

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
PERNAMBUCO (IFPE)
CURSO SUPERIOR EM TECNOLOGIA EM SISTEMAS PARA INTERNET (TSI)**

ERICKSON FERREIRA DA SILVA NASCIMENTO

**IMPROOV: UMA FERRAMENTA DE AUXÍLIO NO PROCESSO DE ENSINO
AUTODIDATA - REGISTRO DE SOFTWARE, DESENVOLVIMENTO E
IMPLANTAÇÃO.**

Igarassu, Pernambuco

2025

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
PERNAMBUCO (IFPE)
CURSO SUPERIOR EM TECNOLOGIA EM SISTEMAS PARA INTERNET (TSI)**

ERICKSON FERREIRA DA SILVA NASCIMENTO

**IMPROOV: UMA FERRAMENTA DE AUXÍLIO NO PROCESSO DE ENSINO
AUTODIDATA - REGISTRO DE SOFTWARE, DESENVOLVIMENTO E
IMPLANTAÇÃO.**

Relatório apresentado como Trabalho de Conclusão de Curso (TCC), como requisito para obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador:

Prof. Ramon Mota de Souza Farias

Igarassu, Pernambuco

2025

ÍNDICE DE ILUSTRAÇÕES

FIGURAS

Figura 1: Tela Inicial da aplicação.....	11
Figura 2: Tela de Criar Missão.....	12
Figura 3: Execução de teste automatizado com Cypress.....	27

TABELAS

Tabela 1: Requisitos funcionais da ferramenta.....	14
Tabela 2: Requisitos não funcionais da ferramenta.....	15
Tabela 3: Caso de Teste Realizar Cadastro via GitHub.....	23
Tabela 4: Caso de Teste Realizar Autenticação via GitHub.....	24
Tabela 5: Caso de Teste Criar Missões.....	24
Tabela 6: Caso de Teste Realizar Missões.....	25
Tabela 7: Caso de Teste Favoritar Projeto.....	26
Tabela 8: Caso de Teste Ver Desempenho.....	27

LISTA DE ABREVIACOES

TI	Tecnologia da Informao
COVID-19	Coronavirus disease 2019 (doena por coronavrus 2019)
TSI	Tecnologia em Sistemas para Internet
IFPE	Instituto Federal de Pernambuco
INPI	Instituto Nacional da Propriedade Industrial
PHP	Hypertext Preprocessor (Preprocessador de Hypertexto)
API	Application Programming Interface (Interface de Programao de Aplicao)
MVC	Modelo-Viso-Controle

RESUMO

Este relatório tem como objetivo descrever o processo de desenvolvimento e registro de software da aplicação Improov, que foi criado a partir da linguagem de programação PHP utilizando o framework Laravel, banco de dados MySQL e com integração à API do Github, como pré-requisito à obtenção de título de Tecnólogo em Sistemas para Internet. Durante o curso, elaboramos um projeto voltado para auxiliar o aprendizado autodidata de estudantes iniciantes na área de tecnologia através de uma plataforma gamificada. Durante o desenvolvimento contribuimos em várias etapas do processo, tais como: análise e pesquisa de público alvo, prototipação, desenvolvimento do layout e do código fonte, implantação do software em serviço de Plataforma de nuvem.

Palavras-chaves: Desenvolvimento de software; Laravel; PHP; MySQL.

ABSTRACT

This report aims to describe the software development and registration process for the Improov application, which was created using the PHP programming language, the Laravel framework, MySQL database, and integration with the Github API, as a prerequisite for obtaining a degree in Internet Systems Technology. During the course, we developed a project aimed at assisting the self-directed learning of beginner students in the technology field through a gamified platform. During development, we contributed to several stages of the process, such as: target audience analysis and research, prototyping, layout and source code development, and software deployment on a cloud platform service.

