



AGRUPAMENTO DE ESCOLAS  
**MANUEL TEIXEIRA GOMES**  
PORTIMÃO

**Escola Secundaria Manuel Teixeira Gomes**

Técnico de Gestão e Programação de Sistemas Informáticos

Professor Orientador: Tiago Lopes

## Relatório da Prova de Aptidão Profissional

TEAM GGDEV – A Plataforma de colaboração entre DEVs



Miguel Loução Brandão Nº17 12ºF

Portimão, 3 de junho de 2024

2023/2024

## Índice

Índice .....	2
Índice de Imagens.....	4
I. Agradecimentos.....	6
II. Introdução .....	7
III. Fundamento .....	8
IV. Metodologia.....	9
Planeamento e Organização .....	9
Desenvolvimento Iterativo .....	9
Conceção e Prototipagem:.....	9
Desenvolvimento Incremental:.....	9
Testes e Validação: .....	9
Ferramentas e Tecnologias.....	10
Autonomia e Aprendizagem Autodirigida.....	10
Conclusões finais sobre a metodologia.....	10
V. Softwares utilizados.....	11
Node.js.....	11
Express.....	11
Electron.....	12
Scss/Sass .....	13
TypeScript.....	13
MongoDB.....	14
Photoshop e Krita .....	15
Visual Studio Code (c) e Notepad++ .....	15
WAMP .....	16
Google Docs.....	17
GitHub e GitHub Desktop .....	17
VII. Calendarização .....	19
VIII. Diagrama de Fluxo de Dados.....	20
IX. Apresentação e explicação de código .....	22
X. Manual do utilizador.....	30
DevMesh .....	30
Crafters Fund.....	37
BrainBurst .....	45
XI. Análise crítica da execução do projeto.....	50
Sucessos do Projeto .....	50

Inovação e Criatividade:.....	50
Autonomia na Resolução de Problemas:.....	50
Qualidade Técnica: .....	50
Desafios Enfrentados .....	50
Gestão de Tempo: .....	50
Lições Aprendidas .....	50
Importância do Planeamento e Flexibilidade: .....	50
Valor da Pesquisa e Aprendizagem Contínua: .....	51
Comunicação e Documentação: .....	<b>Erro! Marcador não definido.</b>
Exposição Final .....	51
Conclusão .....	52
Realizações e Impacto.....	52
Desafios e Aprendizados.....	52
Impacto Futuro e Contribuição .....	52
Conclusão Final.....	53
Fontes de pesquisa de informação .....	54
W3Schools: .....	54
Assistentes Virtuais/IA .....	54

## Índice de Imagens

FIGURA 1 NODEJS .....	11
FIGURA 2 EXPRESS .....	12
FIGURA 3 ELECTRON.....	12
FIGURA 4 SASS .....	13
FIGURA 5 TYPESCRIPT.....	14
FIGURA 6 MONGODB.....	14
FIGURA 7 KRITA    FIGURA 8 PHOTOSHOP .....	15
FIGURA 9 VS CODE    FIGURA 10 NOTEPAD ++ .....	16
FIGURA 11 WAMP.....	16
FIGURA 12 GOOGLE DOCS.....	17
FIGURA 13 GITHUB DESKTOP    FIGURA 14 GITHUB.....	18
FIGURA 15 MAPA DE GANT DESENVOLVIDO NO BITRIX24.....	19
FIGURA 16 DIAGRAMA DE NIVEL 0 - PROJETO PAP.....	20
FIGURA 17 DIAGRAMA DE NIVEL 1 - PROJETO PAP.....	20
FIGURA 18 DIAGRAMA DE NIVEL 2 - ACESSO PLAYER .....	21
FIGURA 19 CÓDIGO CRAFTERSFUND (IMPORTAÇÃO DAS BIBLIOTECAS) .....	22
FIGURA 20 CÓDIGO CRAFTERSFUND (UTILIZAÇÃO DAS PASTAS, INICIAÇÃO DO MONGODB E CRIAÇÃO DO SISTEMA DE SEÇÃO) .....	23
FIGURA 21 CODIGO DO CRAFTERS FUND (CRIAÇÃO DO LOGIN E REGISTO) .....	24
FIGURA 22 CODIGO CRAFTERS FUND (FUNCIONAMENTO DO CARREGAMENTO DE UMA PAGINA, NO CASO AA PAGINA DE TICKETS NA DASHBOARD).....	25
FIGURA 23 CODIGO CRAFTERS FUND (INICIALIZAÇÃO DO SERVIDOR E RENCAMINHANDO DE ERROS).....	25
FIGURA 24 ESTRUTURA DE PASTAS DAS PLATAFORMAS.....	26
FIGURA 25 FICHERIO PACKAGE.JSON O QUAL É UTILIZADO PARA A CONFIGURAÇÃO BASE DO NODEJS.....	27
FIGURA 26 PÁGINA EJS DE EXEMPLO PARTE 1.....	28
FIGURA 27 PÁGINA EJS DE EXEMPLO PARTE 2.....	29
FIGURA 28 PÁGINA EJS DE EXEMPLO PARTE 3.....	29
FIGURA 29 PAGINA DE LOGIN   DEV MESH .....	30
FIGURA 30 FORMS DE REGISTO   DEV MESH .....	31
FIGURA 31 PAGINA HOME   DEV MESH .....	32
FIGURA 32 FORMS PARA POSTAR UMA QUESTÃO   DEV MESH.....	34
FIGURA 33 PAGINA DE FORUM .....	35
FIGURA 34 PAGINA DE PERFIS.....	35
FIGURA 35 PAGINA HOME PARTE 1   CRAFTERS FUND.....	37
FIGURA 36 PAGINA HOME PARTE 2   CRAFTERS FUND.....	37
FIGURA 37 LISTA DE PROJETOS/JOGOS   CRAFTERS FUND .....	38
FIGURA 38 PAGINA DE UM PROJETO .....	39
FIGURA 39 PAGINA DE CRIAÇÃO DO PERFIL.....	41

FIGURA 40 PAGINA DE PERFIL DE EXEMPLO   CRAFTERS FUND.....	42
FIGURA 41 FORMS DE SUPORTE DE UM PROJETO   CRAFTERS FUND .....	43
FIGURA 42 TICKET CRIADO   CRAFTERS FUND.....	43
FIGURA 43 PAGINA DO TICKET   CRAFTERS FUND.....	44
FIGURA 44 POSSIBILIDADE DE RESPONDER AO TICKET   CRAFTERS FUND.....	45
FIGURA 45 BOTAO PARA O DESENVOLVEDOR FECHAR O TICKET   CRAFTERS FUND .....	45
FIGURA 46 PAGINA HOME   BRAINBURST.....	46
FIGURA 47 PAGINA DE ESCOLHA DO TIPO DE MINIJOGO.....	47
FIGURA 48 MENU DE ESCOLHA DO MINIJOGO DE PROGRAMAÇÃO   BRAINBURST .....	48
FIGURA 49 LISTA DE NIVEIS DO CODE MASTER   BRAINBURST .....	48
FIGURA 50 PAGINA DO PRIMEIRO EXERCICIO .....	49

## **I. Agradecimentos**

Neste momento de concretização, olho para trás e observo o percurso que foi o desenvolvimento das plataformas BrainBurst, DevMesh e CraftersFund, projetos que foram inteiramente idealizados e desenvolvidos por mim, mas que contaram com o apoio emocional e o incentivo de pessoas sem as quais essa jornada teria sido consideravelmente mais árdua. Quero expressar a minha profunda gratidão à minha família, que sempre acreditou em mim e no meu potencial.

Aos meus amigos, pelo suporte incondicional, pelas palavras de ânimo e pelos momentos de descontração que me ajudaram a manter o foco e a motivação.

E a todos que, de alguma forma, contribuíram com palavras de incentivo, críticas construtivas e feedbacks valiosos, onde neste ponto quero destacar o professor e diretor de curso Tiago Lopes e o meu orientador de estágio Roberto Vieira, os quais foram fundamentais para o aperfeiçoamento contínuo dos projetos.

Todos os erros e imprecisões que possam existir na plataforma é da minha responsabilidade, e encaro cada um deles como uma oportunidade de aprendizagem e crescimento. A todos vós, o meu sincero obrigado.

## II. Introdução

Este relatório tem como propósito principal detalhar o processo de criação e evolução das plataformas BrainBurst, DevMesh e CraftersFund, cada uma delas concebida para atender a diferentes necessidades dentro da comunidade de desenvolvedores.

O BrainBurst é uma inovadora aplicação e página web que integra um serviço de exercícios de programação (contando com 100 exercícios que podem ser praticados em Python, Java, C#, C e Visual Basic) e minijogos que estimulam o exercício constante do cérebro.

O DevMesh é uma rede social destinada a todos os tipos de desenvolvedores, onde é possível encontrar trabalhos freelancers, fazer posts diários e fazer perguntas relacionadas a erros de código ou outros temas pertinentes.

Por último, o CraftersFund é uma plataforma voltada para desenvolvedores de jogos, especialmente indies, onde podem compartilhar os jogos que estão a desenvolver e buscar financiamento. Esta plataforma inclui um sistema de tickets para fazer questões aos criadores dos jogos e uma funcionalidade de wishlist.

Através desta introdução, pretendo oferecer uma visão abrangente e concisa dos objetivos visados e dos conteúdos que serão explorados ao longo deste documento. Serão abordados os estágios de concepção, design e programação das aplicações, destacando as metodologias adotadas para assegurar a sua funcionalidade, facilidade de uso e atratividade para o público-alvo.

Por fim, o intuito deste documento é compartilhar os êxitos e lições aprendidas ao longo do projeto, contribuindo assim para o conhecimento coletivo na área de desenvolvimento de software e servindo de inspiração para futuras “empreitadas” tecnológicas.

### III. Fundamento

Tal como falado anteriormente este relatório explora detalhadamente as subplataformas encontradas no projeto GGDev, cada uma criada com o intuito de resolver diferentes desafios e servir distintos segmentos da comunidade de desenvolvedores.

O BrainBurst é uma aplicação web multifacetada que não só redefine a interação online através de uma coletânea de minijogos competitivos, como também se destaca como um Ambiente de Desenvolvimento Integrado (IDE) para programação. A ideia do BrainBurst surgiu da perceção da necessidade de uma plataforma que pudesse servir simultaneamente como um meio de entretenimento e uma ferramenta de desenvolvimento, enriquecendo assim a experiência digital dos programadores.

A escolha deste tema decorreu da minha paixão por tecnologia e jogos, aliada ao interesse em responder às crescentes demandas por soluções que combinem entretenimento digital com oportunidades de aprendizagem e desenvolvimento em programação. A integração de um IDE no contexto de uma plataforma de minijogos visa não apenas entreter, mas também inspirar os utilizadores a aprimorar as suas habilidades de programação de uma forma divertida e interativa.

O DevMesh é uma rede social destinada a todos os tipos de desenvolvedores. Esta plataforma permite encontrar trabalhos freelancers, fazer posts diários e perguntas relacionadas a erros de código ou outros temas pertinentes. A ideia do DevMesh nasceu da observação de que os desenvolvedores frequentemente necessitam de um espaço dedicado para interação profissional e troca de conhecimentos. Os objetivos do DevMesh incluem facilitar conexões profissionais, criar oportunidades de trabalho e promover um ambiente de colaboração onde os desenvolvedores podem compartilhar conhecimento e ajudar uns aos outros a resolver problemas técnicos.

Por último, o CraftersFund é uma plataforma voltada para desenvolvedores de jogos, especialmente indies, onde podem compartilhar os jogos que estão a desenvolver e buscar financiamento. Esta plataforma inclui um sistema de tickets para fazer questões aos criadores dos jogos, uma funcionalidade de wishlist e um sistema de comunidades. O CraftersFund foi criado em resposta à crescente necessidade de desenvolvedores independentes terem acesso a recursos financeiros e feedback da comunidade. Os objetivos desta plataforma são proporcionar um espaço onde os desenvolvedores de jogos possam obter financiamento, interagir com potenciais apoiadores e melhorar os seus projetos com base no feedback recebido.

O público-alvo destas plataformas é amplo e diversificado. O BrainBurst atrai jovens entusiastas de tecnologia e jogos que procuram uma forma lúdica e competitiva de entretenimento, bem como programadores e estudantes de informática que desejam aprimorar as suas habilidades num ambiente inovador e estimulante. O DevMesh atende a desenvolvedores de todos os níveis que buscam oportunidades de trabalho e um espaço para troca de conhecimento. O CraftersFund é voltado para desenvolvedores de jogos indies que precisam de apoio financeiro e feedback para os seus projetos.



## **IV. Metodologia**

Na realização dos projetos, foi adotada uma abordagem metodológica centrada na autonomia e na adaptabilidade, características fundamentais para o desenvolvimento eficaz de projetos individuais em tecnologia da informação. Segue uma descrição detalhada da metodologia empregada.

### **Planeamento e Organização**

Tendo em conta a natureza singular dos projetos e a responsabilidade individual, o planeamento inicial e a organização foram cruciais. No qual desenvolvi um cronograma detalhado, identificando fases de desenvolvimento, objetivos a alcançar e prazos para cada etapa. A utilização de um mapa de Gantt permitiu uma visão clara dos projetos, facilitando a gestão do tempo e a priorização de tarefas.

### **Desenvolvimento Iterativo**

Adotei uma abordagem iterativa ao desenvolvimento, permitindo flexibilidade e ajustes contínuos aos projetos. Esta estratégia foi dividida em várias etapas:

#### **Conceção e Prototipagem:**

Inicialmente, foram concebidos esboços e protótipos das interfaces de utilizador e da arquitetura geral do sistema, utilizando ferramentas como papel e lápis, e posteriormente softwares de design gráfico. Esta fase permitiu a visualização e refinamento das ideias antes da implementação.

