

SOCIALIZAÇÃO ENTRE OS MEIOS DIGITAIS GAMIFICADOS: uma proposta de conexão dos indivíduos da Era Moderna

Daniel Giaqueto Gomes
Graduado em Ciência da Computação – Uni-FACEF
daniel_giaqueto_gomes@outlook.com

Mateus de Oliveira Custódio
Graduado em Engenharia de Software – Uni-FACEF
mateusolivera727@gmail.com

Carlos Alberto Lucas
Docente do Uni-FACEF
projetos@profcarloslucas.com.br

Resumo

O presente trabalho investiga o impacto profundo da evolução tecnológica na sociedade contemporânea, com ênfase particular na transformação das interações humanas mediadas por aplicativos de relacionamento. Dentro deste escopo, propõe-se a criação e análise de um software direcionado à esfera dos jogos online. A principal motivação é estabelecer um ambiente em que os jogadores possam se conectar e compartilhar suas experiências virtuais. Inspirados pela revolução das conexões humanas em aplicativos como o Tinder, os autores visam introduzir uma solução inovadora no universo dos jogos, criando uma plataforma de interação especializada para entusiastas de jogos online.

Palavra-chave: Socialização. Redes sociais. Games. Desenvolvimento de software.

Abstract

This study investigates the profound impact of technological evolution in contemporary society, with a particular emphasis on the transformation of human interactions mediated by relationship apps. Within this scope, it proposes the creation and analysis of software targeted for the realm of online games. The main motivation is to establish an environment where players can connect and share their virtual experiences. Inspired by the revolution of human connections in apps like Tinder, the authors aim to introduce an innovative solution in the gaming universe, creating a specialized interaction platform for online gaming enthusiasts.

Keywords: Socialization. Social media. Games. Software development.

1 Introdução

A constante evolução da tecnologia tem impactado profundamente diversos aspectos da sociedade contemporânea. Entre esses, destaca-se o domínio dos aplicativos de relacionamento, que transformaram a maneira como as pessoas se conectam e interagem. Inspirado nesse contexto, o presente trabalho propõe a criação e análise de um software, voltado para uma vertente específica: os jogos online. Este artigo busca contextualizar a relevância do tema e apresentar a motivação subjacente

à criação deste software, que visa proporcionar um ambiente de conexão para jogadores que desejam compartilhar suas experiências virtuais.

O estudo foi dedicado a conceber, criar e, rigorosamente, avaliar um software cujo foco é servir como uma plataforma de interação para aficionados por jogos online. Inspirados pela eficácia e popularidade de aplicativos como o Tinder, que revolucionaram a maneira como as pessoas se conectam e estabelecem relacionamentos na era digital, pretende-se oferecer uma solução inovadora no universo dos jogos. Esta plataforma não só facilitará a conexão entre jogadores com interesses e habilidades similares, mas também fomentará uma comunidade global, em que os usuários poderão compartilhar experiências, estratégias e criar laços com base em seu interesse em comum, relacionado aos jogos.

Em uma era dominada pela tecnologia e pela necessidade de conexão virtual, o objetivo é preencher a lacuna ainda existente no mercado de jogos, que não foi capaz de propiciar a construção de conexões sociais, proporcionando um ambiente virtual seguro e envolvente para todos os entusiastas do universo gamer. Para tanto, as seguintes ferramentas foram utilizadas: React, Vite, React Native, Hoppscotch, TypeScript, NodeJS, SQL Lite e Prisma, responsáveis por tonar possível a realização do projeto.

A estrutura deste artigo é delineada da seguinte forma: A seção 2 dedica-se ao referencial teórico. Em seguida, a seção 3 discorre sobre as ferramentas e metodologias adotadas. A seção 4, por sua vez, detalha o processo de desenvolvimento do software e os procedimentos de teste dos modelos. Os resultados são elucidados na seção 5, enquanto a seção 6 reserva-se para as considerações finais e reflexões sobre o projeto.

2 Referencial Teórico

Nesta seção, serão abordados o tema do projeto, suas aplicações, conceitos, fundamentos e técnicas utilizadas em seu desenvolvimento.

2.1 Socialização

Segundo Émile Durkheim (1975), ao declarar que: “A educação consiste na socialização da geração jovem pela geração adulta”, entende-se que o processo de socialização se inicia através da intrincada teia de relações sociais que se desenvolvem entre os indivíduos ao longo de suas vidas.

A socialização do ser humano é um processo intrincado e vital que começa no momento do nascimento e continua ao longo de toda a vida. Ela envolve a internalização de valores, normas, tradições e comportamentos de um determinado grupo ou sociedade. Através da socialização, os indivíduos aprendem a compreender seu papel no mundo e como interagir de maneira produtiva e harmoniosa com os outros.

Desde os primeiros momentos de vida, os seres humanos são imersos em um processo de aprendizado social. A família é, geralmente, o primeiro e mais influente agente de socialização. É através dela que se aprendem as noções básicas de comportamento, linguagem, moralidade e cultura. À medida que se cresce, outros

agentes, como escolas, amigos, mídia e instituições religiosas, desempenham papéis fundamentais na formação da identidade do indivíduo e percepção de mundo.

A socialização do ser humano tem sido objeto de estudo de diversos filósofos e sociólogos ao longo dos anos. Eles se aprofundaram em como os indivíduos se integram à sociedade e como essa integração afeta suas identidades e compreensões de mundo.

Desde os primórdios da existência humana, o ser humano anseia por meios de expressar sua essência e emoções internas. Esse impulso de autoexpressão tem se manifestado em cada época, em cada gesto e em todas as ações realizadas. Desde o primeiro ritual fúnebre até um pai protegendo seu filho de uma ameaça, evidencia-se o intrínseco entendimento e valorização das conexões profundas e do carinho que se nutre uns pelos outros.

Durante a pandemia da COVID-19, observou-se uma drástica redução nas interações entre as pessoas. Diante deste cenário, a humanidade teve que se reinventar para atender sua intrínseca necessidade de conexão social. Aristóteles, em sua filosofia, defendia que o ser humano é, por natureza, um ser social e que seus relacionamentos são fundamentais para prosperar. Segundo o filósofo, o convívio e as relações humanas são vitais para a felicidade, bem-estar e evolução moral (Tavares, 2023).

Portanto, a socialização é um processo fundamental na formação do ser humano, moldando não apenas a maneira como o indivíduo se percebe, mas também como ele interage e se relaciona com o mundo ao seu redor. Desde os primeiros momentos de vida, o indivíduo é imerso em uma rede de interações que lhe transmitem valores, normas, tradições e comportamentos característicos de sua sociedade. Essas interações, sejam elas com a família, escola, amigos ou outras instituições, são cruciais para o desenvolvimento de sua identidade e sua capacidade de se integrar harmoniosamente à comunidade.

Os processos de socialização também são reflexos das transformações históricas, sociais e culturais. Em diferentes épocas e culturas, os métodos e os valores transmitidos podem variar, mostrando a dinamicidade e adaptabilidade humana diante dos desafios e contextos apresentados. Além disso, é através da socialização que as gerações mais velhas transmitem seu legado e sabedoria para as mais novas, garantindo a continuidade e evolução da sociedade.

Em uma era globalizada e com avanços tecnológicos constantes, os desafios da socialização tornam-se ainda mais complexos. As plataformas digitais, por exemplo, introduziram novos modos de interação, ampliando o alcance das redes sociais, mas também levantando questões sobre autenticidade, privacidade e as nuances das relações online.

No entanto, mesmo diante de desafios, é possível encontrar aspectos positivos. Durante a pandemia de 2019, a necessidade inerente de conexão humana foi ressaltada. Esse período, repleto de incertezas, também marcou uma transformação significativa nas maneiras como as pessoas se conectam. Plataformas, como Zoom, emergiram como ferramentas essenciais, permitindo que estudantes continuassem seus estudos remotamente. Simultaneamente, aplicações como Discord e Skype possibilitaram a realização de reuniões profissionais e proporcionaram momentos de descontração e diálogo entre amigos, demonstrando a adaptabilidade e resiliência humanas diante de adversidades.