SUMÁRIO

1. INTRODUÇÃO.....	7
2. O PROJETO IMPROOV.....	10
2.1. Caracterização da Solução.....	10
2.2. Tecnologia Envolvida.....	12
2.3. Desenvolvimento do Projeto Improov.....	13
2.3.1. Análise e Levantamento de Requisitos.....	13
2.3.2. Primeira Fase: Integração Inicial com GitHub e Desenvolvimento de Missões	15
2.3.2.1. Integração com a API do GitHub.....	16
2.3.2.2. Desenvolvimento de Missões.....	16
2.3.2.3. Análise de Desempenho e Projetos Favoritos.....	16
2.3.3. Segunda Fase: Criação de Times e Expansão das Funcionalidades.....	17
2.3.3.1. Criação e Gerenciamento de Times.....	18
2.3.3.2. Sistema de Missões de Times e Hierarquia de Cargos.....	18
2.3.3.3. Validação de Missões e Sistema de Ranqueamento.....	18
2.3.3.4. Vínculo de Amizades.....	18
2.3.4. Terceira Fase: Integração de Dados Privados e Webhooks.....	19
2.3.4.1. Solicitação de Acesso a Escopos Privados.....	19
2.3.4.2. Reestruturação do Projeto.....	19
2.3.4.3. Implementação de Webhooks.....	21
2.3.4.4. Monitoramento em Tempo Real.....	22
3. IMPLANTAÇÃO E TESTES.....	22
4. REGISTRO DE SOFTWARE.....	29
5. CONSIDERAÇÕES FINAIS.....	31
6. BIBLIOGRAFIA.....	32
ANEXO A – CERTIFICADO DE REGISTRO DE SOFTWARE EMITIDO PELO INPI.....	34

1. INTRODUÇÃO

De acordo com o relatório de Inteligência e Informação Brasscom (2019), o mercado de tecnologia está bastante aquecido: as perspectivas de investimentos até 2022 nas áreas de tecnologia e transformação digital eram de 345,5 bilhões de reais, demandando cerca de 420 mil profissionais até 2024. Esses números apontam para a necessidade de formação de mão de obra qualificada no curto prazo. Ainda segundo a Brasscom (2019), o Brasil formava cerca de 46 mil pessoas com perfil tecnológico por ano, causando uma disparidade geográfica entre oferta e demanda de mão de obra do setor.

Em uma pesquisa realizada pelo MIT *Technology Review Insights* (2023), líderes de empresas do setor de Tecnologia da Informação ao redor do mundo relatam que 64% dos candidatos entrevistados para vagas na área de TI não possuem as competências ou a experiência necessárias. Em outro estudo, realizado pelo ManpowerGroup (2023) sobre a escassez de talentos entre 41 países e territórios, o Brasil está entre os 15 países que encontram maior dificuldade para empregar profissionais qualificados, evidenciando, dessa forma, a necessidade urgente da implementação de estratégias eficazes para minimizar esse déficit.

Em entrevista para o site Factorial, Patrícia Duchnick (2022), especialista em recrutamento na área de tecnologia, compartilha que, "com a possibilidade de atuação remota, os profissionais se tornaram globais com possibilidade de contratação por qualquer empresa no mundo, tornando o mercado mais competitivo". Além de ter que lidar com uma crescente escassez de talentos com habilidades técnicas adequadas, os recrutadores enfrentam dificuldades, também, com a necessidade de se fazer um processo seletivo mais complexo e rigoroso. No entanto, segundo Duchnick (2022), as empresas devem se adaptar ao cenário atual: "é necessário que o recrutador vá ao encontro do candidato onde ele está". Para isso, uma das estratégias apontadas pela especialista é que as empresas devem promover eventos como *Meetups*, *Hackathons*, *Webinars* e apoiar comunidades de tecnologias. Outra estratégia sugerida é oferecer cursos ou oportunidades de especialização (DUCHNICKY, 2022).