#### **Desenvolvimento Incremental:**

Com base nos protótipos, procedeu-se ao desenvolvimento incremental do projeto, começando com as funcionalidades básicas e progredindo para os componentes mais complexos. Esta abordagem facilitou a gestão da complexidade e permitiu focar na qualidade e na eficiência do código.

#### **Testes e Validação:**

Paralelamente ao desenvolvimento, foram realizados testes contínuos, incluindo testes unitários para validar a lógica de programação e testes de usabilidade para assegurar uma experiência do utilizador coesa. A identificação precoce de falhas e a recolha de feedback foram essenciais para o aperfeiçoamento do projeto.

## **Ferramentas e Tecnologias**

A seleção criteriosa de ferramentas e tecnologias desempenhou um papel vital no desenvolvimento. A utilização de Node.js, Express, EJS e MongoDB permitiu a criação de uma aplicação robusta e escalável. A escolha do Electron para a versão desktop e a integração do SCSS/Sass e TypeScript contribuíram para um desenvolvimento eficiente e uma manutenção simplificada do código.

## **Autonomia e Aprendizagem Autodirigida**

Dada a natureza individual do projeto, a autonomia e a capacidade de aprendizagem autodirigida foram fundamentais. Enfrentar desafios técnicos exigiu pesquisa independente, experimentação e a procura de soluções inovadoras, promovendo um profundo desenvolvimento pessoal e profissional.

## **Conclusões finais sobre a metodologia**

A minha metodologia, acredito que é caracterizada pela sua adaptabilidade e foco no desenvolvimento iterativo e contínuo aprendizado, não só facilitou a realização do projeto mas também contribuiu significativamente para o enriquecimento das minhas competências técnicas e de gestão de projetos.

## V. Softwares utilizados

Este segmento do relatório destina-se a oferecer uma visão geral das tecnologias e ferramentas de software que foram fundamentais no desenvolvimento.

A escolha destas tecnologias foi orientada pela necessidade de criar uma aplicação e uma pagina web robusta, interativa e altamente personalizável. Seguidamente, apresento uma listagem do software utilizado na concepção do projeto, acompanhada de uma breve descrição de cada item e uma explicação de como cada aplicação foi empregue.

### Node.js

Node.js é um ambiente de execução para JavaScript do lado do servidor, que permite a criação de aplicações web escaláveis. No projeto, Node.js foi utilizado como a espinha dorsal para construir o servidor, gerir as conexões com a base de dados e processar as requisições HTTP, proporcionando assim a infraestrutura necessária para a minha aplicação, pagina web e os serviços de backend.



*Figura 1 NodeJS*

### Express

Express é um framework para Node.js que simplifica a criação de aplicações web e APIs. Foi utilizado para definir rotas e manipular as requisições e respostas HTTP, facilitando a implementação de funcionalidades como o sistema de autenticação de utilizadores, a lógica dos minijogos e a interação com a IDE de programação.

*Figura 2 Express*

## Electron

Electron é um framework que permite desenvolver aplicações desktop nativas usando tecnologias web como JavaScript, HTML e CSS. No projeto, o Electron foi empregue para criar uma versão desktop da minha aplicação web, oferecendo aos utilizadores uma experiência mais rica e integrada.

*Figura 3 Electron*

## SCSS/SASS

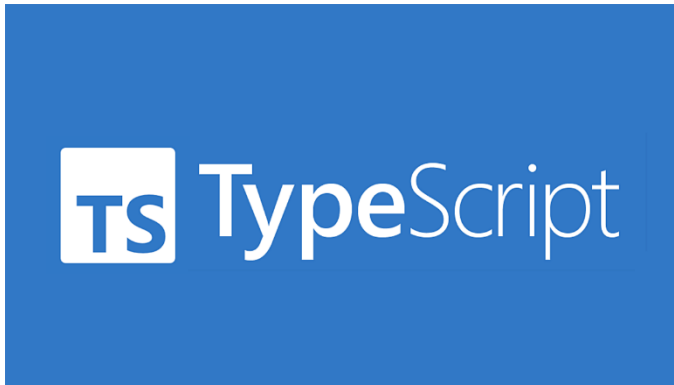
SCSS/SASS é um pré-processador CSS que facilita a escrita de estilos de forma mais dinâmica e organizada. Foi utilizado para estruturar e personalizar a aparência da aplicação, permitindo uma gestão mais eficiente dos estilos e um design mais coeso e adaptável às diversas funcionalidades da plataforma.



*Figura 4 Sass*

## TypeScript

TypeScript é uma linguagem de programação que estende JavaScript adicionando tipagem estática. No projeto, eu escolhi TypeScript para desenvolver tanto o frontend quanto o backend da aplicação, proporcionando uma maior segurança no código através da detecção de erros em tempo de compilação e melhorando a manutenibilidade e escalabilidade do projeto.



*Figura 5 TypeScript*

## **MongoDB**

MongoDB é um base de dados NoSQL orientado a documentos, escolhido pelo seu desempenho, flexibilidade e facilidade de integração com Node.js servindo como o sistema de gestão de base de dados para armazenar todas as informações dos utilizadores.



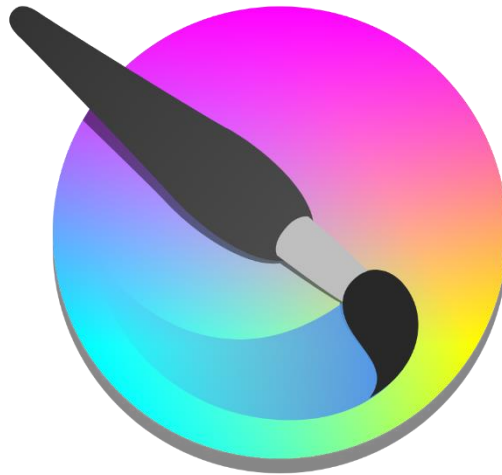
*Figura 6 MongoDB*

## Photoshop e Krita

Adobe Photoshop e Krita são poderosas ferramentas de edição de imagem. Ao longo do desenvolvimento, o Photoshop foi utilizado para o design de interfaces, ícones e elementos gráficos da aplicação, enquanto Krita, uma alternativa open-source, foi empregue na criação de arte e ilustrações mais detalhadas.



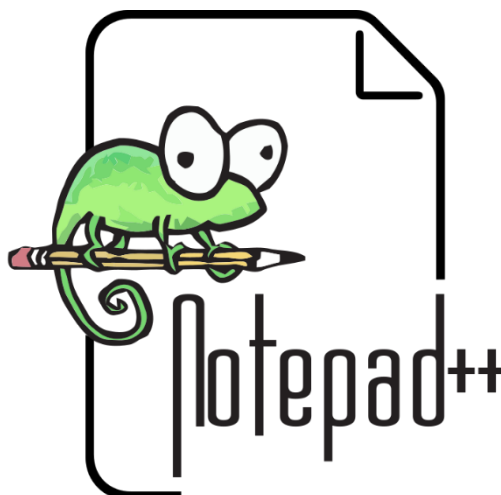
*Figura 7 Krita*



*Figura 8 PhotoShop*

## Visual Studio Code (c) e Notepad++

VSCoDe é um editor de código-fonte desenvolvido pela Microsoft, e Notepad++ é um editor de texto que suporta várias linguagens de programação. Ambos foram utilizados no desenvolvimento para escrever e editar o código. O VSCoDe, com os seus extensos plugins e integração com Git, foi primariamente usado para o desenvolvimento de funcionalidades mais complexas, enquanto Notepad++ foi utilizado para edições rápidas e revisões de código.

*Figura 9 VsCode**Figura 10 Notepad ++*

## WAMP

WAMP é um pacote de software que inclui o Apache web server, MySQL e PHP. No caso foi utilizado para configurar um ambiente de desenvolvimento local durante o tempo em que o projeto estava integrado em php, permitindo testar a aplicação web e a comunicação com o base de dados MongoDB antes do deployment num ambiente de produção.

*Figura 11 Wamp*



## Google Docs

Google Docs forneceu uma plataforma colaborativa para a documentação do projeto, permitindo a criação, edição e partilha em tempo real de documentos técnicos e planos de projeto.



*Figura 12 Google Docs*

## GitHub e GitHub Desktop

GitHub é uma plataforma de hospedagem de código para controlo de versão e colaboração, que permite aos desenvolvedores trabalhar juntos no mesmo projeto de qualquer lugar. O GitHub Desktop é uma aplicação que simplifica o uso do GitHub, oferecendo uma interface gráfica para gerir repositórios. No projeto, o GitHub foi utilizado para o versionamento do código, permitindo um controlo eficaz das diferentes versões do projeto e facilitando a colaboração entre desenvolvedores. O GitHub Desktop, por sua vez, facilitou a interação com o repositório do projeto, tornando mais acessível a tarefa de commit, push, e pull das alterações de código.

*Figura 13 Github Desktop**Figura 14 GitHub*

Cada uma dessas ferramentas desempenhou um papel vital na realização de cada subplataforma, abrangendo desde a construção e gestão do base de dados, passando pelo design gráfico e edição de imagens, até a escrita e manutenção do código, configuração do ambiente de desenvolvimento, documentação, e gestão de versões. Juntas, elas compuseram a base tecnológica que permitiu a criação de uma aplicação e uma pagina web complexa, interativa e inovadora.

VII. Calendarização

A utilização de um mapa de Gantt, é crucial para o sucesso do seu projeto por várias razões.

Primeiro, ele oferece uma representação visual clara do cronograma do projeto, incluindo a duração, início e término de cada tarefa. Isso facilita a identificação de como as tarefas se sobrepõem e as dependências entre elas, permitindo uma melhor alocação de recursos e ajustes no planeamento conforme necessário.

Segundo, ao escolher o mês como unidade de medida para a duração das tarefas, o mapa evita tornar-se demasiado extenso e mantém-se fácil de ler, o que é essencial para o acompanhamento rápido do progresso do projeto. Em suma, o mapa de Gantt abaixo é uma ferramenta fundamental para gerir eficientemente o tempo e os recursos do projeto, assegurando que este seja concluído dentro do prazo.

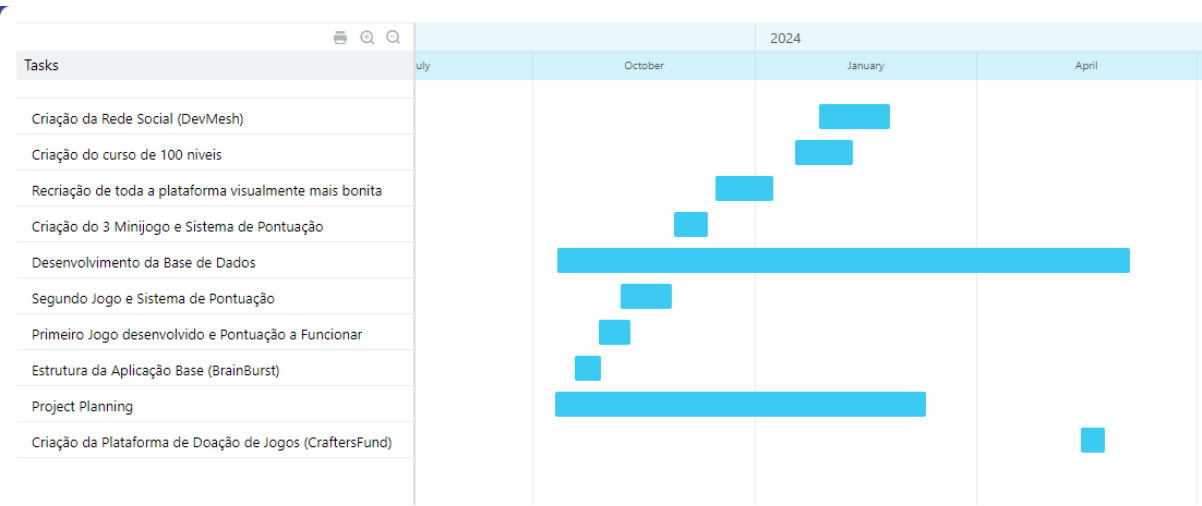


Figura 15 Mapa de Gant desenvolvido no Bitrix24

## VIII. Diagrama de Fluxo de Dados

Desenvolvi o seguinte diagrama de fluxo de dados para o Brain Burst quanto ao seu sistema de Jogos, tocando levemente no sistema de chatting presente no DevMesh (pois originalmente o sistema de chatting estava no BrainBurst), o(s) qual(is) mais se adequava(m) ao desenvolvimento de tal diagrama utilizando o website Draw.io que me deu suporte á criação das formas, setas e textos, tornando mais agil a criação de tal tarefa.



