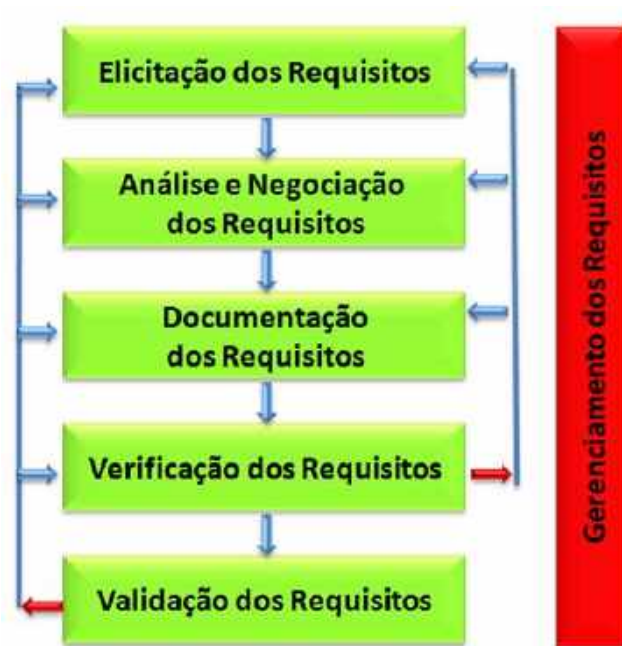
Figura 1 – Diagrama dos processos de levantamento de requisitos



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).

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.

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

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 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;
- b) *Frameworks*/Bibliotecas;
- c) Serviços de homologação - Configuração do servidor de hospedagem.

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 (beta) 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 Caça Palavras: Doenças Tropicais (Versão 2.0) que estará disponível na íntegra no site do EIC na segunda quinzena de agosto de 2020.

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

Analizando 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 (2020)

Ressaltando que esse projeto trata-se da revitalização do jogo já existente no repertório do EIC, Caça Palavras: Doenças Tropicais [EIC \(2020\)](#); 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 contadores de pontuação e tempo 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 comportamento do mouse durante a seleção das letras causado provavelmente pela interface tratada posteriormente neste documento.

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



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

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, diminuir a matriz de letras e aumentar o tamanho das células parece necessário à primeira vista.

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.

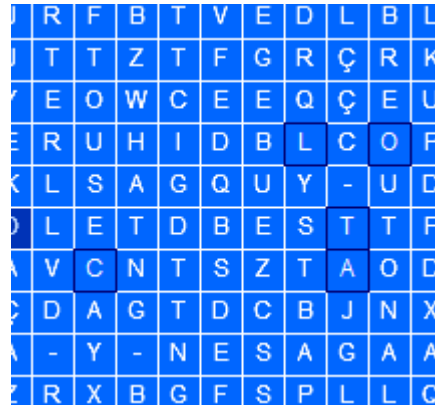
Solução Adotada: Utilizar paradigmas para desenvolvimento responsivo através de *frameworks single-view* deve ser o suficiente para um processamento de *assets* em tempo de execução, por exemplo, aplicando Angular2, ReactJs, ou, até mesmo, apenas desenvolver em JavaScript. Como nenhum trecho de código 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 caça-palavras “O Coquetel” [Coquetel \(2020\)](#) não deixa ambiguidade nas mecânicas presentes no jogo, porém, durante a execução,

é perceptível a ausência de um indicativo de acerto parcial - 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 clique e arraste para selecionar as letras interfere na intuição de jogabilidade.

Figura 5 – Captura de tela dos elementos de mecânica - Versão 1.0



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

Proposta: Buscar *framework*/biblioteca que possibilite o clique e arraste para seleção das letras e acrescentar resposta visual/sonora dos acertos parciais.

Requisitos: Uma ferramenta que permita um controle refinado de eventos, em particular do mouse, será necessário para implementar a mecânica de clique e arraste, 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, finalização do jogo, cronômetro, placar, assim por diante, mas não apresenta suporte a eventos de clique e arraste complexos em comparado com o improvisado através de Canvas (visto em caça palavras comerciais virtuais). Por isso, faz-se necessário a busca por biblioteca que possibilite a mecânica de clique e arraste 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 pontuação não é evidente durante o jogo, assim como, não é notório as causas da perda de pontuação, além de que, o objetivo de apenas evitar a perda de pontos torna o jogo desmotivador para o usuário; com isso, a progressão do jogo como um todo é falha por não proporcionar escalabilidade.