A situação já desafiadora no recrutamento de talentos foi agravada pela pandemia da COVID-19, que teve repercussões profundas no mercado de trabalho global, causando um desemprego nas mais diversas áreas de trabalho. Em 2021,

segundo o Instituto Brasileiro de Geografia e Estatística (IBGE), o número de desempregados no Brasil ultrapassou 15,2 milhões de pessoas, atingindo a taxa de 14,9% no primeiro trimestre. Ao longo do mesmo ano, o mercado de trabalho brasileiro mostrou sinais de recuperação, fechando o ano com a taxa de desemprego em 11,1% (IBGE, 2021). Apesar de o Brasil registrar uma diminuição de desempregados, essa melhora está vinculada ao crescimento na taxa de informalidade, indicando um desafio contínuo na restauração do mercado de trabalho.

Diante dessas estatísticas, é necessário que profissionais em transição de carreira procurem por novas formas de se qualificar para atender as oportunidades disponíveis no mercado de tecnologia através, por exemplo, da realização de cursos em plataformas online ou participando de iniciativas de educação guiadas e autodidatas.

Sabendo desse déficit do mercado e do grande número de profissionais em transição de carreira em busca de uma nova oportunidade de emprego, foi desenvolvido, através das disciplinas práticas do curso de Tecnologia em Sistemas para Internet (TSI), um software que atua de forma híbrida: promovendo a evolução técnica do usuário e servindo como ponte para suas ambições profissionais. Ao longo do curso foram trabalhados os conhecimentos técnicos nas tecnologias do *framework* Laravel, assim como o banco de dados MySQL e a linguagem de programação PHP, que são lecionadas durante grande parte da grade curricular nos cursos de tecnologia do Instituto Federal de Pernambuco (IFPE) *campus* Igarassu. O desenvolvimento desse *software* foi evoluindo e sendo submetido a avaliações ao longo de três semestres, pondo em prática os conceitos aprendidos em sala de aula para criar uma solução viável utilizando tecnologia para atingir um problema existente na comunidade.

O objetivo do projeto desenvolvido é servir como uma ferramenta complementar a sistemas de ensino autodidata na área de tecnologia presentes atualmente no mercado. Baseando-se na teoria da aprendizagem autodirigida, que sugere que os indivíduos podem ser responsáveis por seu próprio aprendizado e desenvolvimento (Knowles, 1975), o *software* visa auxiliar os estudantes na gestão da aprendizagem autogerida, permitindo a organização de rotinas, o gerenciamento de tarefas e a construção de um portfólio prático.

Para apresentar as informações sobre desenvolvimento do projeto e o processo de registro do *software* junto ao Instituto Nacional da Propriedade Industrial (INPI), este trabalho está organizado da seguinte forma: Seção 2, descreve os aspectos gerais do projeto, começando na Seção 2.1, com a caracterização da solução, descrevendo os fatores motivadores da ideia. Na seção 2.2 é apresentado os requisitos mais técnicos do desenvolvimento do *software* e as tecnologias aprendidas na sala de aula e aplicadas na plataforma. Na seção 2.3 é exposto como foi a experiência e processo para o desenvolvimento do *software*, apresentando as dificuldades encontradas. Na seção 3, é descrito o processo de implementação em serviço de plataforma de nuvem e testes do sistema. Na seção 4, é apresentado como foi a experiência e o processo de registro do *software*. E finalmente, na seção 5, as considerações finais deste relatório, apresentando uma síntese das principais informações envolvidas no desenvolvimento do relatório.