2.2 Ser humano, um ser social

Desde tempos ancestrais, o ser humano é considerado um ser intrinsecamente social, incorporando as normas e valores presentes na família, entre seus pares e na sociedade em geral. Portanto, a formação da personalidade do ser humano é resultado, de acordo com Savoia (1989), “[...] de um processo de socialização, onde desempenham um papel fatores inatos e adquiridos”. O que se entendem por fatores inatos são características herdadas geneticamente dos familiares, enquanto os fatores adquiridos derivam da influência social e cultural.

Conforme Strey (2002), ao vir ao mundo, cada pessoa “ingressa em um contexto social estabelecido ao longo de várias gerações e absorvido por meio de interações sociais”. Desde o começo de sua vida, o ser humano é identificado como um ser profundamente inserido na sociedade, assimilando as normas e valores presentes na família, entre seus pares e na comunidade onde reside. Essa natureza social é evidente em várias manifestações culturais, linguísticas e comportamentais.

Desde a pré-história, onde o trabalho conjunto era essencial para a sobrevivência, até as complexas redes de interações da era digital, é notório como os seres humanos são moldados e influenciados pelas conexões com os pais. Segundo Savoia (1989, p. 54), “[...] a formação da personalidade do ser humano é decorrente de um processo de socialização, no qual intervêm fatores inatos e adquiridos.” Com isso, ele quer dizer que os seres humanos herdaram a personalidade geneticamente de seus familiares, e os fatores adquiridos provêm da natureza social e cultural. Essas relações são refletidas nas tradições, na educação e na forma como cada indivíduo compreende e se posiciona no mundo.

2.3 Conceitos de Software

Em 1958, o cientista americano John Wilder Tukey introduziu o termo “Software” em um artigo (Moura, 2020). Ele também foi o pioneiro em usar a palavra “bit” para se referir a um “dígito binário”.

De maneira simples, um software é um conjunto de instruções codificadas que permitem ao computador executar tarefas específicas. É um programa de computador que direciona as operações de um sistema baseado em computador. Este pode variar desde aplicativos simples, como editores de texto, até sistemas complexos, como sistemas operacionais.

Um software é um conjunto de instruções codificadas que orienta o computador a realizar tarefas específicas. Trata-se de um programa que gerencia as operações de um sistema baseado em computador. Ele pode abranger desde aplicativos básicos, como calculadoras, até sistemas complexos como sistemas de gestão empresarial (ERP) de banco de dados.

Segundo Stair e Reynolds (2011), o software consiste em programas que comandam a operação do computador. Para Fernandes (2002), software é uma sentença escrita em uma linguagem computável, para a qual existe uma máquina (computável) capaz de interpretá-la. O software é um conjunto de instruções e informações de dados armazenadas digitalmente. Quando o computador interpreta essas instruções, ele executa tarefas específicas para as quais o software foi criado e projetado.

Cada dispositivo eletrônico, seja ele um computador, smartphone, tablet ou até mesmo uma calculadora, contém vários softwares. Esses softwares são o

“cérebro” lógico do dispositivo, interpretando e processando dados para os componentes físicos (hardwares) funcionarem corretamente. A operação desses softwares se baseia em comandos de linguagens de programação configurados para cada hardware, necessitando de um sincronismo perfeito entre a lógica (software) e o físico (hardware). Esses conjuntos de instruções, quando agrupados, dão origem a programas complexos, e as informações geradas por eles são interpretadas pelo dispositivo.

2.4 Boas práticas

O desenvolvimento de software tem enfrentado desafios crescentes à medida que a tecnologia avança, tornando as boas práticas em programação mais cruciais do que nunca. Estudos científicos na área de tecnologia da informação têm consistentemente enfatizado a importância de criar um código que seja não apenas funcional, mas também eficiente, confiável e sustentável.

Uma consideração primordial é a legibilidade do código. Pesquisas revelam que os programadores frequentemente gastam mais tempo lendo código do que realmente escrevendo, tornando a clareza uma prioridade. Isso pode ser alcançado através da nomenclatura apropriada de variáveis, comentários relevantes e uma estruturação clara do código.

Além disso, a reutilização de código é outra prática promovida pela literatura. Ao aproveitar funções e componentes existentes, os desenvolvedores podem reduzir erros, otimizar o tempo de desenvolvimento e promover consistência.

A importância dos testes unitários também é amplamente reconhecida. Eles desempenham um papel vital na identificação precoce de defeitos, garantindo a robustez do software e diminuindo falhas quando em produção. A revisão de código entre pares é outra prática valiosa, pois permite que os desenvolvedores identifiquem e corrijam problemas que podem ser negligenciados durante o desenvolvimento inicial.

O princípio DRY (“Don't Repeat Yourself”) evita a repetição desnecessária no código, não apenas facilitando a manutenção, mas também garante que o software permaneça coeso e livre de inconsistências (Smanioto, 2020).

Em resumo, a adoção de boas práticas em programação, conforme evidenciado por estudos científicos, é fundamental para atender às demandas da indústria moderna de TI, garantindo que os softwares sejam de alta qualidade e atendam às necessidades dos usuários.

3 Materiais e Métodos

Nesta seção, serão apresentadas as ferramentas e as tecnologias utilizadas para o desenvolvimento do projeto.

3.1 HTML 5

A evolução da Web exigiu o desenvolvimento de tecnologias cada vez mais sofisticadas para permitir a criação de sites e aplicações interativas, acessíveis e eficientes. A quinta versão do HyperText Markup Language, comumente referida como HTML5, emergiu como uma resposta a essas demandas crescentes. Originária

do padrão HTML criado no início dos anos 90, a HTML5 foi projetada e aperfeiçoada pela World Wide Web Consortium (W3C) para abordar as limitações de suas versões antecessoras, ao mesmo tempo em que se adaptava às novas exigências dos desenvolvedores e designers.

Uma das características notáveis do HTML5 é a melhoria na semântica do documento web. Novas tags, como `<article>`, `<section>`, `<nav>`, e `<figure>`, foram introduzidas para permitir que os desenvolvedores descrevam mais precisamente o conteúdo. A introdução de elementos gráficos, como `<canvas>` e `<svg>`, revolucionou a renderização gráfica no navegador, permitindo desenhos 2D sem a necessidade de plugins externos. Além disso, a inclusão do elemento `<video>` e `<audio>` eliminou a necessidade de plugins de terceiros para a reprodução de mídia, simplificando e padronizando a experiência multimídia na web (PEREIRA, 2023).

A conectividade também foi aprimorada no HTML5. Recursos avançados, como WebSockets, oferecem uma comunicação bidirecional em tempo real entre o servidor e o cliente. Este padrão trouxe ainda mecanismos, como o `localStorage` e o `sessionStorage`, que permitem que os sites armazenem dados no lado do cliente, melhorando a eficiência e a velocidade das aplicações web.

A maioria dos navegadores modernos oferece suporte extensivo ao HTML5, embora a adoção completa tenha sido um processo gradual. No entanto, com o tempo, o suporte tornou-se mais universal, consolidando o HTML5 como a escolha padrão para o desenvolvimento web.

Em resumo, o HTML5 representa um marco significativo na história da web, moldando práticas e padrões na comunidade de desenvolvimento e transformando a maneira como os sites são criados e experimentados pelos usuários. À medida que a web continua a evoluir, o HTML5 destaca-se como um testamento do compromisso da comunidade em adaptar-se e crescer em resposta às demandas da era digital.

3.2 CSS 3

O avanço da web trouxe consigo uma demanda crescente por designs mais dinâmicos e responsivos. Nesse contexto, o Cascading Style Sheets (CSS) emergiu como uma ferramenta essencial, e sua terceira versão, o CSS3, revolucionou a estilização web. Esta iteração se destacou por sua modularidade, permitindo uma implementação mais ágil e uma melhor compatibilidade entre navegadores.

Uma das maiores contribuições do CSS3 foi a introdução de animações e transições (Júnior, [s.d.]). Estes recursos permitiram que designers criassem movimentos fluidos e efeitos visuais sem depender de plugins ou scripts externos. Além disso, as transformações 2D e 3D trouxeram uma nova dimensão ao design, possibilitando efeitos como rotação, escala e inclinação de elementos na página.