Proposta: Basear-se na dificuldade de caça palavras comerciais, e, assim, aumentar a escalabilidade do jogo com mais fases, inclusive, com a primeira fase “fácil” com caráter de introdução do jogador com as mecânicas propostas e seus objetivos. Uma alternativa seria a convergência da aba “instruções” (*pop-up* acionado ao clicar no link instruções) com uma “fase-tutorial”. O uso de elementos dinâmicos para marcar as penalidades ou ganhos do jogador é o fator crucial para torná-lo fluido e divertido.

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 ao jogo podem cumprir o papel de introduzir os jogadores as mecânicas básicas do jogo e seus objetivos.

4.1.1.4 Interface do Usuário - UI

O jogo a ser revitalizado possui um único *layout* - o tabuleiro do caça palavras, as dicas e os indicadores de tempo e pontuação - 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 6 palavras durante o tempo decorrido sem interrupção, 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 direto das folhas de estilos (i.e, *Cascading Style Sheets*, CSS sigla em inglês) sem o uso de bibliotecas/*frameworks* de *front-end*, 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

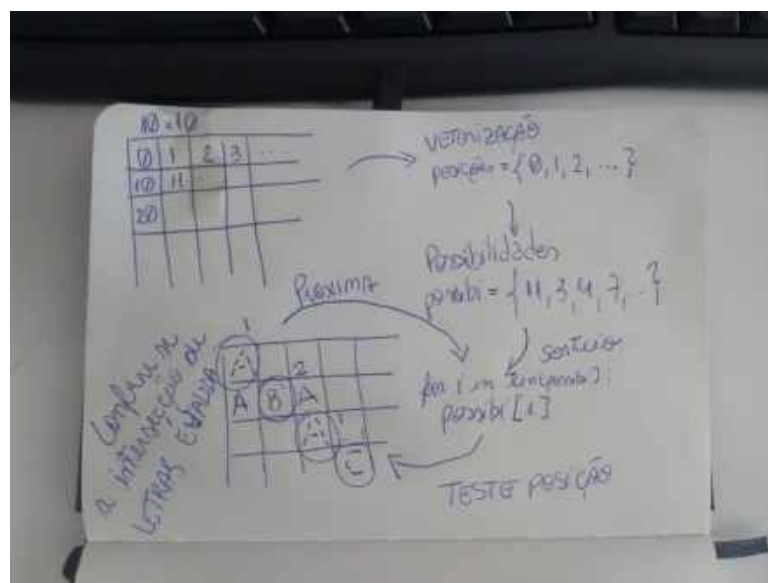
Com a análise realizada anteriormente em Requisitos Estéticos (subseção 4.1.1), extraímos os tópicos e diretrizes para esboçar o projeto, e, deste ponto, planejar o desenvolvimento, andamento e produção da revitalização em termos funcionais.

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, horizontal e diagonal), 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 caça palavras aleatórios 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 6.