2. O PROJETO IMPROOV

2.1. Caracterização da Solução

A solução aborda desafios preexistentes na indústria tecnológica, oferecendo vantagens específicas para enfrentá-los. Nesse contexto, o conceito de aprendizagem autodirigida, amplamente discutido por autores como Malcolm Knowles, manifesta-se em nossa solução por meio dos seguintes aspectos:

- **Resolução da Escassez de Profissionais Qualificados:** A plataforma atua na resolução da escassez de talentos não apenas capacitando o usuário, mas preparando-o para as exigências dos processos de recrutamento através da simulação de desafios reais e da validação de competências técnicas (hard skills) exigidas pela indústria.
- **Auxílio na Transição para a Área de Tecnologia:** Reconhecendo a necessidade de apoio para profissionais em transição de carreira, nossa solução oferece meios que facilitam o *networking* com usuários mais experientes, que podem agir como mentores incentivando e orientando a aprendizagem colaborativa, através, por exemplo, da funcionalidade Times.
- **Suporte a Desenvolvedores Iniciantes:** Para os desenvolvedores iniciantes, nossa plataforma disponibiliza missões que podem guiá-los a buscar conhecimento de forma proativa e o auxilia em seu processo de aperfeiçoamento através de análises de desempenho, promovendo que o estudante identifique quais são suas necessidades de aprendizado e busque de maneira independente por recursos e experiências necessárias para atendê-las.

Quanto às Aplicações do *software*, podemos atingir os seguintes nichos:

- **Instituições de Ensino:** Nossa solução tem a capacidade de auxiliar no ensino superior e o treinamento técnico de maneira complementar, uma vez que nossa aplicação pode contribuir significativamente para construção de conhecimento de estudantes de forma autônoma e gamificada.

- **Empresas de Tecnologia:** A nossa plataforma pode ser utilizada para treinamento e aprimoramento de equipes de tecnologia, permitindo que os funcionários adquiram habilidades relevantes de forma autodirigida. Dessa forma, alinhando-se com as tendências atuais de aprendizado corporativo, que valorizam a autonomia e proatividade do funcionário.
- **Processo Seletivo:** O *software* também pode ser aproveitado na avaliação de habilidades de candidatos, oferecendo uma visão para o recrutador das capacidades técnicas e motivação de cada usuário, através da funcionalidade de ranqueamento.

Além disso, vale ressaltar que a nossa solução possui um potencial significativo para monetização por meio de diferentes modelos de negócios, como assinaturas e parcerias estratégicas com instituições interessadas. Permitindo atender uma variedade de públicos, ampliando nosso alcance e impacto no mercado de aprendizado na área de tecnologia. Na figura 1 a nossa tela inicial apresenta as principais forças da aplicação.

Figura 1: Tela Inicial da aplicação.