Figura 16 Diagrama de nível 0 - Projeto PAP

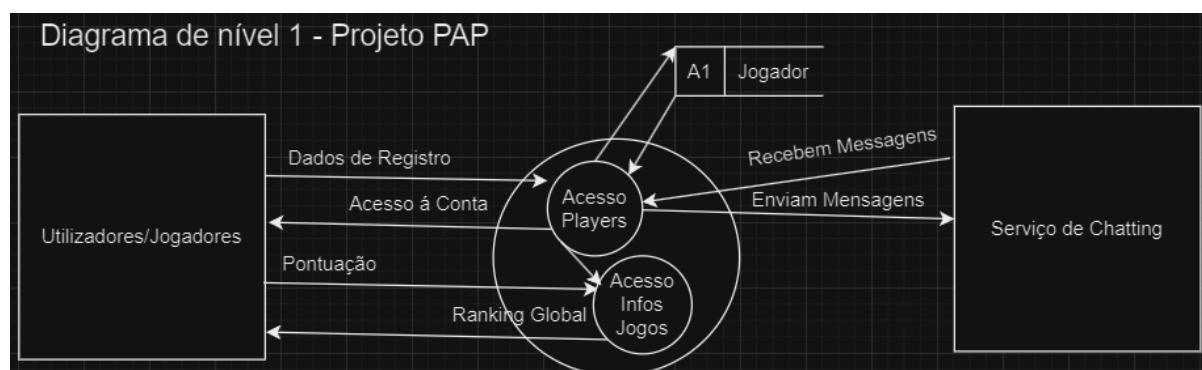


Figura 17 Diagrama de nível 1 - Projeto PAP

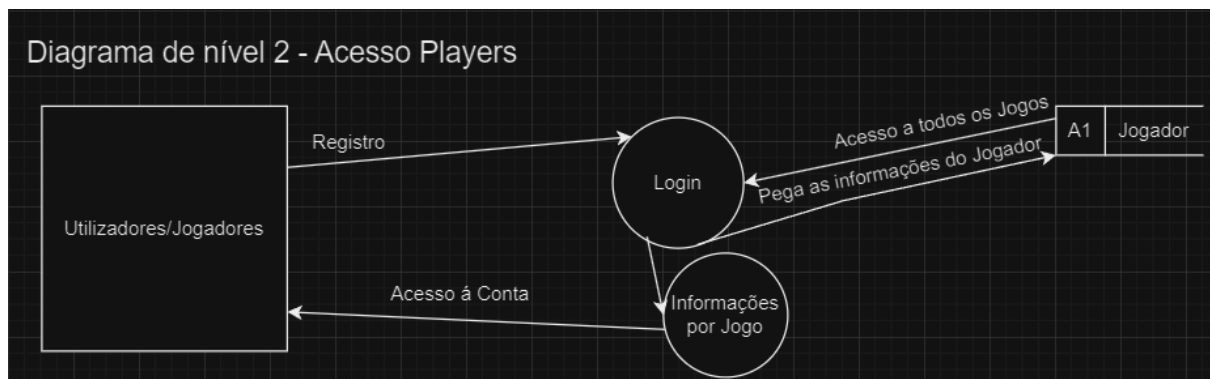


Figura 18 Diagrama de nível 2 - Acesso Player

## IX. Apresentação e explicação de código

Aqui abaixo irei mostrar o código do servidor que desenvolvi para o Crafters Fund (apenas mostrando tal por ser o mais desenvolvido e por todo as três subplataformas estarem estruturadas de igual forma), no qual explorei profundamente o ambiente Node.js. Em cada projeto, empreguei uma variedade de frameworks e bibliotecas para otimizar o desenvolvimento e a funcionalidade. Por exemplo, utilizei o Mongoose e o MongoDB como solução robusta para o armazenamento de dados, aproveitando as suas capacidades de modelagem e consulta. O Express desempenhou um papel fundamental na criação da arquitetura do servidor, simplificando rotas e a manipulação de solicitações HTTP.

Além disso, adotei o EJS como mecanismo de visualização, o que facilitou a renderização de páginas dinâmicas e a passagem eficiente de variáveis para o cliente. Essas escolhas não apenas aprimoraram a experiência do utilizador, mas também contribuíram para um código mais limpo e organizado.

Resumindo, cada componente do ambiente Node.js foi cuidadosamente selecionado e integrado para garantir um desempenho excepcional e uma experiência ao utilizador fluida em todas as aplicações desenvolvidas.

```
const express = require('express');
const bodyParser = require('body-parser');
const session = require('express-session');
const cookieParser = require('cookie-parser');
const MongoStore = require('connect-mongo');
const moment = require('moment');
const bcrypt = require("bcrypt");
const multer = require('multer');
const path = require('path');

const app = express();
const port = 5000;

require('dotenv').config();
```

Figura 19 Código CraftersFund (Importação das bibliotecas)

```
const sessionConfig = {
  secret: process.env.SECRET_SESSION_KEY,
  resave: false,
  saveUninitialized: true,
  cookie: {
    secure: process.env.NODE_ENV === 'production',
    sameSite: process.env.NODE_ENV === "production" ? 'none' : 'lax',
  },
  store: MongoStore.create({
    mongoUrl: process.env.MONGO_URI
  })
};

app.engine("html", require("ejs").renderFile);
app.set("view engine", "html");
app.use("/public", express.static(path.join(__dirname, "public")));
app.use("/uploads", express.static(path.join(__dirname, "uploads")));
app.use("/node_modules", express.static(path.join(__dirname, "node_modules")));
app.use("/src", express.static(path.join(__dirname, "src")));
app.set("views", path.join(__dirname, "views"));
app.use(bodyParser.urlencoded({ extended: true }));
app.use(express.json());
app.use(cookieParser());
app.use(session(sessionConfig));

const { MongoClient, ObjectId } = require('mongodb');
const client = new MongoClient(process.env.MONGO_URI);

async function connectDB() {
  try {
    await client.connect();
    return client.db('CraftersFund');
  } catch (err) {
    console.error("Failed to connect to MongoDB", err);
    process.exit(1);
  }
}
```

Figura 20 Código CraftersFund (Utilização das pastas, Iniciação do MongoDB e criação do sistema de sessão)

```
app.post('/loginUser', async (req, res) => {
  const { email, password } = req.body;

  try {
    const db = await connectDB();
    const collection = db.collection('users');
    const user = await collection.findOne({ email: email });

    if (user) {
      const passwordMatch = await bcrypt.compare(password, user.password);

      if (passwordMatch) {
        req.session.user = user;
        res.redirect('/');
      } else res.render('main/auth/login.html', { user: req.session.user, pageName: "Login", error: 'Password is incorrect' });

      return
    } else {
      return res.render('main/auth/login.html', { user: req.session.user, pageName: "Login", error: 'No user found with this email' });
      return
    }
  } catch (err) {
    console.error("Server Error", err);
    return res.render('main/auth/login.html', { user: req.session.user, pageName: "Login", error: 'Server error, please try again later.' });
  }
});

app.post('/registerUser', async (req, res) => {
  var {
    name, username, email, password, gender, role, phone, dob, address,
    cardNumber, cardName, cardCountry, zipCode, expDate, cvv, profileIcon
  } = req.body;

  const hashedPassword = await bcrypt.hash(password, 10);

  if (!profileIcon || profileIcon === "") profileIcon = "public\\assets\\img\\default-avatar.png"

  const user = {
    name,
    username,
    email,
    password: hashedPassword,
    role,
    profileIcon,
    personalInfo: {
      gender,
      phone,
      dateOfBirth: dob,
      address
    },
    billingInfo: {
      cardNumber,
      cardName,
      cardCountry,
      zipCode,
      expDate,
      cvv
    }
  };

  try {
    const db = await connectDB();
    const collection = db.collection('users');
    await collection.insertOne(user);
    res.json({ "info": "Registration complete" });
  } catch (err) {
    console.error("Error saving user to database:", err);
    res.status(500).json({ "error": "Internal server error" });
  }
});
```

Figura 21 Código do Crafters Fund (Criação do Login e Registo)



```
// ===== Dashboard ===== //
```

```
app.get('/Dashboard/Tickets', async (req, res) => {
  if (!req.session.user) return res.redirect("/login");

  try {
    const db = await connectDB();
    const projectsCollection = db.collection('projects');
    const ticketsCollection = db.collection('tickets');
    let tickets;

    if (req.session.user.role === "Entrepreneur") {
      const projects = await projectsCollection.find({ ownerId: req.session.user._id }).toArray();
      const projectIds = projects.map(project => project._id);
      tickets = await ticketsCollection.find({ projectId: { $in: projectIds } }).toArray();
    } else {
      tickets = await ticketsCollection.find({ "user._id": req.session.user._id }).toArray();
    }

    const updatePromises = tickets.map(ticket => {
      if (!ticket.priority) {
        return ticketsCollection.updateOne(
          { _id: ticket._id },
          { $set: { priority: "Low" } }
        );
      }
      return null;
    }).filter(p => p !== null);

    await Promise.all(updatePromises);

    return res.render("dashboard/card-view.html", {
      user: req.session.user,
      pageName: "Dashboard",
      tickets: tickets,
    });
  } catch (err) {
    console.error("Erro ao acessar o banco de dados:", err);
    res.status(500).send("Erro interno do servidor");
  }
});
```

Figura 22 Código Crafters Fund (Funcionamento do carregamento de uma página, no caso a página de tickets no dashboard)

```
app.get("/*", async (req, res) => {
  const db = await connectDB();
  const projects = await db.collection('projects').find({}).sort({ createdAt: -1 }).limit(3).toArray();
  const devs = await db.collection('users').find({ role: "Entrepreneur" }).sort({ createdAt: -1 }).limit(6).toArray();

  if (req.path === "/" ) {
    if (req.session.user) {
      return res.render("main/index.html", { user: req.session.user, pageName: "Home", projects: projects, countries, devs });
    } else {
      return res.render("main/index.html", { user: null, pageName: "Home", projects: projects, countries, devs });
    }
  } else {
    return res.render("main/404.html", { user: req.session.user, pageName: "404" });
  }
});

app.listen(port, () => {
  console.log('Servidor rodando em http://localhost:${port}');
});
```

Figura 23 Código Crafters Fund (Inicialização do Servidor e encaminhamento de erros)

A seguir, é possível ver a estrutura de pastas que utilizei de maneira semelhante nas três subplataformas. Nesta estrutura, os 'node\_modules' contêm todas as dependências necessárias. A pasta 'public' armazena todos os dados estáticos, como imagens, vídeos, sons, etc. A pasta 'src' é reservada para arquivos externos, enquanto a pasta 'uploads' é destinada a salvar todas as imagens carregadas para a plataforma. Por fim a pasta 'views' é responsável por armazenar todos os arquivos .html.

Além disso, há alguns arquivos importantes na raiz do projeto. O arquivo '.env' armazena variáveis de ambiente, protegendo-as de serem acessadas pelo cliente. O arquivo 'package.json' gerencia todas as dependências e scripts necessários para o Node.js, e o 'index.js' contém todo o código do servidor que descrevi anteriormente.

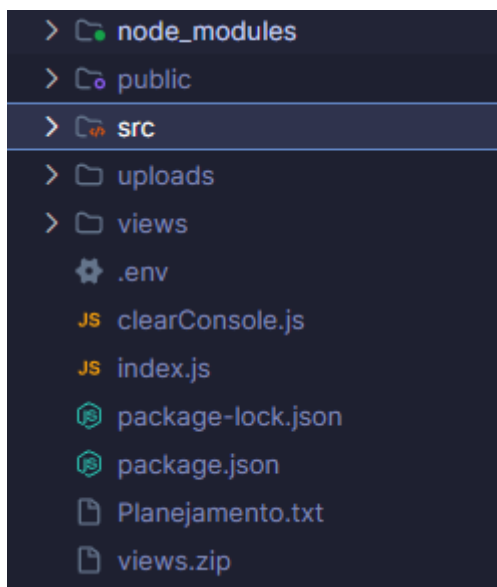


Figura 24 Estrutura de Pastas das Plataformas

```
{
  "name": "cryptoproperty",
  "version": "1.0.0",
  ▶ Debug
  "scripts": {
    "start": "node index.js"
  },
  "author": "",
  "license": "ISC",
  "description": "",
  "dependencies": {
    "bcrypt": "^5.1.1",
    "body-parser": "^1.20.2",
    "connect-mongo": "^5.1.0",
    "cookie-parser": "^1.4.6",
    "dotenv": "^16.4.5",
    "dropzone": "^6.0.0-beta.2",
    "echarts": "^5.5.0",
    "ejs": "^3.1.10",
    "express": "^4.19.2",
    "express-session": "^1.18.0",
    "moment": "^2.30.1",
    "mongodb": "^6.6.1",
    "multer": "^1.4.5-lts.1",
    "nodemon": "^3.1.0",
    "open": "^10.1.0",
    "sanitize-html": "^2.13.0"
  }
}
```

Figura 25 Ficheiro package.json o qual é utilizado para a configuração base do nodejs

Abaixo está um exemplo de uma página onde utilizei o motor de visualização EJS. Esta página, em particular, é responsável por carregar todos os projetos/jogos no catálogo da plataforma Crafters Fund, exibindo-os em forma de lista.

```
<%- include('../partials/header.html') %>

<div class="container content">
  <div class="card mb-3">
    <div class="card-body">
      <div class="row flex-between-center">
        <div class="col-sm-auto mb-2 mb-sm-0">
          <%-
            let startIndex = (page - 1) * limit + 1;
            let endIndex = startIndex + projects.length - 1;
          %>
          <h3 class="mb-0">Showing <%= startIndex %>-<%= endIndex %> of <%= totalProjects %> Products</h3>
        </div>
        <div class="col-sm-auto">
          <div class="row gx-2 align-items-center">
            <div class="col-auto">
              <form class="row gx-2">
                <div class="col-auto"><small>Sort by:</small></div>
                <div class="col-auto">
                  <select class="form-select form-select-sm" name="sort" aria-label="Bulk actions" onchange="this.form.submit()">
                    <option value="match" <%= currentSort === 'match' ? 'selected' : '' %>>Best Match</option>
                    <option value="newest" <%= currentSort === 'newest' ? 'selected' : '' %>>Newest</option>
                    <option value="price" <%= currentSort === 'price' ? 'selected' : '' %>>Price</option>
                  </select>
                </div>
              </form>
            </div>
            <div class="col-auto pe-0"><a class="text-600 px-1" href="/Projects-Grid?page=<%= page %>&limit=24" data-bs-toggle="tooltip" title="Product Grid"><span
              class="fas fa-th"></span></a></div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="card">
  <div class="card-body p-0 overflow-hidden">
    <div class="row g-0">
      <%-
        function formatValue(number) {
          if (number >= 1000000) {
            return (number / 1000000).toFixed(1) + ' M';
          } else if (number >= 1000) {
            return (number / 1000).toFixed(1) + ' K';
          }
        }
        return number.toString();
      %>
    </div>
  </div>
</div>
</div>
```