Figura 6 – Esquema de geração aleatória de caça palavras NxM

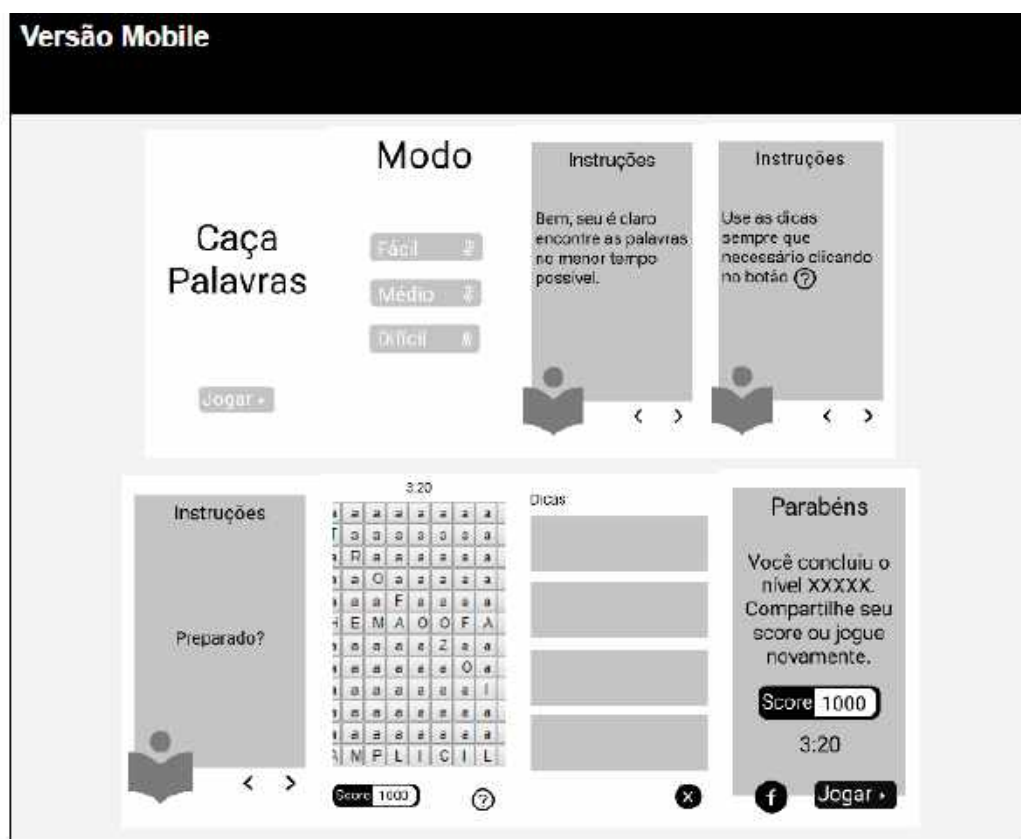


Fonte: Próprios autores

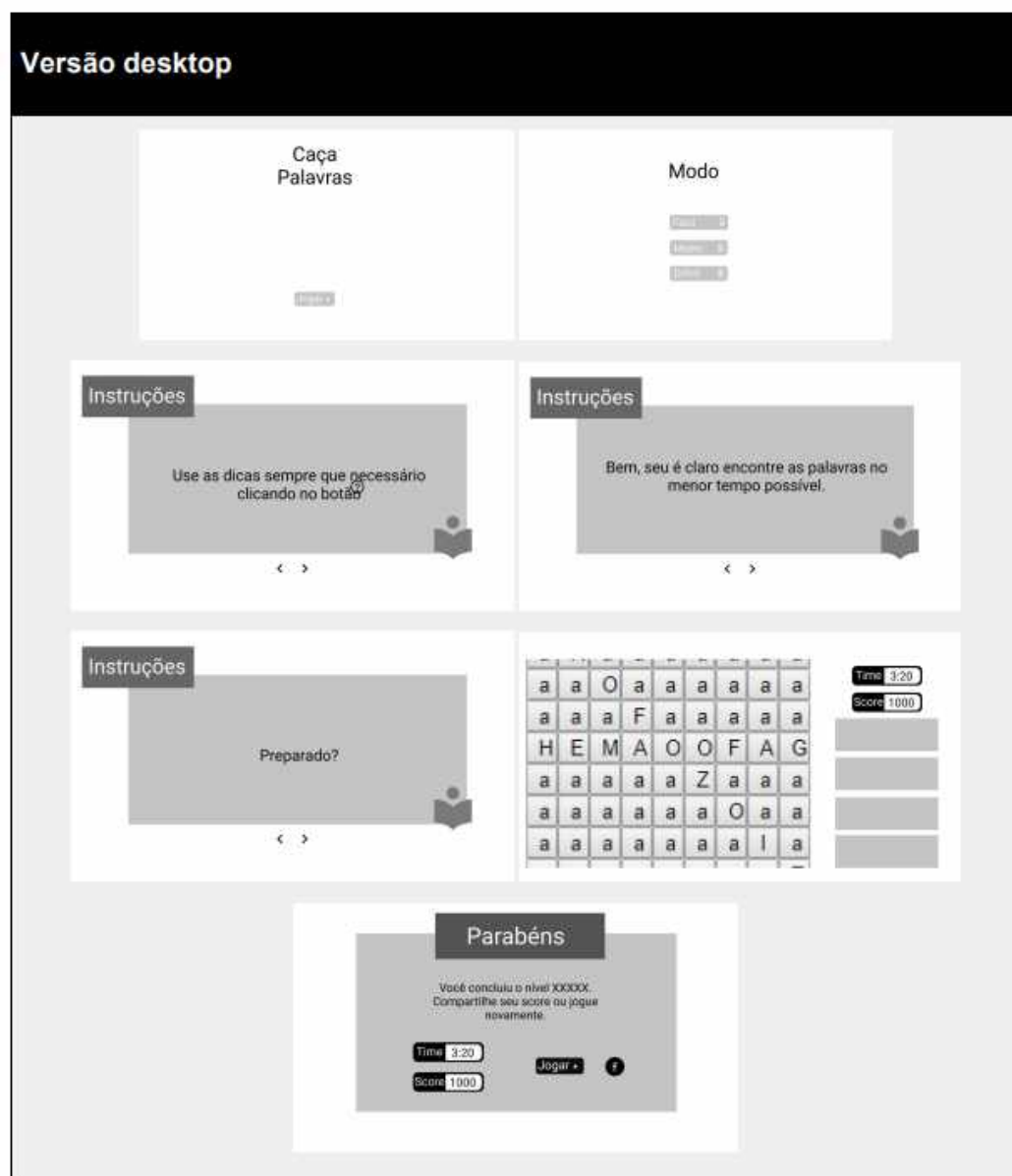
4.1.3 Protótipo de *Layout*

Utilizando a ferramenta *Figma* [Figma \(2020\)](#), 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 7 e 8.