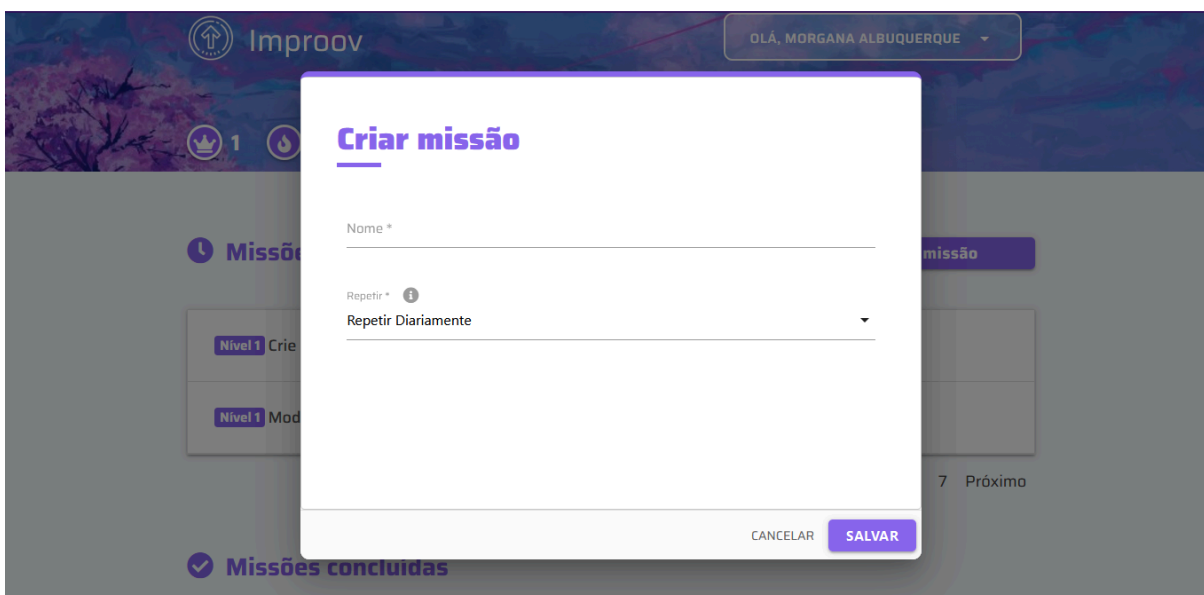


Fonte: Própria Autoria, 2023

2.2. Tecnologia Envolvida

Durante o desenvolvimento da solução, aplicamos a linguagem de programação PHP e a tecnologia Laravel, *framework* PHP de código aberto para desenvolvimento de aplicativos responsivos, por oferecer um conjunto de ferramentas e serviços que facilitam a criação de software e utilizar o padrão de arquitetura MVC (Modelo-Visão-Controle). Para fornecer os dados e estatísticas dos usuários, utilizamos a API (*Application Programming Interface*) do GitHub e para produzirmos a interface gráfica, foi utilizado os *frameworks* Bootstrap e Materialize. Além disso, o MySQL foi escolhido como sistema gerenciador de banco de dados por ser uma base de dados relacional, apresentar a ferramenta adicional de Interface gráfica MySQL Workbench para manipulação do banco de dados e possuir fácil integração com a linguagem PHP. A Figura 2 mostra um exemplo de interface desenvolvida no projeto utilizando elementos do Bootstrap e Materialize.

Figura 2: Tela de Criar Missão



Fonte: Própria Aatoria, 2023

Para o desenvolvimento do *software*, fizemos uso do editor de código Visual Studio Code (VS Code), por ter recursos que agilizam o processo de criação do código fonte, como múltipla seleção de texto, auto completar, terminal de *prompt* de comando integrado, avisos de erro de sintaxe do código e diversos *plugins*

disponíveis, assim como da plataforma de gerenciamento e armazenamento de código GitHub, pois apresenta um plano básico e gratuito, em que é possível ter ferramentas que auxiliam na organização de projetos, como a visualização de quadros *Kanban*, e o gerenciamento e definição de prazos para as tarefas a serem solucionadas.

2.3. Desenvolvimento do Projeto Improov

O desenvolvimento do projeto Improov foi realizado em três fases principais, cada uma focada em aspectos específicos do sistema para assegurar uma integração eficaz com o GitHub e proporcionar uma experiência enriquecedora aos usuários. Esta seção detalha cada uma dessas fases, desde a concepção inicial até a implementação de funcionalidades avançadas.

2.3.1. Análise e Levantamento de Requisitos

De acordo com Ian Sommerville (2011), os requisitos de um *software* são "as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento" (SOMMERVILLE, 2011, p. 57). Esses requisitos orientaram o design, a implementação e a verificação do sistema, além de garantir que o produto final atenda às expectativas dos usuários.

Os requisitos de *software* são frequentemente categorizados com requisitos funcionais, que descrevem as funcionalidades específicas do sistema, e requisitos não funcionais, que especificam critérios de qualidade, como desempenho, disponibilidade e segurança (SOMMERVILLE, 2011, p. 59).

- **Requisitos Funcionais:** Esta seção apresenta o conjunto de requisitos funcionais especificados para o produto e as suas prioridades. A prioridade de um requisito é classificada como essencial (crucial para o funcionamento), importante (sistema pode ser usado, mesmo que de forma não satisfatória) e desejável (a aplicação pode existir sem ele). A Tabela 2 especifica os requisitos funcionais implementados no sistema.

Tabela 1: Requisitos funcionais da ferramenta.