Figura 26 Página EJS de exemplo parte 1

```

%>
<% projects.forEach(function(project) { %>
  <div class="row my-2">
    <div class="col-sm-5 col-md-4 d-flex align-items-center">
      <div class="position-relative h-sm-100 overflow-hidden w-100" style="height: 150px !important;">
        <a class="d-block h-100 d-flex align-items-center w-100" href="/project/<%= project.id %>">
          <% if (project.ImagesOfPlace && project.ImagesOfPlace.length > 0) { %>
            <% if (Array.isArray(project.ImagesOfPlace)){ %>
              " />
            <% } else { %>
              
            <% } %>
          <% } else { %>
            
          <% } %>
        </a>
      <div class="badge rounded-pill bg-success position-absolute top-0 end-0 me-2 mt-2 fs-11 z-2">New</div>
    </div>
  </div>
  <div class="col-sm-7 col-md-8">
    <div class="row">
      <div class="col-lg-8">
        <h5 class="mt-3 mt-sm-0"><a class="text-1100 fs-9 fs-lg-8" href="/project/<%= project.id %>"><%= project['project-name'] %></a></h5>
        <p class="fs-10 mb-2 mb-md-3"><a class="text-500" href="#">
          <%= project['project-tag'].join(' & ') %>
        </a></p>
        <ul class="list-unstyled d-none d-lg-block">
          <li>
            <span class="fas fa-circle" data-fa-transform="shrink-12"></span>
            <span style="font-size: 0.85em;"> Required Funding:
              <%= formatValue(project['project-totalbudget'] * project['project-fundingpercentage'] / 100) %> <%= project['project-budgetcurrency'].
toUpperCase() %>
            </span>
          </li>
          <li>
            <span class="fas fa-circle" data-fa-transform="shrink-12"></span>
            <span style="font-size: 0.85em;"> Profit Share:
              <%= formatValue(project['project-profitshare'] * project['project-estimatedroi'] / 100) %>
              <%= project['project-budgetcurrency'].toUpperCase() %>
            </span>
          </li>
        </ul>
      </div>
    </div>
  </div>
} %>

```

Figura 27 Página EJS de exemplo parte 2

```
</span>
</li>
</li>
Estimated ROI: <%= formatValue(project['project-estimatedroi'].toLocaleString()) %> <%= project
'project-budgetcurrency'.toUpperCase() %>
</span>
</li>
</ul>
</div>


#### <%= formatValue(project['project-totalbudget']) %></h4> --- </span><span class="fa fa-star text-warning"></span><span class="fa fa-star <a class="btn btn-sm btn-outline-secondary border-300 d-lg-block me-2 me-lg-0 href="#"><span class="far fa-heart"></span><span </div> </div> </div> <%= }); %> </div> </div> <% if (totalPages > 1) { %> <% for (let i = 1; i ≤ totalPages; i++) { %> <a class="btn btn-sm btn-falcon-default <%= i == page ? 'text-primary' : ''" href="/Projects-List?page=<%= i %>&limit=24"><%= i %></a> <% } %> <% } %> </div> </div> </main> %- include('../partials/footer.html') %>


```

Figura 28 Página EJS de exemplo parte 3

## X. Manual do utilizador

### DevMesh

Assim que aceder o DevMesh, como mostrado abaixo, é obrigatório fazer o login, caso ainda não esteja logado, ou realizar um novo registo. No processo de registo, o utilizador precisará fornecer algumas informações básicas, conforme indicado abaixo. Essas informações incluem nome de utilizador, endereço de e-mail, senha e confirmação de senha. Este passo garante que todos os utilizadores tenham uma conta segura e personalizada, permitindo acesso completo às funcionalidades e recursos da plataforma."

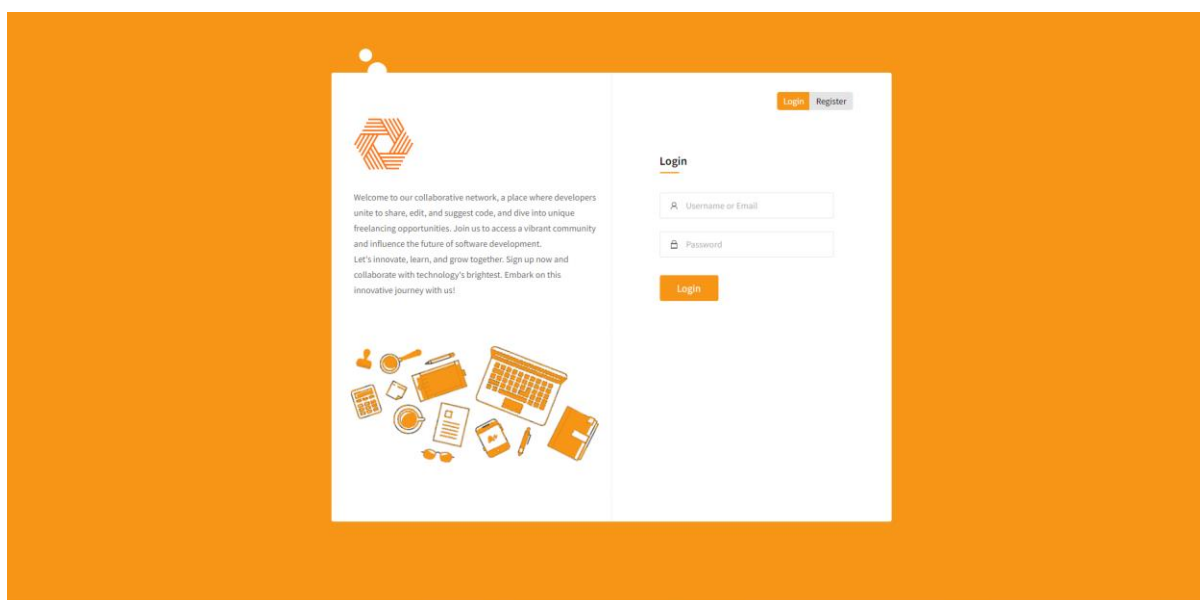


Figura 29 Pagina de Login | DevMesh

[Login](#)[Register](#)

## Register

☐ Yes, I understand and agree to the DevMesh Terms & Conditions.

[Get Started](#)

Figura 30 Forms de Registo | DevMesh

Abaixo, pode ver a página inicial, onde são exibidas algumas informações pertinentes. Onde nesta página, é possível fazer uma postagem diária inserindo as informações nos campos pedidos. Ao clicar no botão 'Make a Question', será redirecionado para a página onde pode adicionar uma nova questão, seja relacionada a um erro de código, edição de imagens, ou qualquer outro assunto, a qual será redirecionada para o Forum da plataforma.

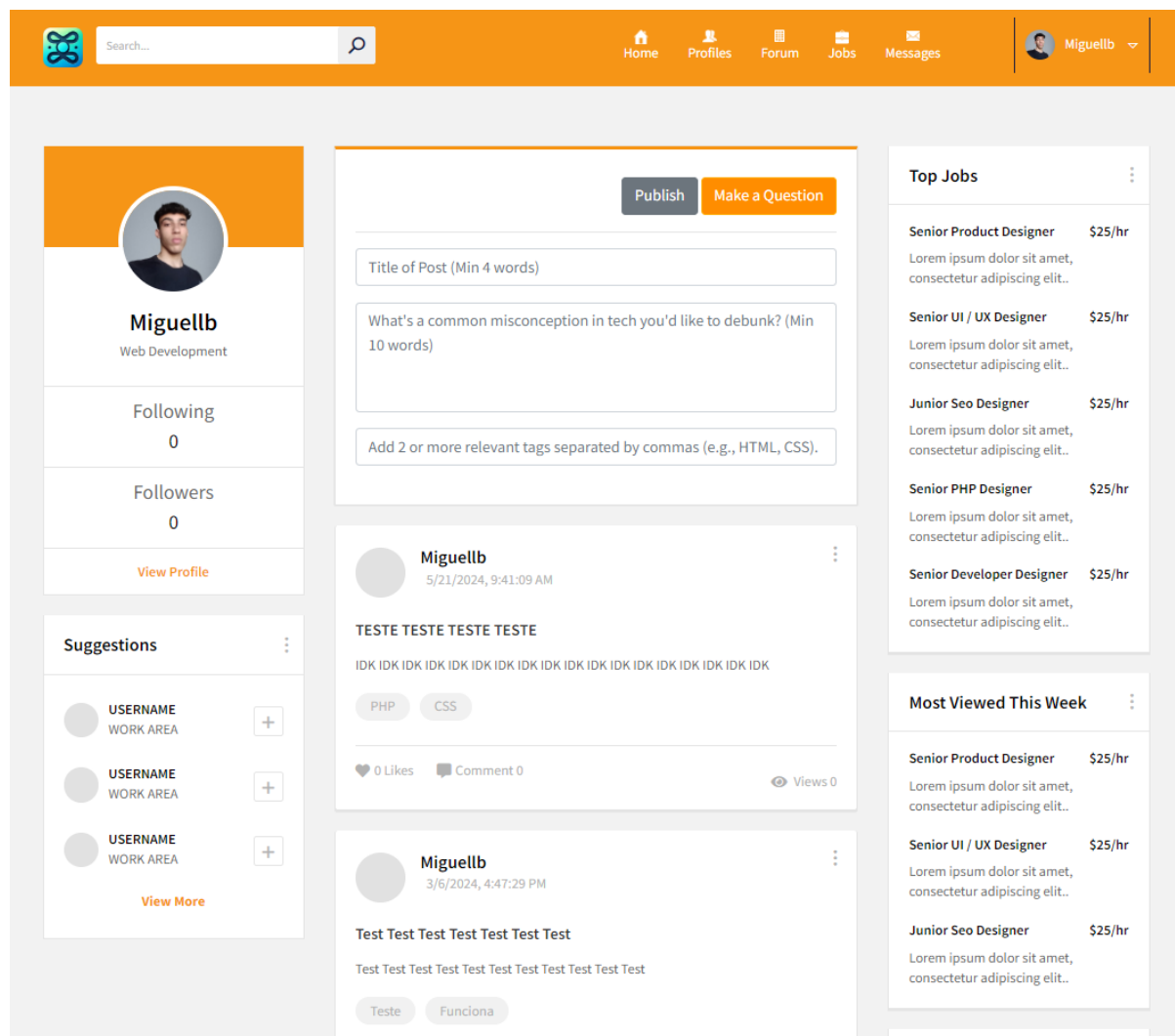
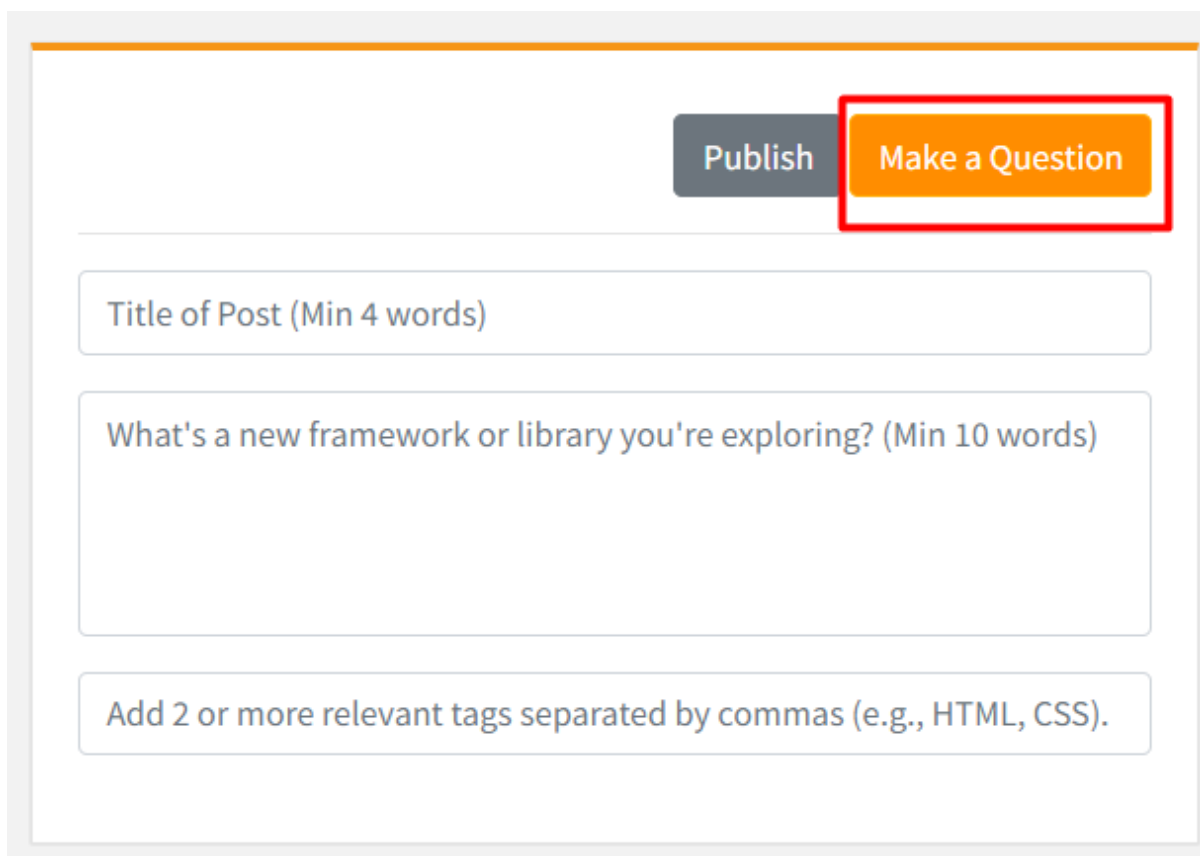


Figura 31 Pagina Home | DevMesh





The image shows a web form for creating a forum post. At the top right, there are two buttons: a grey 'Publish' button and an orange 'Make a Question' button, which is highlighted with a red rectangular border. Below the buttons, there are three input fields. The first field is labeled 'Title of Post (Min 4 words)'. The second field is labeled 'What's a new framework or library you're exploring? (Min 10 words)'. The third field is labeled 'Add 2 or more relevant tags separated by commas (e.g., HTML, CSS)'.