Figura 7 – Quadros exportados do Figma - Versão mobile Beta



Fonte: Próprios autores

Figura 8 – Quadros exportados do Figma - Versão *desktop* Beta

Fonte: Próprios autores

4.2 Versão Beta

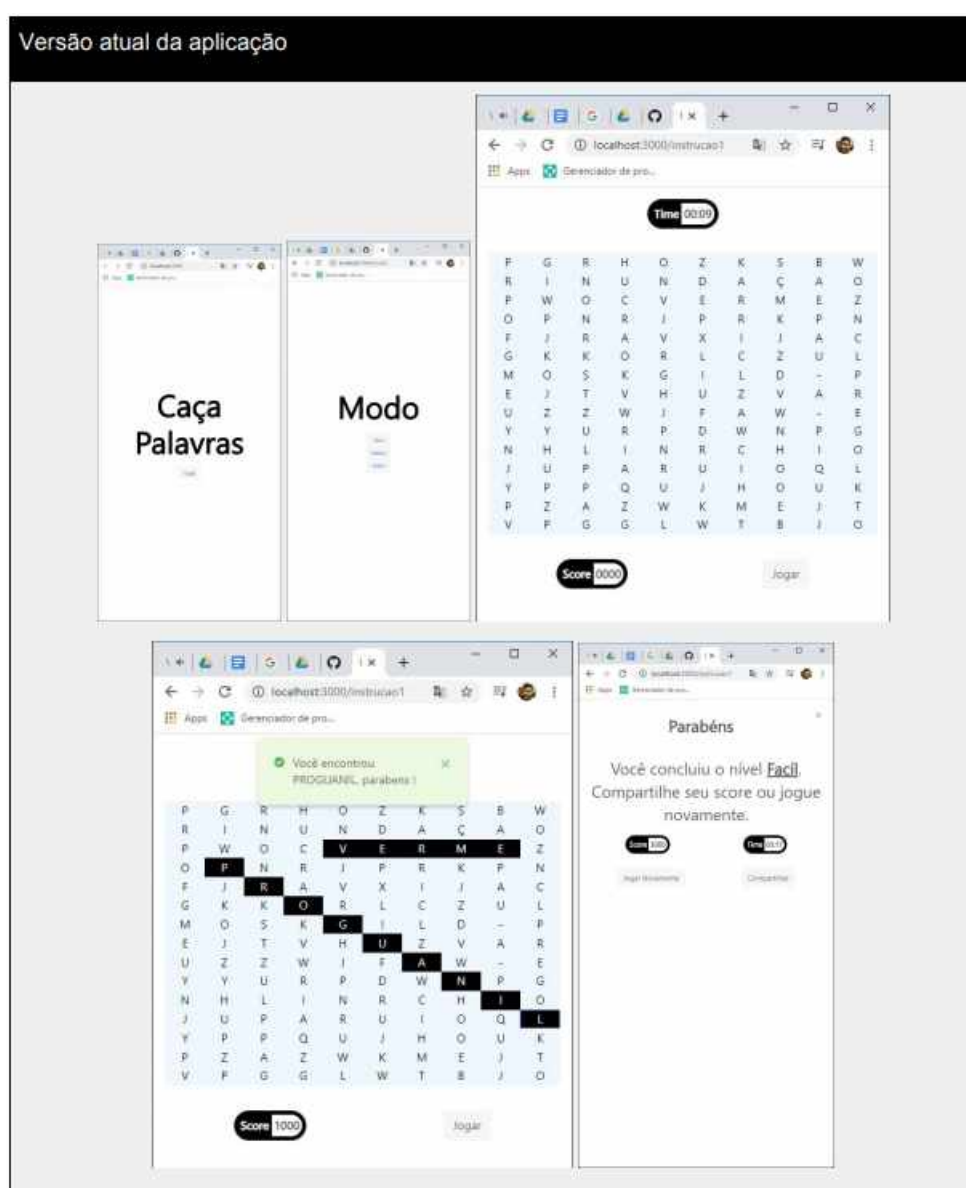
Esta versão foi produzida e concluída durante a produção do relatório parcial elaborado para o PUB. Nessa versão as funcionalidades, desenvolvimento da Experiência de Usuário (UX - do inglês User Experience) e Interface de Usuário (UI - do inglês User Interface) implementadas foram:

- Distribuição aleatória de palavras a qualquer dimensão do tabuleiro de letras;
- Nível de dificuldade;

- c) Roteamento de telas;
- d) Cronômetro;
- e) Pontuação;
- f) Estrutura de balanceamento pré concebida;
- g) Estrutura de *Layout Front-end Flexbox-CSS*;
- h) Tela de finalização do jogo.

O resultado da implementação desta versão pode ser observada na Figura 9

Figura 9 – Captura das telas implementadas - Versão Beta



Fonte: Próprios autores

4.3 Versão Final

Por fim, a versão final da revitalização do jogo, Caça Palavras: Doenças Tropicais (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/caca-palavra. Segue nas figuras 12 e 13 as capturas de tela desta versão (2.0).

4.3.1 Layout Final

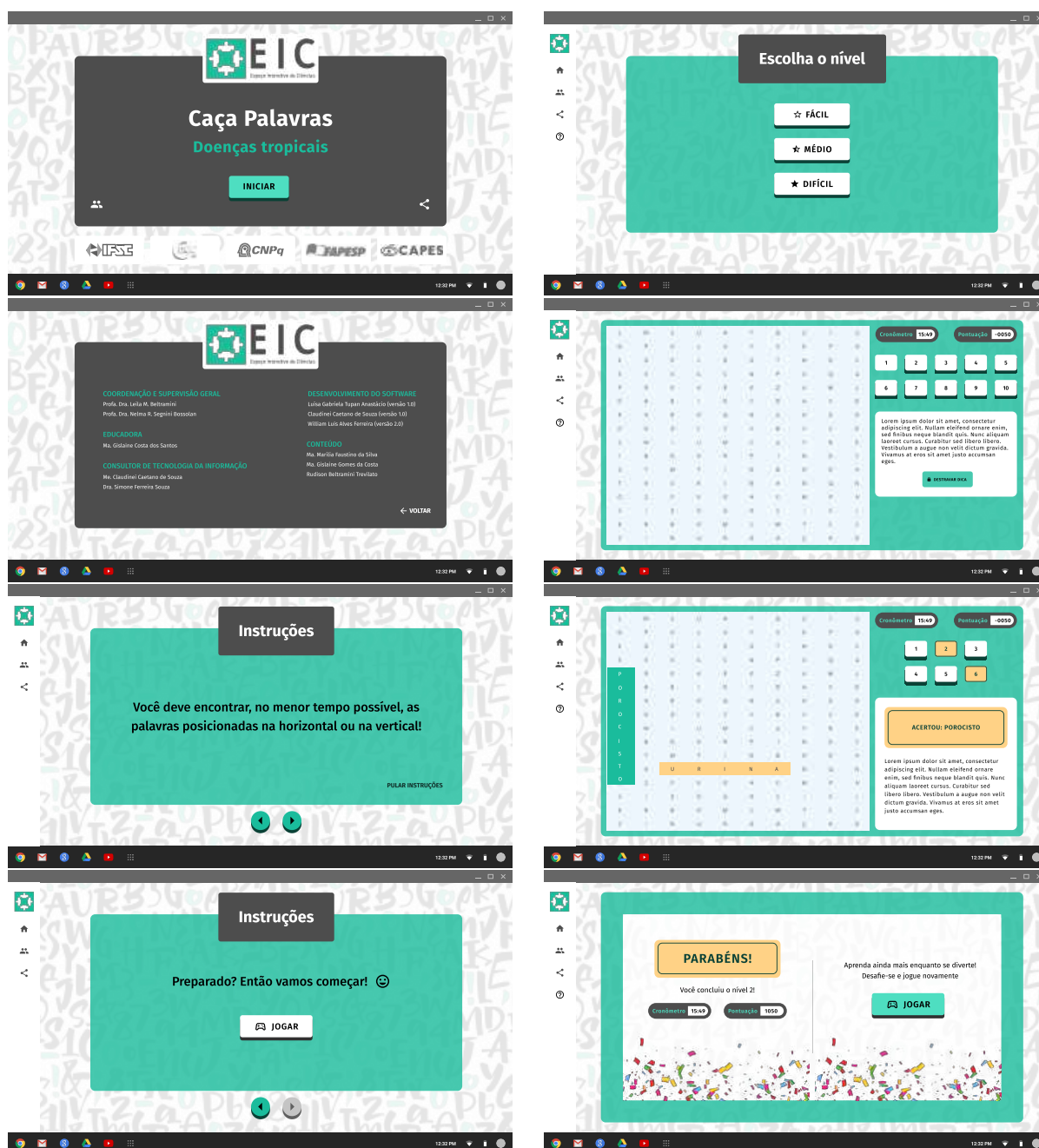
O *layout* desenvolvido pela consultora técnica de informação Dra. Simone Ferreira Souza apresentados na figura 10 e 11 representa a revisão e aprimoramento tendo como base os *Layout* apresentado nas figuras 7 e 8.