Código	Título	Descrição	Prioridade
RF001-RF003	Autenticação e Acesso com o GitHub	Permitir login e cadastro utilizando conta do GitHub. Listar projetos do usuário.	Essencial
RF004-RF006	Favoritar e visualizar projetos	Permitir ao usuário marcar projetos como favoritos, visualizar detalhes e acessar no GitHub.	Desejável
RF007-RF009	Nível de progresso	Implementar sistema de níveis e experiência ao concluir missões.	Importante
RF010-RF016	Gerenciar missões	Permitir ao usuário criar, editar e concluir missões.	Essencial
RF017-RF022	Facilitar a interação entre usuários	Permitir fazer solicitações de amizade e visualizar uma lista de amigos.	Desejável
RF023	Sair do sistema	Permitir que o usuário encerre a sessão.	Essencial
RF024-RF030	Análise de Desempenho	Gerar relatórios e gráficos que analisam em detalhe a atividade do usuário, incluindo métricas de tempo de foco, frequência de commits e missões concluídas.	Essencial
RF031-RF033	Gerenciar missões de equipe	Permitir a criação, edição e exclusão de missões específicas para a equipe.	Importante
RF034-RF038	Gerenciar membros da equipe	Permitir adicionar, remover e editar membros da equipe. Definir papéis para os membros (participante, administrador, proprietário) com diferentes níveis de permissão.	Desejável
RF039	Ranking da equipe	Criar um ranking interno para a equipe, exibindo os membros mais engajados com base na conclusão de missões.	Importante

Fonte: Própria Autoria, 2023

- **Requisitos Não Funcionais**

Esta seção apresenta o conjunto de requisitos não-funcionais, que se tratam de requisitos voltados para a sua arquitetura e o desempenho da ferramenta ao ser utilizada pelo usuário. A Tabela 3 especifica estes requisitos.

Tabela 2: Requisitos não funcionais da ferramenta.

Código	Título	Descrição	Prioridade
NF001	Interface intuitiva	A interface deve ser fácil de usar e agradável esteticamente.	Essencial
NF002	Disponibilidade do sistema	O sistema deve estar funcionando e disponível pelo menos 90% do tempo de forma clara e objetiva, sem erros para utilização dos usuários.	Importante
NF003	Proteção contra ataques	O sistema deve ser protegido contra ataques como CSRF, que ocorre quando outras pessoas conseguem se passar pelo usuário legítimo.	Essencial
NF004-NF006	Conjunto de tecnologias utilizadas	O sistema será desenvolvido utilizando PHP (versão 7.0 ou superior) e Laravel, com banco de dados MySQL.	Desejável

Fonte: Própria Aatoria, 2023

2.3.2. Primeira Fase: Integração Inicial com GitHub e Desenvolvimento de Missões

A primeira fase do desenvolvimento foi marcada pela integração inicial com o GitHub através de sua API pública, permitindo o acesso a dados públicos dos usuários. As principais funcionalidades implementadas nesta fase incluíram:

- **Integração com a API do GitHub:** Acesso aos dados públicos como repositórios, commits e contribuições.
- **Desenvolvimento de Missões Sistêmicas e do Usuário:** Implementação de missões geradas pelo sistema e missões criadas pelos próprios usuários.
- **Análise de Desempenho:** Utilização de gráficos para analisar o desempenho dos desenvolvedores, destacando o número de projetos, tecnologias utilizadas, entre outros.
- **Projetos Favoritos:** Funcionalidade que permite aos usuários marcar projetos favoritos para fácil acesso.
- **Sistema de Níveis:** Conclusão de missões geradas pelo sistema, proporcionando experiência e permitindo a progressão de nível dentro do sistema.

2.3.2.1. Integração com a API do GitHub

A integração com o GitHub foi realizada por meio da API REST pública da plataforma, utilizando requisições HTTP autenticadas para acesso aos endpoints de usuários e repositórios. A aplicação consome dados em formato JSON, incluindo informações como número de repositórios públicos, commits recentes, linguagens mais utilizadas, contribuições e atividades registradas no perfil do usuário. Esses dados são processados e armazenados no sistema para geração de métricas, análise de desempenho e criação de missões automáticas baseadas na atividade do desenvolvedor.