As inovações não pararam por aí. O CSS3 facilitou a implementação de gradientes e sombras, anteriormente dependentes de imagens ou truques mais complexos. Essa mudança simplificou o processo e tornou as páginas mais leves em termos de carregamento.

Outro marco foi a introdução das “media queries”, que é um recurso do CSS3 utilizado para aplicar estilos específicos com base nas características do dispositivo em que o conteúdo está sendo visualizado, com isso, deram origem ao design

responsivo. Agora, os sites poderiam adaptar-se automaticamente às dimensões da tela do dispositivo do usuário, garantindo uma experiência otimizada, seja em um desktop ou smartphone.

Utilizou-se um framework chamado Tailwind CSS Lançado por Adam Wathan, Jonathan Reinink, David Hemphill, e Steve Schoger, o Tailwind CSS introduziu uma abordagem diferente para estilizar páginas web. Em vez de usar classes semânticas, como “.btn-red” ou “.header”, o Tailwind permite que os desenvolvedores usem classes de utilidade, como “.bg-red-500” ou “.text-xl”, para aplicar estilos diretamente. Esta abordagem, embora possa parecer complexa no início, oferece um nível incrível de flexibilidade, permitindo que os desenvolvedores criem interfaces complexas sem sair do HTML e sem a necessidade de escrever CSS customizado.

Por fim, a capacidade de incorporar fontes da web no CSS3 expandiu o leque de opções tipográficas para os designers, permitindo uma maior expressividade e coesão de marca nos sites. Em resumo, enquanto a web evoluía, o CSS3 desempenhou um papel central, proporcionando ferramentas que enriqueceram a estética e a funcionalidade dos websites modernos.

3.3 REACT/REACT NATIVE

O React é uma biblioteca JavaScript, lançada pelo Facebook em 2013, que trouxe consigo um novo paradigma para o desenvolvimento de aplicações web. Centrado no conceito de componentes, o React permite que desenvolvedores construam interfaces de usuário modulares e reutilizáveis, facilitando a manutenção e a escalabilidade de aplicações complexas (De Andrade, 2020).

O core do React se baseia em uma abordagem declarativa de programação, na qual os desenvolvedores especificam o que desejam visualizar, enquanto a biblioteca gerencia eficientemente as atualizações e renderizações no DOM através de um mecanismo conhecido como Virtual DOM. Esta característica garante que apenas as partes modificadas da interface sejam atualizadas, proporcionando uma experiência de usuário fluida e otimizada.

Em contrapartida, o React Native, lançado em 2015, levou os conceitos e a eficiência do React para o mundo do desenvolvimento móvel. Ao invés de traduzir componentes React para elementos do DOM, como no caso da web, o React Native traduz esses componentes para elementos nativos das plataformas iOS e Android. Isso significa que, com um único código-base escrito predominantemente em JavaScript, os desenvolvedores conseguem criar aplicações para ambos os sistemas operacionais, mantendo a sensação e a performance de apps nativos.

Além de reduzir o tempo e os recursos necessários para desenvolver aplicações móveis multiplataforma, o React Native tem uma comunidade robusta e em constante crescimento, oferecendo uma vasta gama de plugins e extensões que facilitam a integração com funcionalidades e serviços específicos.

3.4 SQL Lite

SQLite é uma maravilha da engenharia de software quando se trata de sistemas de gerenciamento de banco de dados relacional. Ao contrário dos tradicionais sistemas RDBMS, como MySQL, PostgreSQL ou Oracle, o SQLite não opera com

base na arquitetura cliente-servidor (Vieira, 2021). Em vez disso, ele é autocontido e não requer um servidor separado para funcionar. Isso significa que um aplicativo que usa SQLite tem seu próprio sistema de banco de dados integrado, tornando-o perfeitamente adequado para aplicações desktop, dispositivos móveis e até mesmo para alguns aplicativos de servidor de pequeno a médio porte.

Uma das características mais notáveis do SQLite é sua leveza. A biblioteca completa pode ocupar menos espaço do que um MB, dependendo das configurações de compilação. Além disso, o SQLite é projetado para ser eficiente e rápido, proporcionando uma solução robusta para armazenamento de dados sem o overhead ou a complexidade associada a sistemas RDBMS maiores.

Outra vantagem é sua portabilidade; é possível copiar um arquivo de banco de dados SQLite de um dispositivo ou plataforma para outro e esperar que ele funcione sem problemas. Isso se deve à natureza autônoma do SQLite, onde o banco de dados é armazenado em um único arquivo, facilitando backup, transferência e manutenção.

Embora o SQLite seja simples e leve, ele é poderoso, suportando a maioria dos recursos SQL padrão, incluindo transações ACID (Atomicidade, Consistência, Isolamento, Durabilidade). Devido à sua facilidade de uso, eficiência e versatilidade, o SQLite tornou-se a escolha padrão para muitos desenvolvedores ao implementar bancos de dados locais em aplicações e serviços.

4 Desenvolvimento

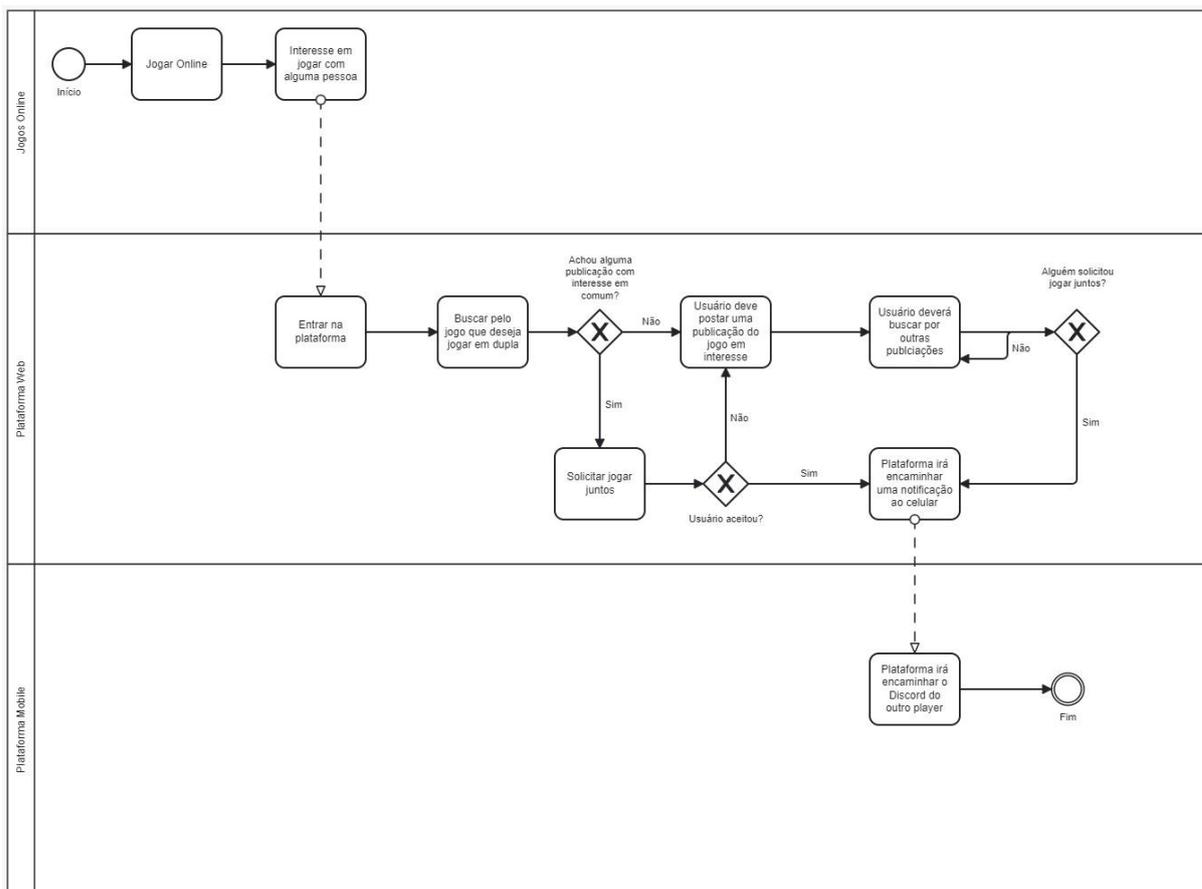
Esta seção tem por objetivo apresentar o ciclo de desenvolvimento seguido no projeto e o detalhamento de cada uma das etapas a serem apresentadas, como o desenvolvimento do software, implementação do banco de dados e o teste do software.