Figura 32 Botao para fazer uma questão para o Forum

Post a Question

Title

Enter a concise and informative title for your question.

Body

Provide a detailed description of your question.

\*\*\*

Tags

Add relevant tags separated by commas (e.g., HTML, CSS).

Post Question

Close

Figura 33 Forms para postar uma questão | DevMesh

Na página do fórum, você pode ver todas as questões postadas. Abaixo, você pode visualizar uma questão que eu havia postado anteriormente.

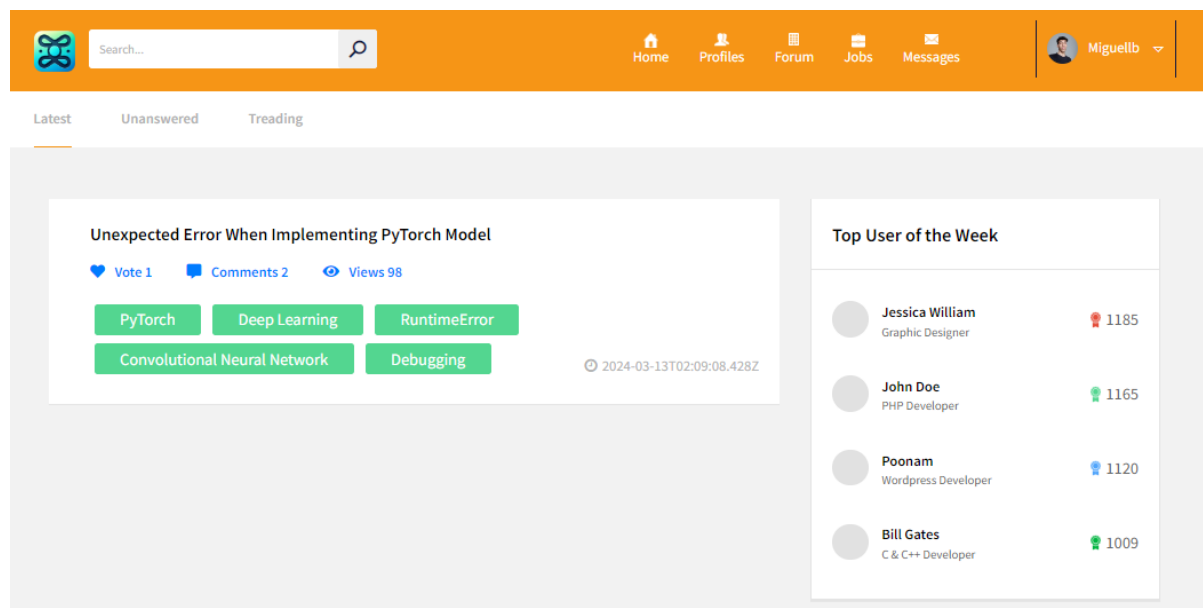


Figura 34 Pagina de Forum

Já na página de perfis, são listados os perfis dos utilizador mais recomendados para a pessoa, na qual ela pode visualizar o perfil e até falar no sistema de chatting disponível na plataforma

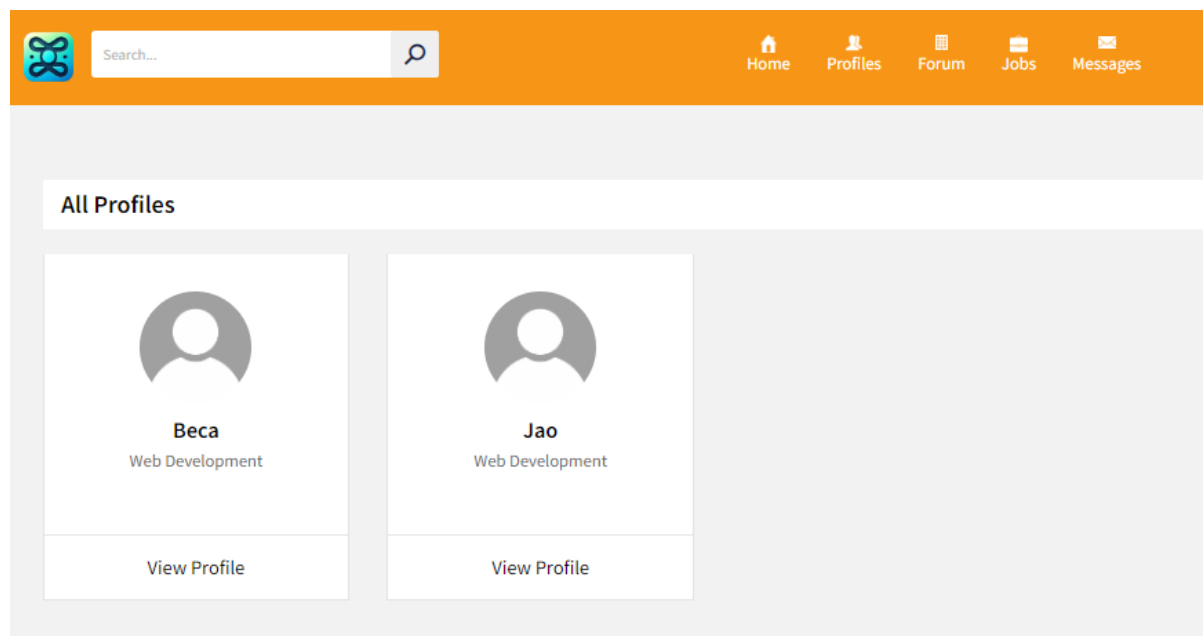


Figura 35 Pagina de Perfis

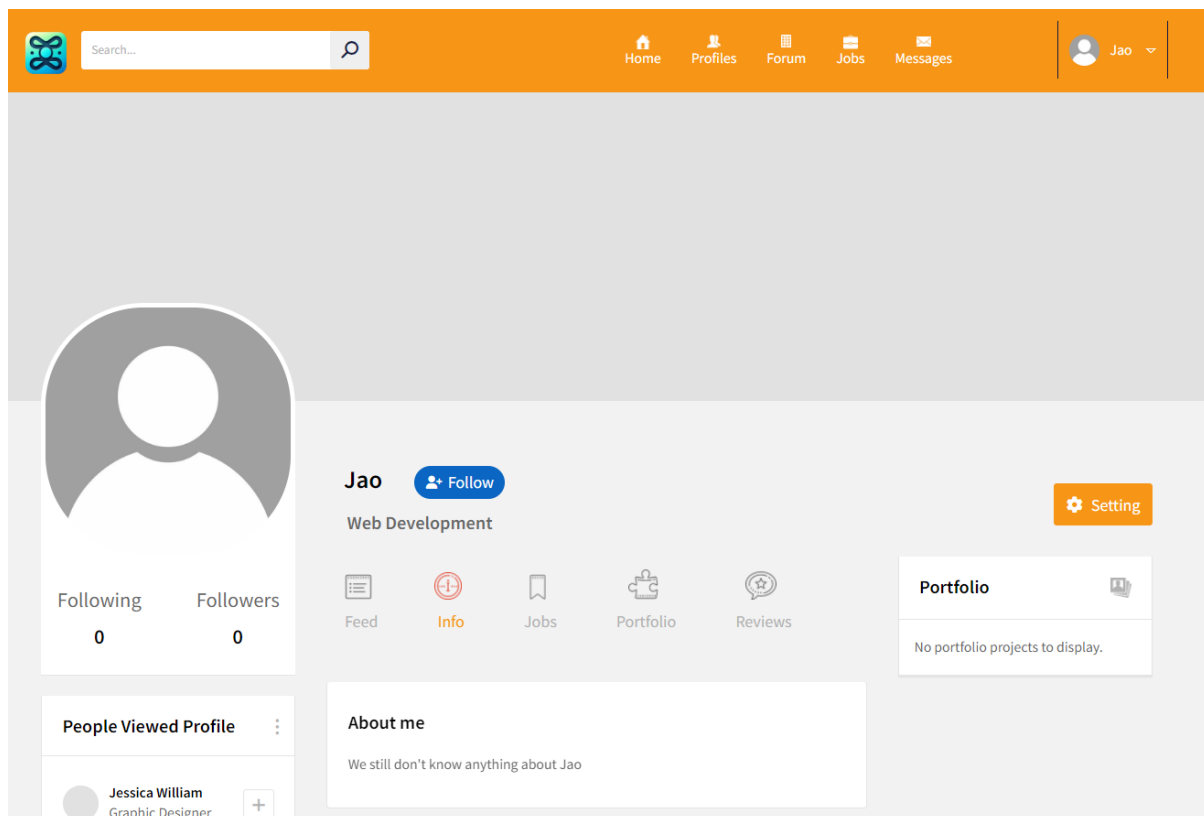


Figura 36 Página de perfil do Jão

## Crafters Fund

Na página de Home da plataforma do Crafters Fund é possível entender de forma concisa a ideia da própria plataforma, começando com uma “capa” de destaque que descreve resumidamente do que se trata a plataforma seguida por dois caixas onde uma redireciona para a lista de jogos enquanto o outro de propõem a te tornares um desenvolvedor na plataforma, por fim lista os desenvolvedores da plataforma (no caso só tinha integrado um) e é listado os 3 principais projetos (aqui só apresentado um por ser o único projeto disponível)

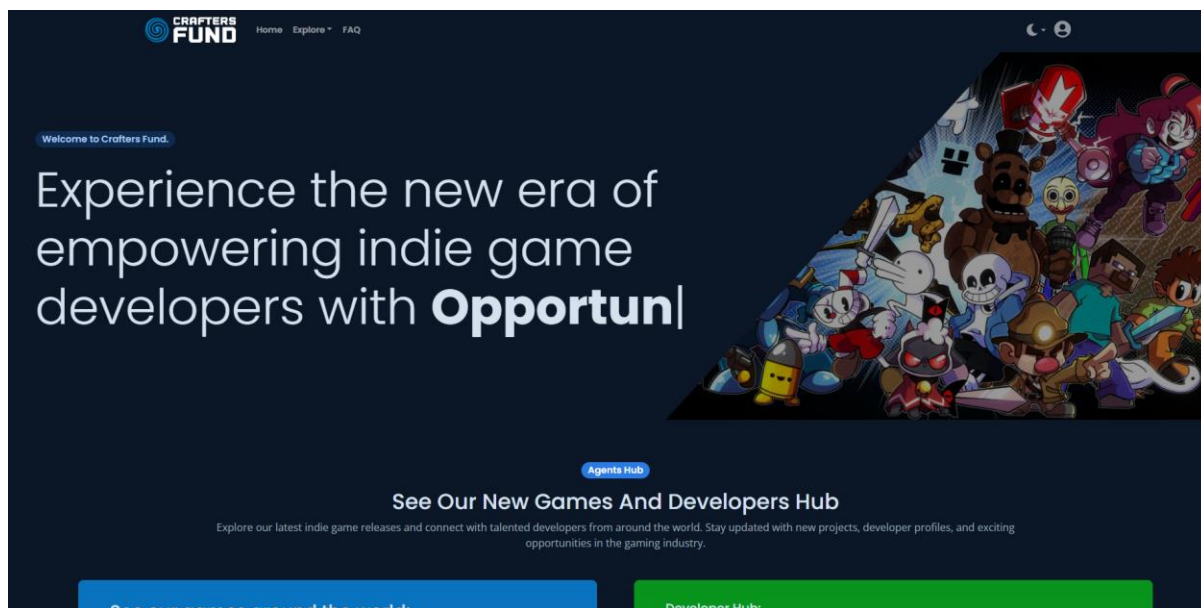


Figura 37 Pagina Home Parte 1 | Crafters Fund

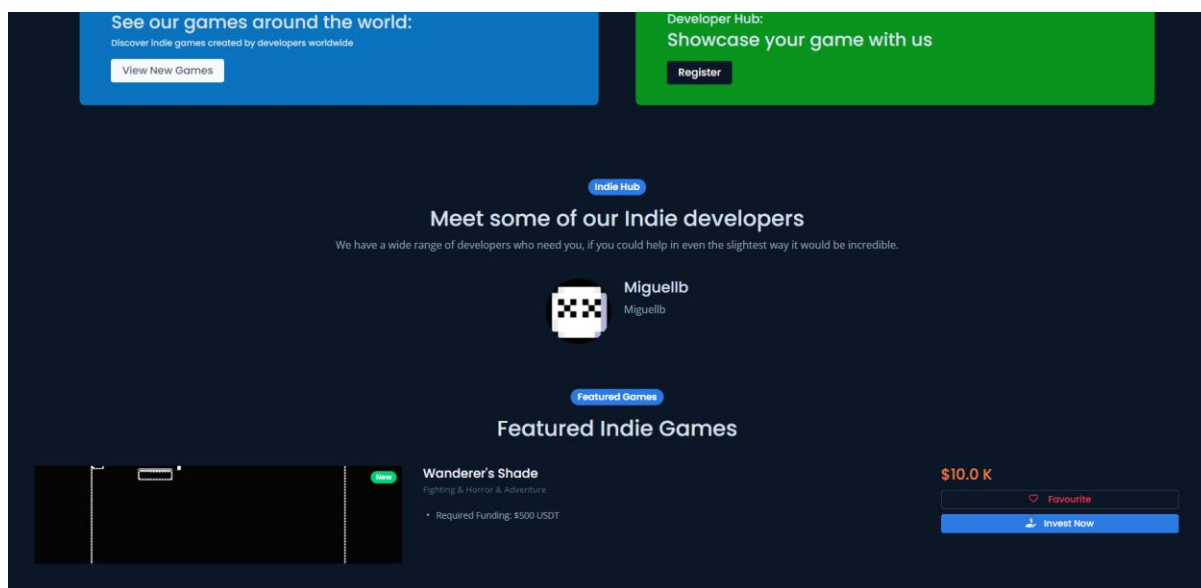


Figura 38 Pagina Home Parte 2 | Crafters Fund

Ao ir para a página de Explore é ilustrado a lista de projetos que é possível financiar junto da página de informações do projeto (onde já tinha deixado previamente o jogo que já desenvolvi)