2.3.2.2. Desenvolvimento de Missões

As missões foram desenvolvidas para incentivar os usuários a completar tarefas específicas relacionadas ao uso do GitHub e ao desenvolvimento de software. Foram implementadas duas categorias principais de missões:

- **Missões Sistêmicas:** Geradas automaticamente com base na análise das atividades do usuário no GitHub, como criação de repositórios, realização de commits, abertura de pull requests e participação em issues.
- **Missões de Usuário:** Criadas pelos próprios usuários, permitindo personalização das tarefas e desafios, como estabelecer metas de commits semanais, criação de projetos em determinada linguagem, participação em projetos open source ou conclusão de funcionalidades específicas dentro de um repositório.

A conclusão das missões sistêmicas oferece aos usuários experiência e permite a progressão de nível dentro do sistema, incentivando o aprendizado contínuo e a participação ativa na plataforma.

2.3.2.3. Análise de Desempenho e Projetos Favoritos

A análise de desempenho foi implementada por meio de gráficos estatísticos desenvolvidos com biblioteca de visualização (Chart.js), permitindo a exibição de métricas como número de repositórios, frequência de commits por período, linguagens mais utilizadas e taxa de conclusão de missões. Os dados são coletados via API do GitHub e armazenados no banco de dados para processamento e

geração de indicadores de desempenho individual. Além disso, a funcionalidade de projetos favoritos permite aos usuários marcar e acessar rapidamente seus repositórios preferidos, facilitando a navegação e o gerenciamento de projetos dentro do Improov.

2.3.3. Segunda Fase: Criação de Times e Expansão das Funcionalidades

A segunda fase do desenvolvimento focou na criação e gestão de times, expandindo significativamente as funcionalidades do Improov. As principais implementações desta fase incluíram:

- **Criação e Gerenciamento de Times:** Implementação de times onde os usuários podem colaborar e competir.
- **Sistema de Missões de Times:** Administradores de times podem criar missões que são distribuídas a todos os membros do time.
- **Hierarquia de Cargos nos Times:** Implementação de controle de permissões baseado em papéis (Role-Based Access Control – RBAC), estruturado em três níveis: Proprietário, Administrador e Usuário. O Proprietário possui permissão total sobre o time, incluindo criação e exclusão; o Administrador pode criar e validar missões; e o Usuário pode apenas executar missões e acompanhar seu desempenho. As permissões são controladas no backend por meio de regras de autorização implementadas no sistema.
- **Validação de Missões:** Missões concluídas pelos usuários são validadas por Administradores ou Proprietários, proporcionando experiência ao usuário após a validação.
- **Sistema de Ranqueamento:** O sistema de ranqueamento é baseado na pontuação acumulada por meio da conclusão de missões, sendo que cada missão possui um valor de experiência (XP) definido conforme seu nível de dificuldade. A pontuação é atualizada automaticamente após validação da missão e exibida em ranking interno do time, ordenado de forma decrescente.
- **Vínculo de Amizades:** Funcionalidade que permite o envio e aceitação de solicitações de amizade entre usuários. O vínculo é estabelecido de forma bidirecional após confirmação, permitindo visualização de desempenho, comparação de métricas e participação conjunta em desafios.

2.3.3.1. Criação e Gerenciamento de Times

A funcionalidade de criação e gerenciamento de times foi desenvolvida para fomentar a colaboração e a competição entre os usuários. Os times podem ser criados por qualquer usuário autenticado, que automaticamente assume o papel de proprietário. Cada time possui apenas um proprietário, podendo haver múltiplos Administradores e Usuários. O sistema permite a inclusão e remoção de membros mediante convite ou aprovação, criar missões e definir metas de equipe.