4.1 Ciclo do desenvolvimento

O projeto é uma plataforma web inovadora com o objetivo de conectar jogadores que, embora desconhecidos entre si, compartilham um interesse comum em jogos específicos, facilitando a formação de grupos para jogarem juntos. Esta solução não somente fortalece a comunidade de jogadores, mas também enriquece a experiência de jogo ao permitir que entusiastas se unam, independentemente de sua rede de contatos prévia.

Para dar vida a essa visão, optou-se por utilizar uma combinação tecnológica de API RestFul e SPA (Single Page Application). A API RestFul, no contexto de desenvolvimento web, refere-se a uma arquitetura que segue princípios REST (Representational State Transfer), permitindo que sistemas conversem entre si de maneira eficiente e padronizada. Já a abordagem SPA permite que nossa plataforma ofereça uma experiência de usuário fluida e responsiva, carregando conteúdo dinamicamente sem a necessidade de recarregar a página inteira. Esta combinação não apenas garante escalabilidade e performance, mas também proporciona uma interação intuitiva e agradável para nossos usuários.

Figura 1 – BPMN do projeto



Fonte: <https://bpmn.io>

4.2 5w2h – Plano de ação

What (O que é o produto?) - O produto é uma plataforma online que conecta jogadores que desejam encontrar parceiros para jogar jogos online em conjunto.

Why (Por que foi desenvolvido?) - Foi desenvolvido para proporcionar uma experiência mais envolvente e social aos jogadores online, permitindo que eles encontrem facilmente outros jogadores com interesses e níveis de habilidade semelhantes.

Who (Para quem é direcionado?) - A plataforma é direcionada a jogadores de todas as idades que desejam encontrar companheiros para jogar em equipe em jogos multiplayer online.

Where (Onde será utilizado?) - A plataforma é uma plataforma online acessível a partir de dispositivos como computadores, smartphones e tablets, permitindo que os jogadores acessem de qualquer lugar com conexão à internet.

When (Quando será utilizado?) - A plataforma estará disponível para uso contínuo, permitindo que os jogadores acessem e encontrem parceiros a qualquer momento, de acordo com sua necessidade.

How (Como funciona?) - Os jogadores têm a possibilidade de criar perfis na plataforma, indicando os jogos de seu interesse. Quando encontrarem outros usuários com interesses similares, poderão optar por jogar juntos, permitindo assim o compartilhamento dos seus respectivos canais no Discord. Além disso, é possível buscar por outros jogadores utilizando filtros como o jogo desejado, horários disponíveis e nível de habilidade, facilitando o envio de convites para partidas em conjunto.

How much (Quanto custa?) - A plataforma foi concebida como um recurso público em benefício dos jogadores, sem qualquer custo associado para os usuários. A fonte de receita será proveniente da inclusão de anúncios no site, visando a monetização da plataforma.

4.3 Desenvolvimento da tela inicial

Este programa é um elemento React que simboliza um painel de propaganda. O painel tem um encabeçamento, uma explicação e um botão “Faça um post e encontre seu Duo!”. Ele emprega estilos CSS para determinar a apresentação, abrangendo matizes de plano de fundo e dimensões.

O botão “Inicie uma Publicação” age como um disparador para inaugurar um diálogo ou modal, no qual os utilizadores podem elaborar uma publicação de propaganda.

Figura 2 – Print 1

```
TS CreateAdBanner.tsx X
web > src > components > TS CreateAdBanner.tsx > CreateAdBanner
1 import { MagnifyingGlassPlus } from "@phosphor-icons/react";
2 import * as Dialog from "@radix-ui/react-dialog";
3
4 export function CreateAdBanner() {
5   return (
6     <div className="pt-1 bg-duo-gradient self-stretch rounded-lg mt-8 overflow-hidden">
7       <div className="bg-[#2A2634] px-8 py-6 flex justify-between items-center">
8         <div>
9           <strong className="text-2xl text-white font-black">Cansado de jogar solo?</strong>
10          <span className="text-zinc-400 block">Faça um post e encontre um Duo!</span>
11        </div>
12
13        <Dialog.Trigger className="py-3 px-4 bg-violet-500 hover:bg-violet-600 text-white rounded">
14          <MagnifyingGlassPlus size={24}/>
15          Crie um Post
16        </Dialog.Trigger>
17      </div>
18    </div>
19  )
20 }
21
```

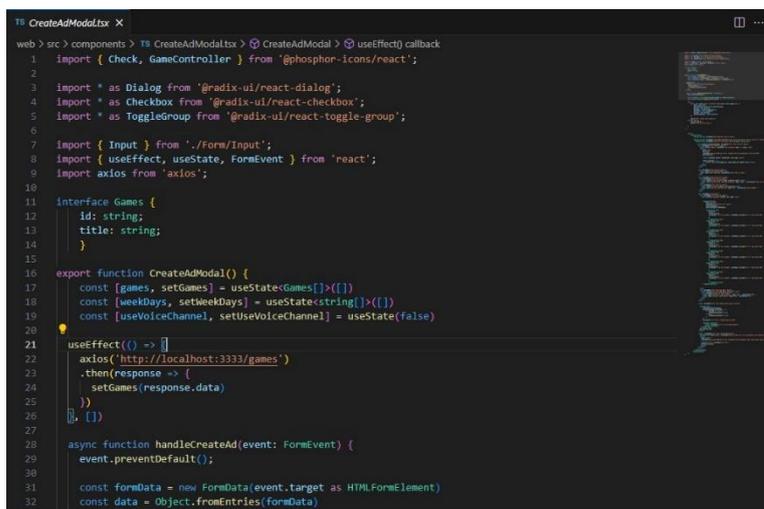
Fonte: Elaborado pelos próprios autores, 2023.

A código abaixo está relacionado ao desenvolvimento em React, denominado CreateAdModal, que desempenha um papel crucial na facilitação da criação de publicações relacionadas a jogos pelos utilizadores. Este componente importa bibliotecas e componentes essenciais, além de gerenciar estados para armazenar informações cruciais, como a lista de jogos disponíveis, os dias da semana selecionados e as preferências de chat de voz dos utilizadores.

Uma chamada à API é realizada para obter a lista de jogos, e uma função `handleCreateAd` é implementada para processar o formulário submetido, enviando dados para a API e exibindo alertas de sucesso ou erro. A interface renderizada pelo código oferece aos utilizadores campos para seleção de jogos, inserção de informações de identificação, especificação de dias da semana e horários, bem como uma opção para configurar o chat de voz.

Em suma, este componente React desempenha um papel vital na interação dos utilizadores com a plataforma, permitindo-lhes criar e compartilhar publicações relacionadas a jogos de maneira eficaz.

Figura 3 – Print 2



```
web > src > components > T8 CreateAdModal.tsx > CreateAdModal > useEffect() callback
1  import { Check, GameController } from '@phosphor-icons/react';
2
3  import * as Dialog from '@radix-ui/react-dialog';
4  import * as Checkbox from '@radix-ui/react-checkbox';
5  import * as ToggleGroup from '@radix-ui/react-toggle-group';
6
7  import { Input } from './Form/Input';
8  import { useEffect, useState, FormEvent } from 'react';
9  import axios from 'axios';
10
11 interface Games {
12   id: string;
13   title: string;
14 }
15
16 export function CreateAdModal() {
17   const [games, setGames] = useState<Games[]>([])
18   const [weekDays, setWeekDays] = useState<string[]>([])
19   const [useVoiceChannel, setUseVoiceChannel] = useState(false)
20
21   useEffect(() => {
22     axios('http://localhost:3333/games')
23       .then(response => {
24         setGames(response.data)
25       })
26     , []
27   )
28
29   async function handleCreateAd(event: FormEvent) {
30     event.preventDefault();
31
32     const formData = new FormData(event.target as HTMLFormElement)
33     const data = Object.fromEntries(formData)
```

Fonte: Elaborado pelos próprios autores, 2023.