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, ou seja, sem uso de *framework*/biblioteca, e, por fim, documentação interna sobre o formato JavaDoc3.

Complementando a seção 4.2, todas as funcionalidades e diretrizes implementadas foram:

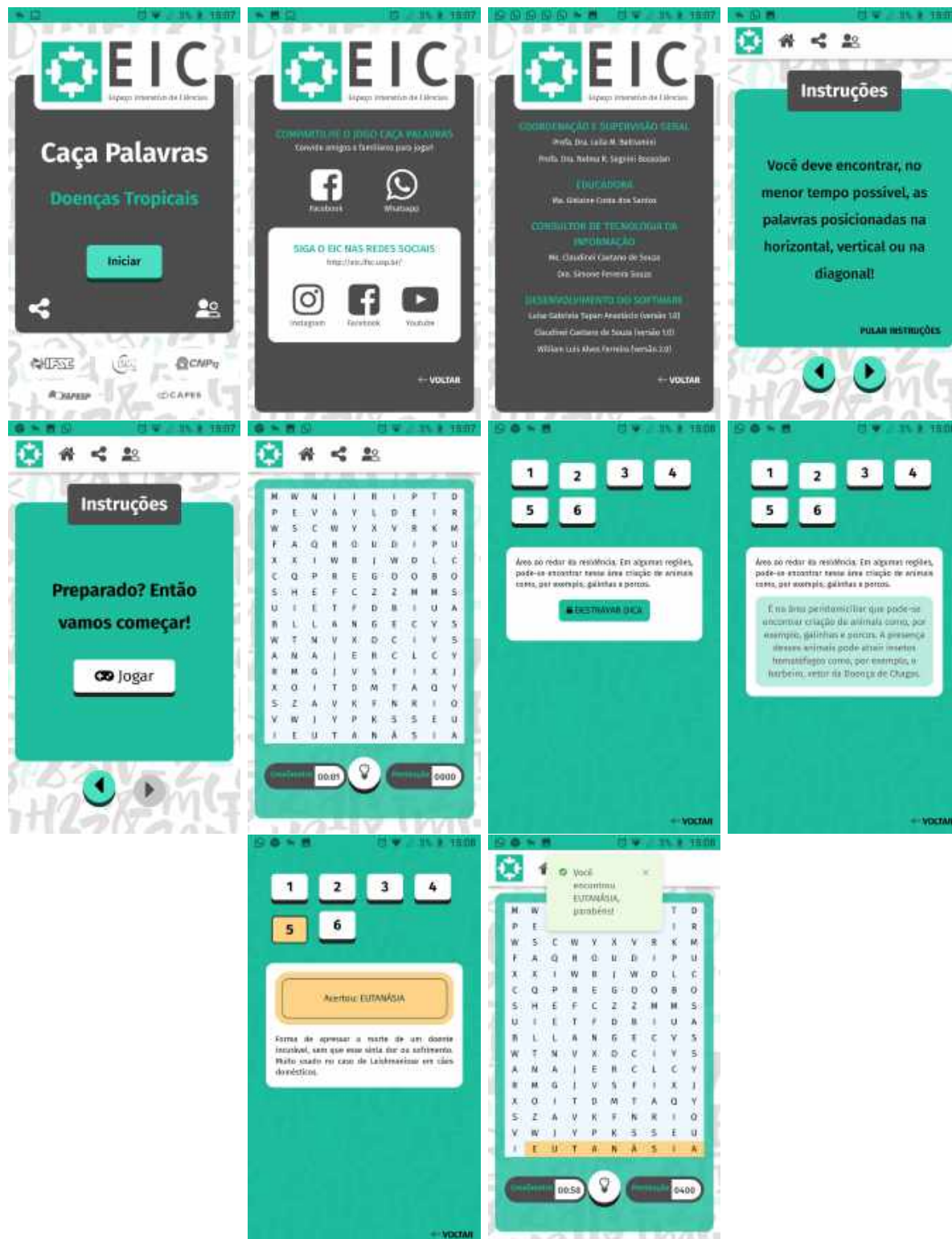
- a) Distribuição aleatória de palavras a qualquer dimensão do tabuleiro de letras;
- b) Nível de dificuldade;
- c) Roteamento de telas;
- d) Cronômetro;
- e) Pontuação;
- f) Estrutura de balanceamento, ou seja, flexibilidade em configurar o tabuleiro, por exemplo, definir conjunto de X palavras, Y linhas e W colunas, para qualquer X, Y e W;
- g) Estrutura de *Layout Front-end Flexbox-CSS*;
- h) Tela de finalização do jogo;
- i) Compartilhamento nas redes sociais;
- j) Apresentação adequada dos patrocinadores e participantes no desenvolvimento do projeto;
- k) Identidade Visual alinhada as diretrizes do EIC.

Figura 10 – Principais quadros - Versão Final *desktop*

Fonte: Próprios autores

Figura 11 – Principais quadros - Versão Final *mobile*

Fonte: Próprios autores

Figura 13 – Captura de tela *mobile* Caça Palavras: Doenças tropicais (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 com produção em ritmo constante.

5.1 Comentários Complementares

O entendimento sobre os aspectos didáticos incluídos no jogo ainda faz-se necessário para escalabilidade de forma correta dos níveis, 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.

A estrutura básica do jogo é caça palavras genérico, ou seja, sua aplicação não se restringe ao tema Doenças tropicais, 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”.

5.2 Implementação Futura

Em caso de manutenção ou de novas revitalizações seria imprescindível a implementação da mecânica “clique e arraste” para a seleção intuitiva das palavras, pois, infelizmente, tornou-se inviável com a biblioteca e paradigmas selecionados para produção deste projeto.

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.

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 13.
- 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*. 2020. O Coquetel. Disponível em: <<http://coquetel.com.br/>>. Acesso em: 09 jan 2020. Citado na página 13.
- 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 8.
- EIC. *Caça Palavras: Doenças Tropicais*. 2020. EIC. Disponível em: <<http://eic.ifsc.usp.br/caca-palavra/>>. Acesso em: 09 jan 2020. Citado 5 vezes nas páginas 6, 7, 12, 13 e 14.
- FIGMA, G. *Figma*. 2020. Figma. Disponível em: <<https://www.figma.com/about/>>. Acesso em: 09 jun 2020. Citado na página 17.
- 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 9.
- PRESSMAN, R.; MAXIM, B. *Engenharia de Software-8ª Edição*. [S.l.]: McGraw Hill Brasil, 2016. Citado na página 8.
- 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 10.