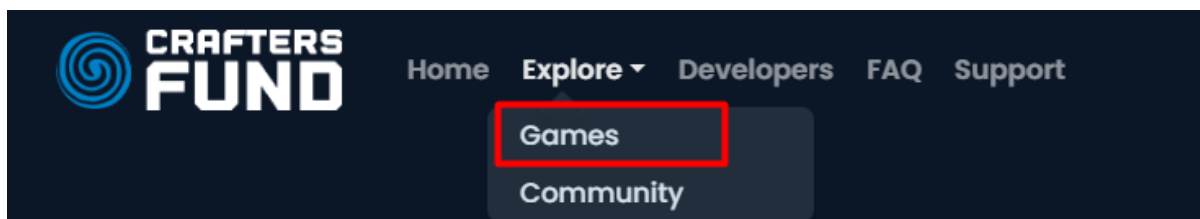


Figura 39 Botao de redirecionamento para a lista de jogos no menu | Crafters Fund

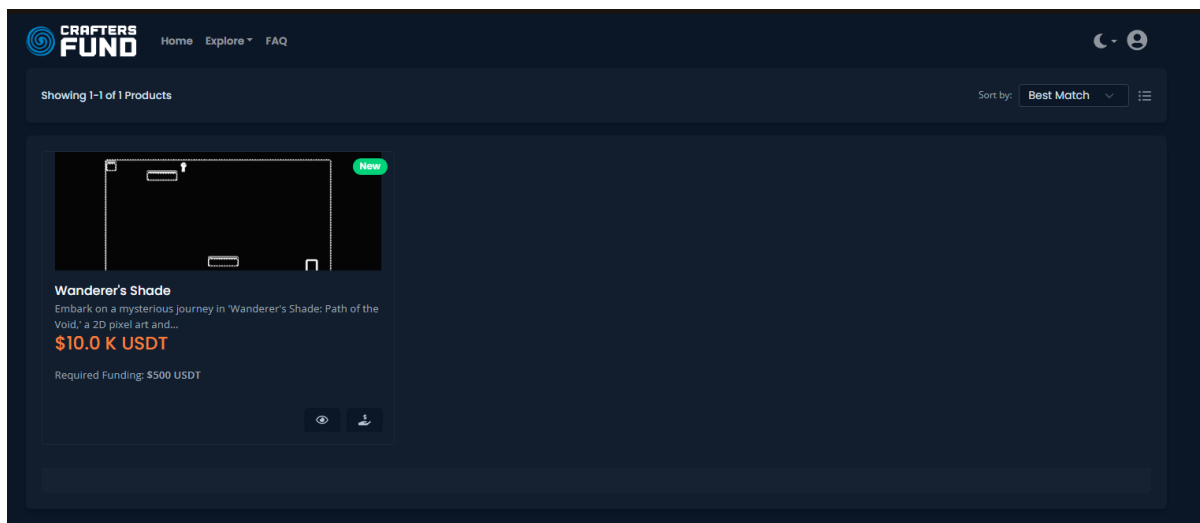


Figura 40 Lista de Projetos/Jogos | Crafters Fund

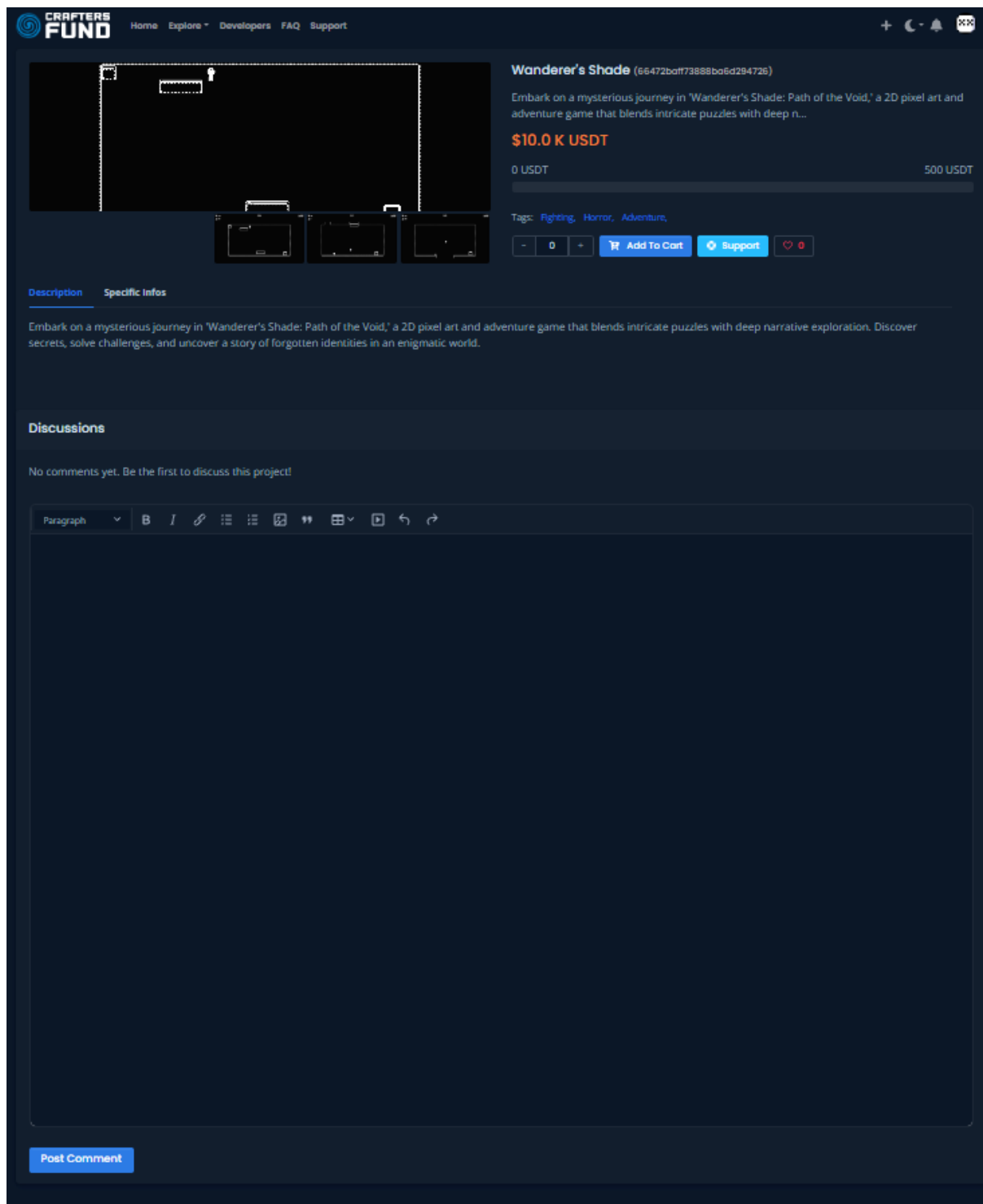


Figura 41 Pagina de um Projeto

Ao aceder a página de perfil, se você ainda não tiver um perfil, será redirecionado para a página de criação de perfil. Caso contrário, você poderá visualizar o seu perfil e os seus seguidores.

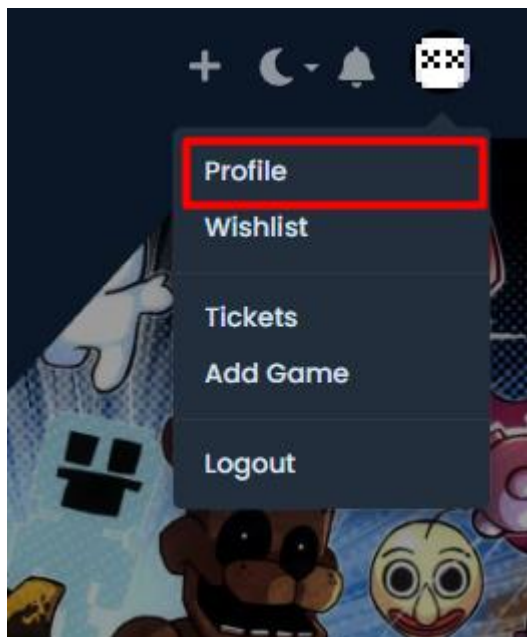



Figura 42 Botao de redirecionamento para o Perfil | Crafters Fund



**CRAFTERS FUND** Home Explore Developers FAQ Support

Submit/Change your Profile [Submit](#)

 Update your profile picture  
Upload a 300x300 jpg image with a maximum size of 400KB

Miguelb

Your title

Your location

**About you**

Paragraph B I Link List Table Image Quote Video Undo Redo

**Experience**

[Add Experience](#)

**Education**

[Add Education](#)

Figura 43 Pagina de Criação do Perfil

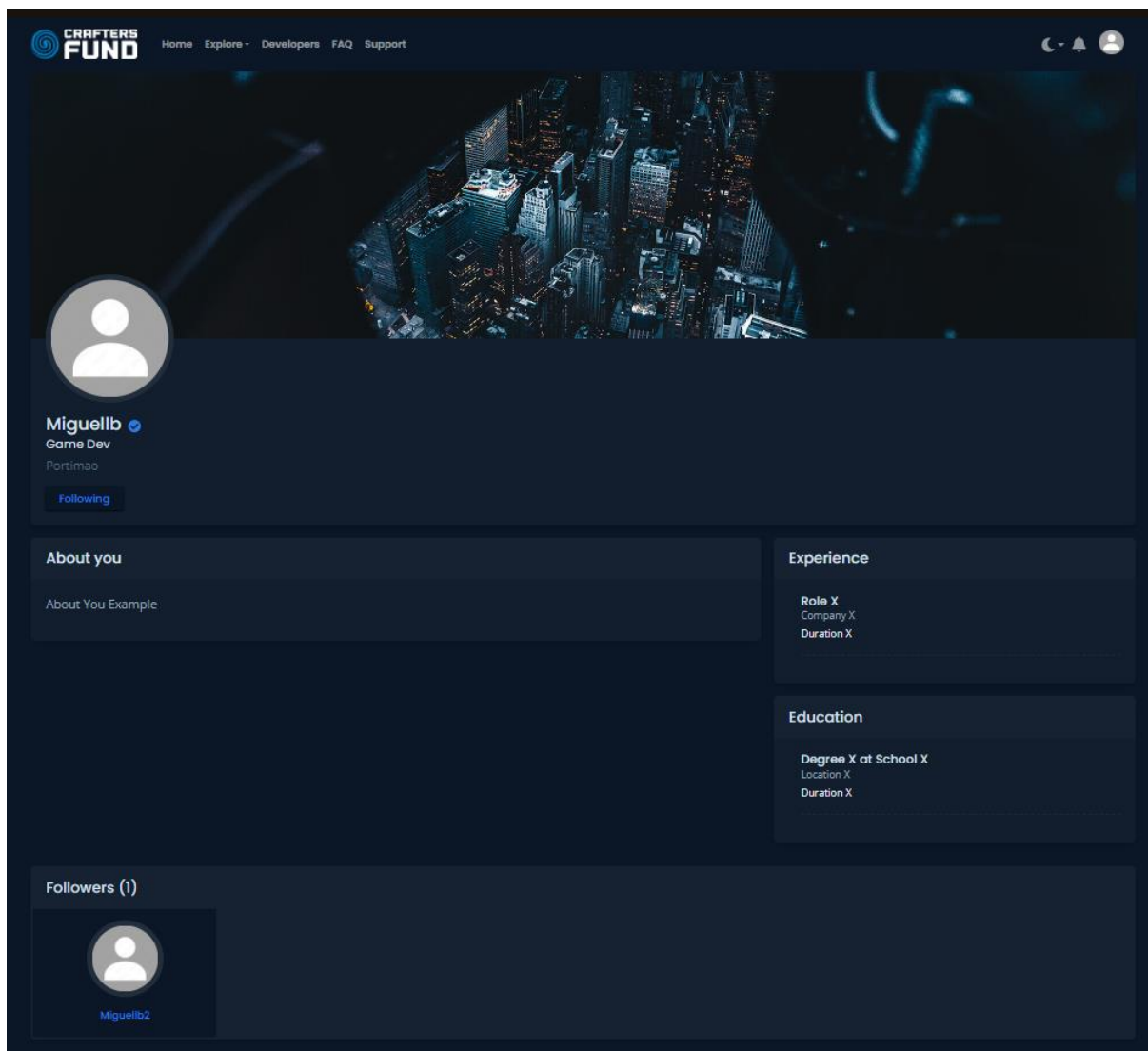


Figura 44 Pagina de Perfil de Exemplo | Crafters Fund

Ao ir para a página de suporte, é possível criar um ticket para um projeto. O criador do projeto pode responder e se comunicar continuamente com o utilizador através do ticket até que a questão seja resolvida. Quando o problema for resolvido, o criador pode então clicar em 'close' para encerrar o ticket.