No contexto deste artigo científico, apresentou-se o componente React denominado `GameBanner`, uma construção essencial para a exibição dinâmica de banners de jogos em aplicações interativas. Este componente, intrinsecamente versátil, aceita três propriedades fundamentais: `bannerUrls`, que recebe uma URL para a imagem do banner do jogo; `title`, uma string que encapsula o título do jogo em questão; e `adsCount`, um valor numérico inteiro que quantifica o número de anúncios associados ao jogo.

No cerne do componente, um elemento HTML de âncora (`<a>`) envolve todo o conteúdo do banner, conferindo-lhe funcionalidade de link. Dentro deste elemento, a imagem do banner é renderizada, com a URL definida dinamicamente pela propriedade `bannerUrls`.

Justaposta à imagem, uma sobreposição informativa, criada através de um elemento `<div>`, é posicionada na base do banner, exibindo o título do jogo em negrito, adquirido da propriedade `title`, e o número de anúncios correspondentes, extraído da propriedade `adsCount`. O `GameBanner` emerge como uma ferramenta altamente adaptável e imprescindível para a representação visual e textual de informações sobre jogos, encontrando aplicabilidade em contextos relacionados a jogos e entretenimento.

Este componente ilustra a abordagem modular e flexível inerente ao desenvolvimento de interfaces React, permitindo que os desenvolvedores integrem e personalizem facilmente banners de jogo em seus aplicativos. Além disso, o GameBanner desempenha um papel crucial na experiência do usuário, proporcionando uma apresentação visual envolvente e informativa de informações relacionadas a jogos, tornando-o uma adição valiosa a aplicações de entretenimento e jogos.

Este artigo científico contextualiza o componente dentro do cenário mais amplo de desenvolvimento de interfaces de usuário dinâmicas e interativas, destacando sua utilidade e versatilidade na representação eficaz de dados relacionados a jogos.

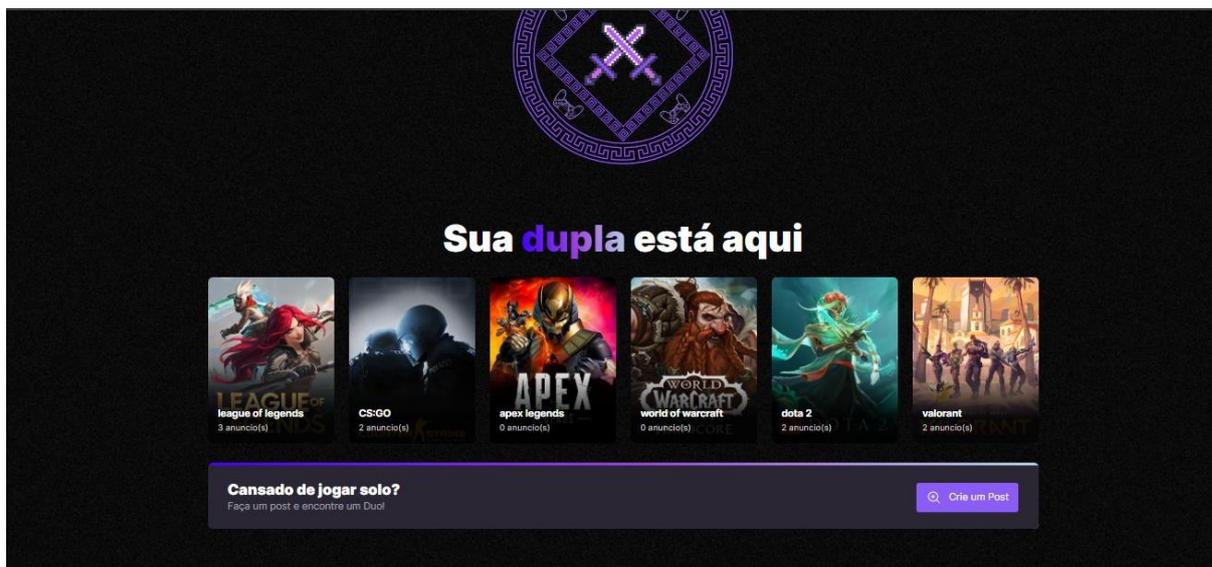
Figura 4 – Print 3

```
TS GameBanner.tsx X
web > src > components > TS GameBanner.tsx > GameBanner
1 interface GameBannerProps {
2   bannerUrls: string;
3   title: string;
4   adsCount: number;
5 }
6
7 export function GameBanner(props: GameBannerProps) {
8   return (
9     <a href="" className="relative rounded-lg overflow-hidden">
10      <img src={props.bannerUrls} alt="" />
11
12      <div className="w-full pt-16 pb-4 px-4 bg-game-gradient absolute bottom-0 left-0 right-0">
13        <strong className="font-bold text-white block">{props.title}</strong>
14        <span className="text-zinc-300 text-sm block">{props.adsCount} anúncio(s)</span>
15      </div>
16    </a>
17  )
18 }
```

Fonte: Elaborado pelos próprios autores, 2023.

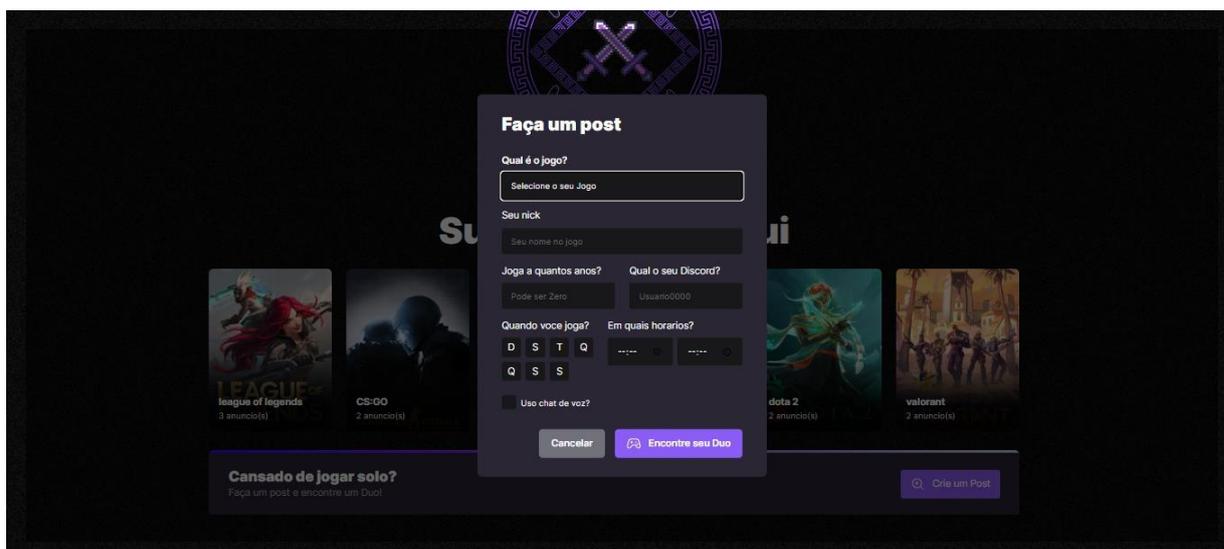
E assim ficou finalizada a parte web após desenvolvida:

Figura 5 – Print 4



Fonte: Elaborado pelos próprios autores, 2023.

Figura 6 – Print 5



Fonte: Elaborado pelos próprios autores, 2023.

4.4 Desenvolvimento do Mobile

Neste módulo, será apresentado o desenvolvimento completo da seção mobile do projeto. Esta abordagem permite uma separar o processo de desenvolvimento, que foi subdividido em etapas distintas por meio da linguagem React. Essa técnica é empregada com o propósito de promover uma organização eficiente do código-fonte e facilitar a detecção de eventuais erros no futuro. Isso significa que, em caso de problemas, os desenvolvedores terão maior facilidade para identificar e corrigir as falhas.

Figura 7 – Print 6



Fonte: Elaborado pelos próprios autores, 2023.

Esta seção refere-se à página inicial da aplicação móvel, que desempenha um papel fundamental na experiência do usuário. É responsável por apresentar a interface inicial da aplicação, na qual os usuários podem iniciar sua busca por outros jogadores.

Nesta etapa, são incorporados elementos de identidade visual, como nosso logotipo, tipo de fonte e plano de fundo, contribuindo para uma experiência coesa e reconhecível.

Figura 8 – Print 7

```
TS index.tsx ...Home U X TS style.ts U TS index.tsx ...GameCard U
src > screens > Home > TS index.tsx > Home
1 import { View, Image, FlatList } from 'react-native';
2
3
4 import logoImg from '../assets/logo-teste.png';
5 import { GameCard } from '../components/GameCard';
6
7 import { Heading } from '../components/Heading';
8
9 import { GAMES } from '../utils/games';
10 import { styles } from './styles';
11
12
13 export function Home() {
14   return (
15     <View style={styles.container}>
16       <Image
17         source={logoImg}
18         style={styles.logo}
19       />
20
21       <Heading
22         title="Encontre sua dupla!"
23         subtitle="Selecione seu jogo"
24       />
25
26       <FlatList
27         data={GAMES}
28         keyExtractor={item => item.id}
29         renderItem={({ item }) => (
30           <GameCard
31             data={item}
32           />
33         )
34       />
35     )
36   }
37 }
```

Fonte: Elaborado pelos próprios autores, 2023.

Esta seção aborda o processo de estilização da página inicial, enfocando a importância da estética. No início da aplicação, empregou-se uma técnica específica para melhorar a visibilidade dos limites da lista de busca de jogos, facilitando a compreensão de onde começa e termina essa funcionalidade.

Figura 9 – Print 8

```
TS index.tsx U TS style.ts U X
src > screens > Home > TS style.ts > ...
1 import { StyleSheet } from 'react-native';
2
3 export const styles = StyleSheet.create({
4   container: {
5     flex: 1,
6     alignItems: 'center',
7   },
8   logo: {
9     width: 214,
10    height: 128,
11    marginTop: 74,
12    marginBottom: 48
13  },
14  contentList: {
15    paddingLeft: 32,
16    paddingRight: 64,
17  }
18 });
19 });
```

Fonte: Elaborado pelos próprios autores, 2023.

A presente pesquisa explora a utilização do Prisma como Object-Relational Mapping (ORM) em um servidor de API desenvolvido com o framework Express.js. O foco primordial do servidor é a administração de informações relacionadas a jogos e anúncios. A imagem fornece uma abordagem do código implementado os

tópicos que englobam desde as importações de bibliotecas até a configuração do Express, a integração do Prisma, a definição das rotas da API e a inicialização do servidor.

Esta investigação evidencia como a combinação entre o Prisma e o Express.js proporciona uma implementação eficaz de operações Create, Read, Update, Delete (CRUD) para a manipulação de dados, contribuindo significativamente para a simplificação e eficiência no desenvolvimento de APIs.

Figura 10 – Print 9

```
server > src > TS server.ts > ...
1 import express from 'express'
2 import cors from 'cors'
3
4 import { PrismaClient } from '@prisma/client';
5 import { convertHourStringToMinutes } from './utils/convert-hour-string-to-minutes';
6 import { convertMinutesToHourString } from './utils/convert-minutes-to-hour-string';
7
8 const app = express()
9
10 app.use(express.json())
11 app.use(cors())
12
13 const prisma = new PrismaClient({
14   log: ['query']
15 })
16
17 app.get('/games', async (request, response) => {
18   const games = await prisma.game.findMany({
19     include: {
20       _count: {
21         select: {
22           ads: true,
23         }
24       }
25     }
26   })
27
28   return response.json(games);
29 });
30
31 app.post('/games/:id/ads', async (request, response) => {
32   const gameId = request.params.id;
```

Fonte: Elaborado pelos próprios autores, 2023.

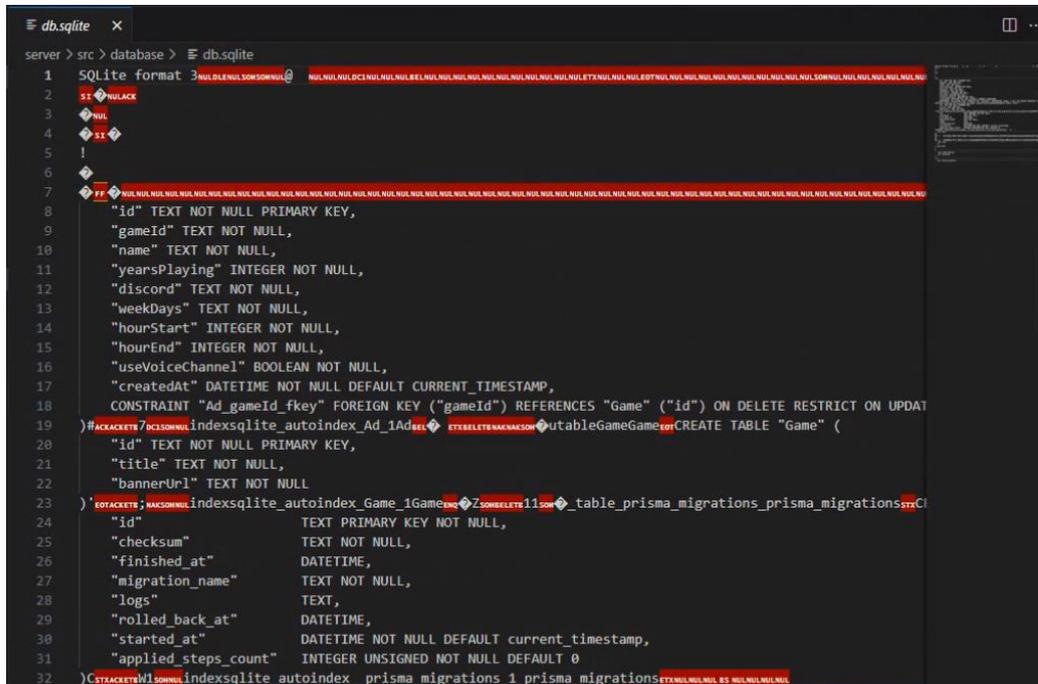
Figura 11 – Print 10

```
server > src > TS server.ts > app.get('/games') callback > @ games > include > _count > select
34 const gameId = request.params.id;
35 const body = any = request.body;
36
37 const ad = await prisma.ad.create({
38   data: {
39     gameId,
40     name: body.name,
41     yearsPlaying: body.yearsPlaying,
42     discord: body.discord,
43     weekDays: body.weekDays.join(','),
44     hourStart: convertHourStringToMinutes(body.hourStart),
45     hourEnd: convertHourStringToMinutes(body.hourEnd),
46     useVoiceChannel: body.useVoiceChannel
47   }
48 })
49
50 return response.status(201).json(ad);
51
52 app.get('/games/:id/ads', async (request, response) => {
53   const gameId = request.params.id;
54
55   const ads = await prisma.ad.findMany({
56     select: {
57       id: true,
58       name: true,
59       weekDays: true,
60       useVoiceChannel: true,
61       yearsPlaying: true,
62       hourStart: true,
63       hourEnd: true,
64     },
65   });
```

Fonte: Elaborado pelos próprios autores, 2023.

A figura apresentada abaixo ilustra a estrutura do SQLite, onde tabelas são criadas para armazenar todos os posts gerados.