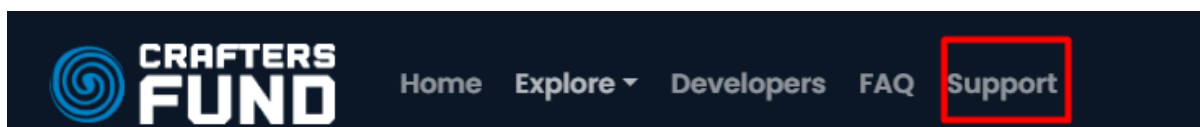


Figura 45 Botão de redirecionamento para a página de suporte | Crafters Fund

The image displays the 'Support' page form. It has a dark theme. At the top, it says 'Support'. Below that, there's a section 'Select Game for Ticket' with a dropdown menu showing 'Wanderer's Shade'. Then, there's a 'Title (Max. 80 Characters)' field with 'Title Example' entered. Below the title is a 'Description' field with a rich text editor toolbar (Paragraph, Bold, Italic, Link, etc.) and 'Description Example' entered. At the bottom left of the form is a blue 'Submit' button.

Figura 46 Forms de suporte de um Projeto | Crafters Fund

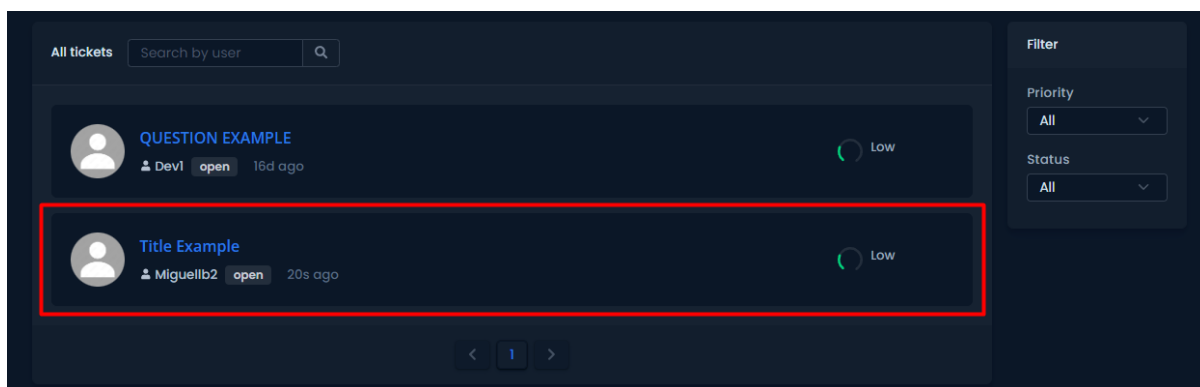


Figura 47 Ticket Criado | Crafters Fund

Nas duas imagens que disponibilizei a seguir é demonstrado com o dono do Projeto vê os tickets que lhe foram enviados e como o investidor vê.

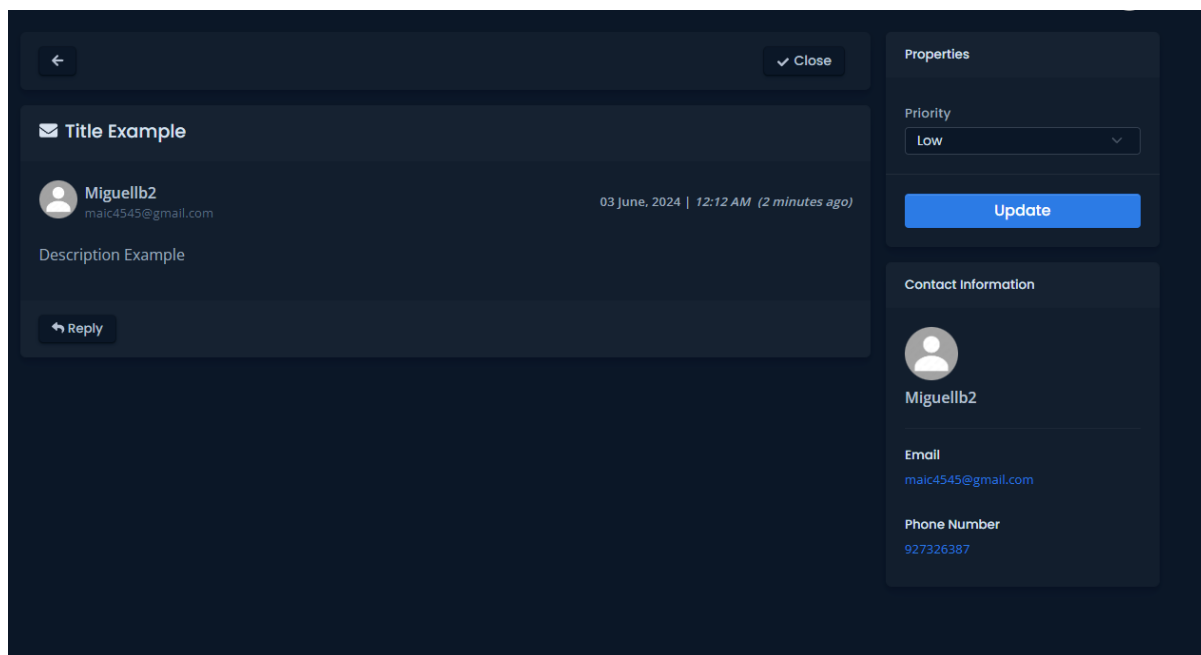


Figura 48 Página do Ticket pela visão do dono do Projeto | Crafters Fund

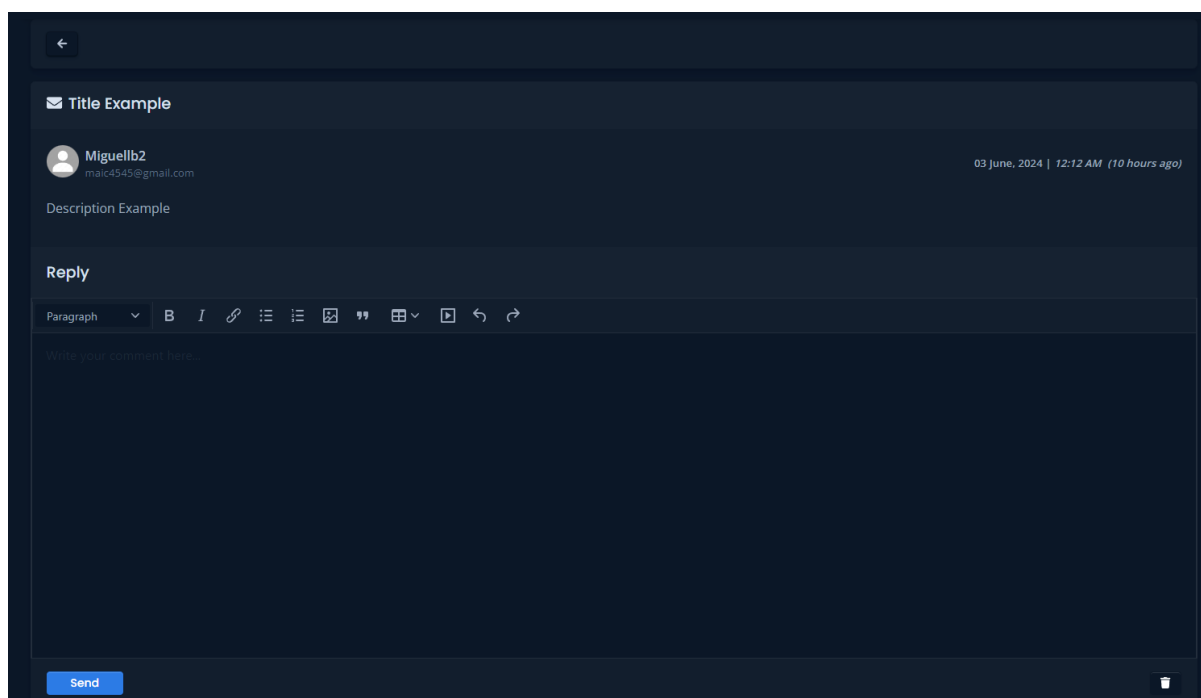


Figura 49 Página do Ticket pela visão do investidor | Crafters Fund

The screenshot displays a ticket interface with a dark theme. At the top, the title 'Title Example' is shown with an envelope icon. Below it, the user 'Miguelb2' (maic4545@gmail.com) is listed, along with the timestamp '03 June, 2024 | 12:12 AM (2 minutes ago)'. The 'Description Example' is visible. The 'Reply' section features a rich text editor with a toolbar containing options for Paragraph, Bold (B), Italic (I), Link, List, Image, Quote, Table, Video, Undo, and Redo. The text area contains the placeholder 'Write your comment here...'. A blue 'Send' button is at the bottom left, and a trash icon is at the bottom right.

Figura 50 Possibilidade de responder ao Ticket | Crafters Fund

This screenshot shows the same ticket interface as Figure 50, but with a 'Close' button (marked with a checkmark) highlighted by a red rectangle in the top right corner. To the right of the ticket details is a 'Properties' sidebar containing a 'Priority' dropdown set to 'Low', an 'Update' button, and a 'Contact Information' section.

Figura 51 Botao para o desenvolvedor fechar o ticket | Crafters Fund

## BrainBurst

Ao entrar na subplataforma Brain Burst tem algumas paginas de informação mas na pagina inicial é possível escolher entre instalar a aplicação ou abri-la em browser

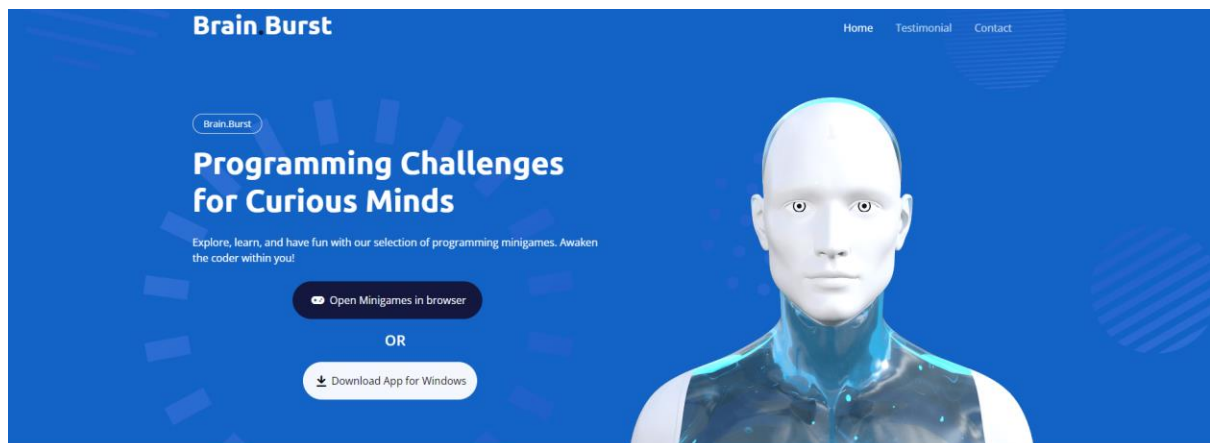


Figura 52 Pagina Home | BrainBurst

Ao clicar no browser (no aplicativo é igual) é possível escolher entre duas opções:

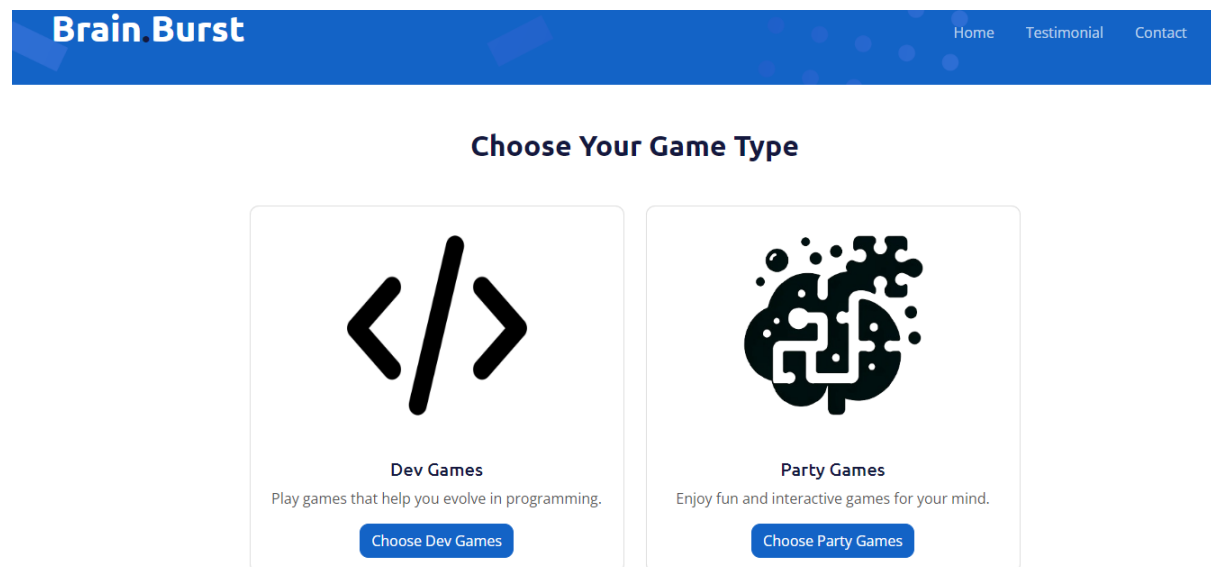


Figura 53 Pagina de escolha do tipo de Minijogo

Caso se escolha o DevGames terá estes dois minijogos, no caso o Code Sherrif ficou em desenvolvimento, então ainda não é utilizável, já o Code Master é possível abrir e escolher entre 100 níveis do curso que vão escalando de dificuldade e tema, mais abaixo é possível ver o primeiro exercício que se dá como exemplo pois todos os exercícios seguem da mesma página apenas mudando o propósito.