Figura 12 – Print 11

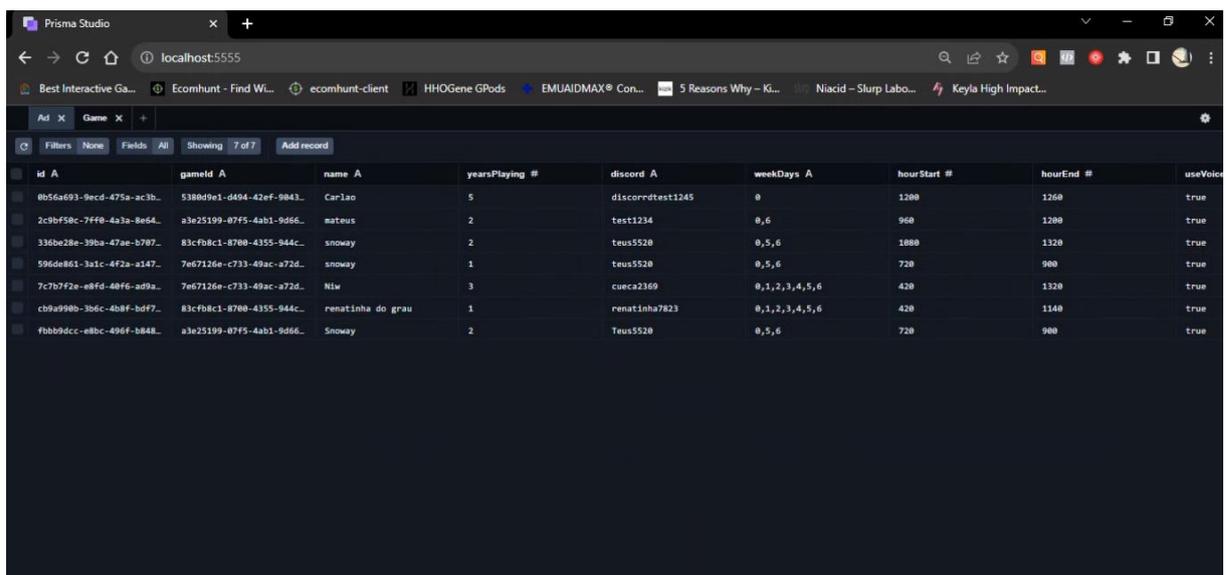


```
server > src > database > db.sqlite
1 SQLite format
2
3
4
5
6
7
8 "id" TEXT NOT NULL PRIMARY KEY,
9 "gameId" TEXT NOT NULL,
10 "name" TEXT NOT NULL,
11 "yearsPlaying" INTEGER NOT NULL,
12 "discord" TEXT NOT NULL,
13 "weekDays" TEXT NOT NULL,
14 "hourStart" INTEGER NOT NULL,
15 "hourEnd" INTEGER NOT NULL,
16 "useVoiceChannel" BOOLEAN NOT NULL,
17 "createdAt" DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
18 CONSTRAINT "Ad_gameId_fkey" FOREIGN KEY ("gameId") REFERENCES "Game" ("id") ON DELETE RESTRICT ON UPDA
19 #CREATE TABLE "Game" (
20 "id" TEXT NOT NULL PRIMARY KEY,
21 "title" TEXT NOT NULL,
22 "bannerUrl" TEXT NOT NULL
23 )
24 "id"
25 TEXT PRIMARY KEY NOT NULL,
26 "checksum"
27 TEXT NOT NULL,
28 "finished_at"
29 DATETIME,
30 "migration_name"
31 TEXT NOT NULL,
32 "logs"
33 TEXT,
34 "rolled_back_at"
35 DATETIME,
36 "started_at"
37 DATETIME NOT NULL DEFAULT current_timestamp,
38 "applied_steps_count"
39 INTEGER UNSIGNED NOT NULL DEFAULT 0
40 )
41 #CREATE TABLE "prisma_migrations" (
42 "id" TEXT NOT NULL PRIMARY KEY,
43 "checksum" TEXT NOT NULL,
44 "finished_at" DATETIME,
45 "migration_name" TEXT NOT NULL,
46 "logs" TEXT,
47 "rolled_back_at" DATETIME,
48 "started_at" DATETIME NOT NULL DEFAULT current_timestamp,
49 "applied_steps_count" INTEGER UNSIGNED NOT NULL DEFAULT 0
50 )
```

Fonte: Elaborado pelos próprios autores, 2023.

E essa é a representação dela após ter sido criada:

Figura 13 – Print 12

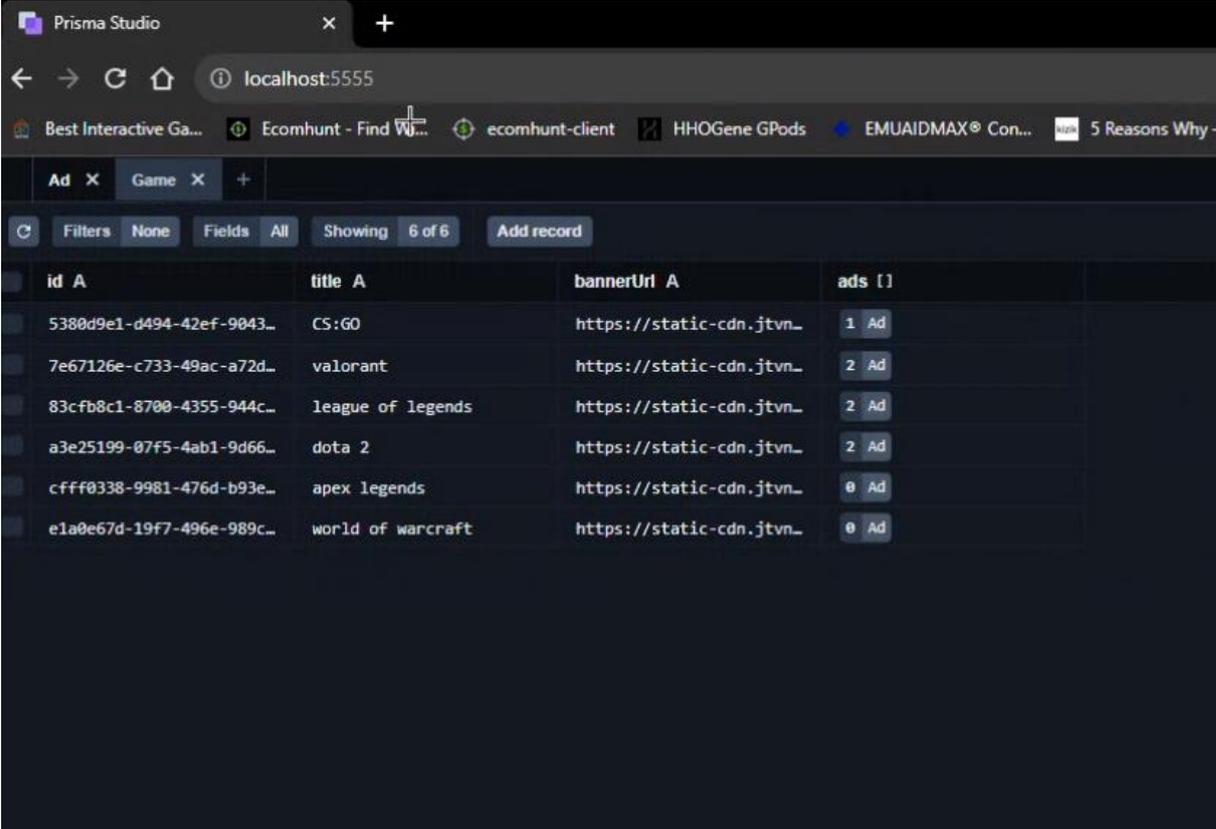


id	gameId	name	yearsPlaying	discord	weekDays	hourStart	hourEnd	useVoice
8b56a693-9ecd-475a-ac3b...	5380d9e1-d494-42ef-9843...	Car1ao	5	discorrdtest1245	0	1200	1260	true
2c7bf58c-7ff8-4a3a-8e64...	a3e25199-07f5-4a01-9d66...	matus	2	test1234	0,6	960	1200	true
336be28e-39ba-47ae-b787...	83cfb8c1-8700-4355-944c...	snoway	2	teus5528	0,5,6	1880	1320	true
596de861-3a1c-4f2a-a147...	7e67126e-c731-49ac-a72d...	snoway	1	teus5528	0,5,6	720	900	true
7c7b7f2e-e8fd-40f6-ad9a...	7e67126e-c733-49ac-a72d...	Niw	3	cueca2369	0,1,2,3,4,5,6	420	1320	true
cb9a990b-3b6c-4b8f-bdf7...	83cfb8c1-8700-4355-944c...	renatinha do grau	1	renatinha7823	0,1,2,3,4,5,6	420	1140	true
fbb9d4cc-e8bc-496f-b848...	a3e25199-07f5-4a01-9d66...	Snoway	2	Teus5528	0,5,6	720	900	true

Fonte: Elaborado pelos próprios autores, 2023.

Esta visualização exibe as tabelas que armazenam informações de todos os jogos cadastrados.

Figura 14 – Print 13



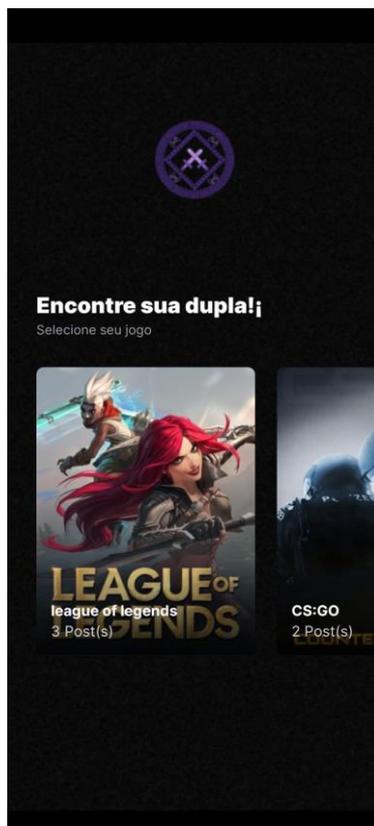
id A	title A	bannerUrl A	ads []
5380d9e1-d494-42ef-9043...	CS:GO	https://static-cdn.jtvn...	1 Ad
7e67126e-c733-49ac-a72d...	valorant	https://static-cdn.jtvn...	2 Ad
83cfb8c1-8700-4355-944c...	league of legends	https://static-cdn.jtvn...	2 Ad
a3e25199-07f5-4ab1-9d66...	dota 2	https://static-cdn.jtvn...	2 Ad
cfff0338-9981-476d-b93e...	apex legends	https://static-cdn.jtvn...	0 Ad
e1a0e67d-19f7-496e-989c...	world of warcraft	https://static-cdn.jtvn...	0 Ad

Fonte: Elaborado pelos próprios autores, 2023.

Abordou-se a implementação de um servidor de API voltado para a gestão de dados de jogos e anúncios. Utilizou-se o Prisma como um ORM para interagir com um banco de dados SQLite, enquanto o framework Express.js facilitou o desenvolvimento das rotas da API.

Foram explorados aspectos que incluem importações de bibliotecas, configurações do Express, integração do Prisma e definições de rotas. Destacou-se a finalização bem-sucedida do desenvolvimento, enfatizando como a combinação do Prisma e do Express.js simplifica a criação de APIs eficazes para a manipulação de dados. E esse é o projeto mobile desenvolvido:

Figura 15 – Print 14



Fonte: Elaborado pelos próprios autores, 2023.

5 Conclusão

Nesta seção, será apresentado o resultado do desenvolvimento do software e os testes realizados na plataforma.

O advento da tecnologia digital trouxe consigo transformações notáveis em quase todos os aspectos da vida cotidiana. Em particular, a maneira como os indivíduos se conectam e interagem sofreu mudanças profundas, impulsionadas principalmente pela ascensão dos aplicativos de relacionamento. Estas plataformas, ao redefinirem as nuances das interações humanas, demonstraram o poder da tecnologia em moldar e enriquecer as conexões interpessoais, tornando-se ferramentas indispensáveis na sociedade contemporânea.

Dentro deste vasto domínio de interações mediadas pela tecnologia, este estudo se concentrou especificamente na esfera dos jogos online. Com a crescente popularidade e engajamento em jogos virtuais, identificou-se uma necessidade latente de plataformas que permitam aos jogadores não apenas competir, mas também compartilhar e discutir suas experiências. Inspirado por aplicativos de relacionamento bem-sucedidos, o trabalho propôs um software inovador destinado a preencher essa lacuna, oferecendo um espaço dedicado para a comunidade de jogadores se conectar de maneira mais significativa.

Em conclusão, à medida que a fronteira entre o real e o virtual continua a se dissipar, é imperativo reconhecer e abordar as demandas emergentes de socialização digital. Este estudo contribui para essa discussão, destacando a importância

e o potencial de soluções gamificadas que não apenas entretêm, mas também facilitam conexões humanas profundas e significativas no ambiente digital.

Assim, o artigo apresenta uma proposta inovadora para melhorar a socialização entre jogadores em ambientes online, utilizando a gamificação como ferramenta central. Além disso, como trabalhos futuros, propõe-se expandir essa plataforma para abranger outras formas de interação digital e explorar seu potencial em diferentes contextos e comunidades, visando aprimorar ainda mais a experiência de conexão e compartilhamento no mundo digital.

Referências

DE ANDRADE, Ana Paula. Conheça o React, biblioteca para desenvolvimento Web. **Treinaweb**, 2020. Disponível em: <<https://www.treinaweb.com.br/blog/conheca-o-react-biblioteca-para-desenvolvimento-web#:~:text=O%20React%20lan%C3%A7ado%20em%202013,e%20com%20uma%20%C3%B3tima%20performance>>. Acesso em: 11 set. 2023.

DURKHEIM, Émile. A educação, sua natureza e seu papel. In: DURKHEIM, Émile. **Educação e Sociologia**. 10. ed. São Paulo: Melhoramentos, 1975. Disponível em: <https://aedmoodle.ufpa.br/pluginfile.php/310886/mod_glossary/attachment/5113/DURKHEIM%20-%20Educa%C3%A7%C3%A3o%20e%20Sociologia%20-%20Cap%201.pdf>. Acesso em: 17 ago. 2023.

FERNANDES, Jorge. O uso de softwares na resolução de problemas: uma experiência realizada na UFAL/CAMPUS Arapiraca. **Conedu**. Disponível em: <https://www.editorarealize.com.br/editora/anais/conedu/2017/TRABA-LHO_EV073_MD1_SA13_ID6846_16102017233221.pdf>. Acesso em: 14 jul. 2023.

MOURA, Diego. Event Sourcing – Big Picture. **Medium**, maio de 2020. Disponível em: <https://medium.com/@diego_moura/event-sourcing-big-picture-86cc36a2c851>. Acesso em: 21 ago. 2023.

PEREIRA, Gabriel. O que é HTML 5? Saiba quem criou, para que serve e mais curiosidades. **Techtudo**, 2023. Disponível em: <<https://www.techtudo.com.br/listas/2023/06/o-que-e-html-5-edsoftwares.ghtml>>. Acesso em: 14 ago. 2023.

SAVOIA, Mariângela Gentil. **Psicologia social**. São Paulo: McGraw-Hill, 1989.

SMANIOTO, Daniel. DRY, YAGNI, KISS? Princípios de design de software que todo desenvolvedor deveria seguir. **Software Design Principles**, 2020. Disponível em: <<https://danielsmanioto.com/blog/2020/01/18/dry-yagni-kiss-principios-de-design-de-software-que-todo-desenvolvedor-deveria-seguir/>>. Acesso em: 10 set. 2023.

STAIR, Ralph M.; REYNOLDS, George W. **Princípios de sistemas de informação**. 9. ed. São Paulo: Cengage Learning, 2011.

STREY, Marlene Neves. **Psicologia Social Contemporânea**. 7. ed. Rio de Janeiro: Vozes, 2002.

TAVARES, Fábio Roberto. **Ética, política e sociedade**. Indial: Uniasselvi, 2013. Disponível em: < <https://www.uniasselvi.com.br/extranet/layout/request/trilha/materiais/livro/livro.php?codigo=13955>>. Acesso em: 18 ago. 2023.

VIEIRA, Danielle. SQLite: o que é, como funciona e qual é a diferença entre o MySQL. **HostGator**, 2021. Disponível em: <<https://www.hostgator.com.br/blog/sqlite-o-que-e-como-funciona-e-qual-e-a-diferenca-entre-o-mysql/>>. Acesso em: 29 ago. 2023.