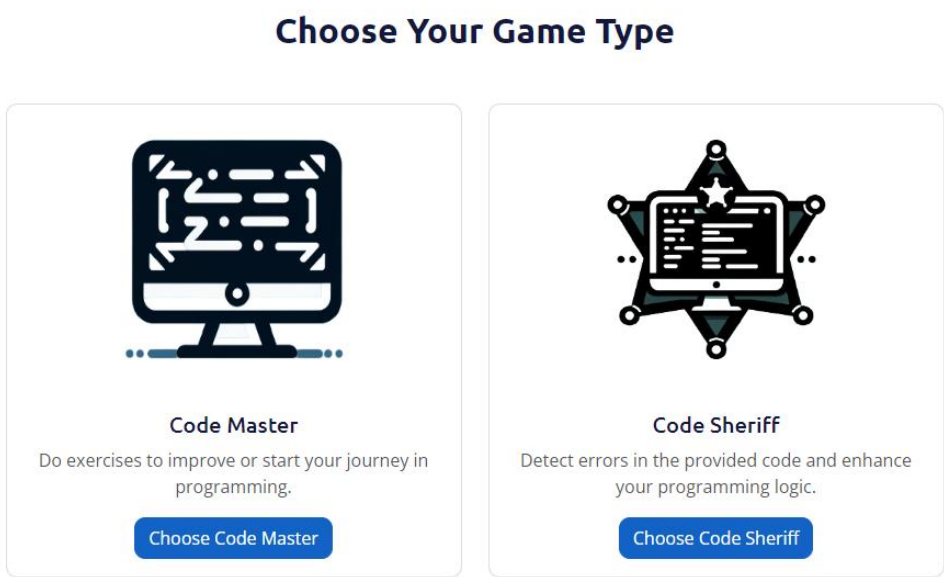


Figura 54 Menu de Escolha do Minijogo de Programação | BrainBurst

Programming Challenges		
Title	Tags	Difficulty
3. Circling Around with Pi and R	math formulas	Easy
5. Odd One Out or Even Steven?	conditional logic	Easy
20. Word Counter	strings counting	Easy
23. Anagram Groups	strings hashing	Intermediate
27. Dijkstra's Shortest Path Algorithm	algorithms graphs	Intermediate
47. Edit Distance Problem	dynamic programming strings	Advanced
49. Maximum Subarray Sum	arrays dynamic programming	Advanced
50. Implement Trie (Prefix Tree)	data structures trie prefix tree	Advanced

Figura 55 Lista de Níveis do Code Master | BrainBurst



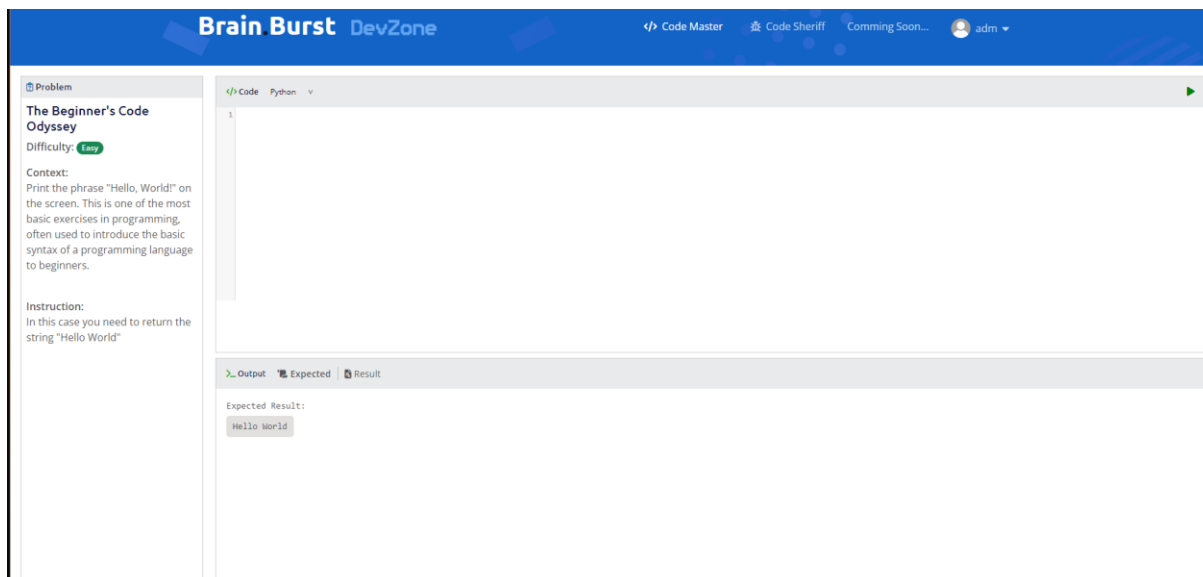


Figura 56 Pagina do Primeiro Exercício

## **XII. Análise crítica da execução do projeto**

A execução do projeto representou uma jornada de aprendizagem intensiva e desenvolvimento pessoal, oferecendo uma oportunidade valiosa para aplicar conhecimentos teóricos em um contexto prático. Esta análise crítica visa refletir sobre os aspetos chave do projeto, destacando sucessos, desafios e lições aprendidas.

### **Sucessos do Projeto**

#### **Inovação e Criatividade:**

Uma das maiores conquistas foi a capacidade de integrar diversas funcionalidades, como exercícios de programação, minijogos, um serviço de chat, uma rede social e serviço de “solidariedade”, numa única plataforma. Esta abordagem inovadora não só cumpriu o objetivo de criar um ambiente estimulante e educativo para os utilizadores, mas também demonstrou habilidades criativas e técnicas avançadas.

#### **Autonomia na Resolução de Problemas:**

O desenvolvimento do projeto exigiu uma significativa capacidade de pesquisa e resolução de problemas de forma independente. A capacidade de superar desafios técnicos, desde a integração de diferentes tecnologias até à otimização da experiência do utilizador, reflete um crescimento profissional notável.

#### **Qualidade Técnica:**

A adoção de práticas de desenvolvimento como testes contínuos resultou num produto final robusto e de alta qualidade. A atenção dada à usabilidade e ao design da interface demonstra um compromisso com a criação de uma aplicação que não só funciona bem mas também é agradável de usar.

### **Desafios Enfrentados**

#### **Gestão de Tempo:**

Um dos principais desafios foi a gestão eficaz do tempo, especialmente considerando a complexidade e a amplitude do projeto. A conciliação das tarefas de desenvolvimento com outras responsabilidades académicas e pessoais exigiu um planeamento cuidadoso e priorização eficiente das atividades.

#### **Lições Aprendidas**

#### **Importância do Planeamento e Flexibilidade:**

Ficou evidente que um planeamento cuidadoso é crucial para o sucesso de um projeto desta envergadura. No entanto, também aprendi a importância da flexibilidade, adaptando-se às mudanças e ajustando o planeamento conforme necessário.

**Valor da Pesquisa e Aprendizagem Contínua:**

O desenvolvimento reforçou a necessidade de estar constantemente a aprender e a atualizar conhecimentos, especialmente num campo tão dinâmico como o da tecnologia.

**Exposição Final**

Em suma, a execução do projeto foi uma experiência enriquecedora, proporcionando uma perspetiva valiosa sobre o desenvolvimento de software e gestão de projetos. As dificuldades encontradas foram superadas com determinação e serviram como oportunidades de aprendizagem, contribuindo para o desenvolvimento.

## Conclusão

A conclusão deste projeto reflete uma jornada enriquecedora e desafiadora que abrangeu desde a concepção até a implementação de três subplataformas distintas: BrainBurst, DevMesh e CraftersFund. Cada uma destas subplataformas foi criada com o intuito de atender a necessidades específicas dentro da comunidade de desenvolvedores, oferecendo ferramentas e funcionalidades que promovem o aprendizado, a colaboração e o financiamento de projetos.

## Realizações e Impacto

- **Inovação e Criatividade:** A integração de funcionalidades diversas, como exercícios de programação, minijogos, redes sociais para desenvolvedores e plataformas de financiamento, resultou em um produto final inovador e multifacetado. A capacidade de combinar entretenimento com ferramentas de desenvolvimento realça a originalidade e utilidade das subplataformas.
- **Autonomia e Resolução de Problemas:** A abordagem autodirigida do projeto fomentou o desenvolvimento de habilidades de pesquisa, resolução de problemas e autonomia. A superação de desafios técnicos e a implementação de soluções eficazes demonstram um crescimento significativo em competências técnicas e de gestão de projetos.
- **Qualidade Técnica:** A aplicação de metodologias iterativas, testes contínuos e a utilização de tecnologias modernas garantiram um produto final robusto e de alta qualidade. O foco na experiência do utilizador e na usabilidade evidenciou um compromisso com a excelência.

## Desafios e Aprendizados

- **Gestão de Tempo:** A complexidade do projeto e a necessidade de equilibrar responsabilidades académicas e pessoais apresentaram desafios significativos na gestão do tempo. A experiência sublinhou a importância de um planeamento cuidadoso e da capacidade de adaptação.
- **Flexibilidade e Planeamento:** A experiência destacou a necessidade de flexibilidade no planeamento, permitindo ajustes e adaptações conforme necessário. Esta lição será valiosa em futuros empreendimentos.
- **Aprendizagem Contínua:** A constante necessidade de aprender e atualizar conhecimentos sublinhou a importância da pesquisa contínua e do desenvolvimento profissional. A utilização de recursos como W3Schools e assistentes virtuais, como o ChatGPT, demonstrou-se essencial para resolver dúvidas e aprimorar o projeto.

## Impacto Futuro e Contribuição

Este projeto não só resultou em produtos que atendem às necessidades específicas da comunidade de desenvolvedores, mas também contribuiu para o conhecimento coletivo na área de desenvolvimento de software. As lições aprendidas e as metodologias aplicadas servirão de inspiração para futuras empreitadas tecnológicas, proporcionando uma base sólida para projetos futuros.

## Exposição Final

Em resumo, a execução deste projeto foi uma experiência profundamente gratificante e educativa. As dificuldades enfrentadas foram transformadas em oportunidades de aprendizado, resultando em um produto final que combina inovação, funcionalidade e usabilidade. Através desta jornada, foi possível desenvolver habilidades técnicas e de gestão de projetos, ao mesmo tempo em que se contribuía para a comunidade de desenvolvedores com soluções práticas e inspiradoras.

## Fontes de pesquisa de informação

‘ChatGPT. (2023/2024). Obtido de ‘OpenAi | ChatGPT: <https://chatgpt.com>

W3schools. (2023/2024). Obtido de W3Schools Online Web Tutorials: <https://www.w3schools.com>

Durante o desenvolvimento do projeto, qualquer dúvida ou desafio técnico que surgiu foi eficazmente solucionado com o apoio de fontes de informação especializadas. Para garantir a precisão técnica e a aplicação das melhores práticas no desenvolvimento da aplicação, recorri predominantemente ao W3Schools, bem como à assistência de assistentes virtuais e inteligências artificiais (como o ChatGPT).

### W3Schools:

Este recurso foi inestimável para a resolução de dúvidas relacionadas com programação e desenvolvimento web. O W3Schools oferece tutoriais, exemplos de código e referências para uma vasta gama de tecnologias web, incluindo HTML, CSS, JavaScript, e frameworks e bibliotecas específicas usadas no projeto. A facilidade de acesso e a clareza das explicações fornecidas por este recurso foram cruciais para a implementação eficiente de funcionalidades e a resolução de problemas de programação.

### Assistentes Virtuais/IA

A utilização de assistentes virtuais e plataformas de inteligência artificial, como o ChatGPT, desempenharam um papel complementar significativo na pesquisa de informações e na resolução de dúvidas. Estes assistentes permitiram obter respostas rápidas e orientação sobre questões de programação, conceitos de desenvolvimento de software e melhores práticas de design. A capacidade de interagir com estas ferramentas em tempo real e receber feedback instantâneo acelerou o processo de aprendizagem e contribuiu para a tomada de decisões informadas ao longo do projeto.

A combinação destas fontes de pesquisa de informação provou ser extremamente eficaz, permitindo não só resolver dúvidas específicas de forma rápida e eficiente, mas também enriquecer o conhecimento técnico necessário para o desenvolvimento de um projeto complexo e inovador.