2.3.3.2. Sistema de Missões de Times e Hierarquia de Cargos

Dentro dos times, foi implementado um sistema de missões específico, onde os Administradores podem criar missões que são distribuídas a todos os membros. A hierarquia foi implementada por meio de um sistema de controle de permissões no backend, no qual cada perfil possui ações específicas autorizadas. O controle é realizado por verificação de papéis no momento da requisição, impedindo que usuários executem operações fora de suas permissões, como validação de missões ou exclusão de times.

2.3.3.3. Validação de Missões e Sistema de Ranqueamento

A validação das missões ocorre de forma manual pelos Administradores ou Proprietários do time, que analisam evidências apresentadas pelo usuário, como links de repositórios, commits realizados ou pull requests abertos. Em missões sistêmicas baseadas na API do GitHub, a validação pode ocorrer automaticamente mediante verificação do cumprimento de critérios objetivos, como número mínimo de commits ou criação de repositório em determinado período.

2.3.3.4. Vínculo de Amizades

A funcionalidade de vínculo de amizade foi introduzida para fortalecer a comunidade dentro do Improov. Os usuários podem estabelecer vínculos de amizade, o que facilita a colaboração e a comunicação entre desenvolvedores. Este recurso promove um ambiente de suporte mútuo e networking, essencial para o desenvolvimento profissional e pessoal dos usuários.

2.3.4. Terceira Fase: Integração de Dados Privados e Webhooks

A terceira fase envolveu a ampliação da integração com o GitHub para incluir dados privados e a utilização de Webhooks para monitoramento em tempo real. As principais implementações desta fase incluíram:

- **Solicitação de Acesso a Escopos Privados:** Modificação da integração para solicitar e utilizar dados privados dos usuários, como repositórios privados.
- **Reestruturação do Projeto:** Ajustes na arquitetura do sistema para suportar o acesso a dados privados, garantindo a segurança e privacidade das informações.
- **Implementação de Webhooks:** Utilização de Webhooks do GitHub para receber notificações em tempo real sobre atividades dos usuários, como commits e pull requests.
- **Monitoramento em Tempo Real:** Monitoramento contínuo das atividades dos usuários no GitHub, proporcionando feedback imediato e preciso.

2.3.4.1. Solicitação de Acesso a Escopos Privados

Para obter uma análise mais completa e personalizada, tornou-se necessário acessar dados privados dos usuários. A solicitação de acesso a escopos privados foi um passo crucial, envolvendo a conformidade com as políticas de segurança e privacidade do GitHub. Esta integração permitiu ao Improov acessar repositórios privados e outras informações restritas, proporcionando uma visão mais abrangente das atividades dos usuários.

2.3.4.2. Reestruturação do Projeto

Inicialmente, o projeto Improov realizava integração com a API REST do GitHub utilizando apenas dados públicos, limitando o monitoramento às informações disponíveis sem autenticação avançada. Essa abordagem restringia a análise de atividades em repositórios privados e reduzia as possibilidades de personalização das métricas dos usuários.

Para ampliar as funcionalidades do sistema, foi implementado o protocolo OAuth 2.0, permitindo autenticação segura e concessão de permissões específicas

por meio de escopos (scopes). Após análise da documentação oficial do GitHub, foram selecionados os escopos **repo**, para acesso a repositórios privados, para obtenção de informações detalhadas do perfil, para identificação segura do usuário na aplicação e outros que atendem às nossas necessidades.

A adoção desses escopos seguiu o princípio do menor privilégio, concedendo apenas as permissões estritamente necessárias ao funcionamento do sistema. A autenticação passou a gerar tokens de acesso utilizados nas requisições autenticadas à API, possibilitando a coleta de dados mais detalhados sobre commits, pull requests e atividades em repositórios privados.

Essa reestruturação foi fundamental para aprimorar o monitoramento das atividades dos usuários, permitindo análises mais precisas, geração de métricas personalizadas e validação automatizada de missões, mantendo a conformidade com as boas práticas de segurança recomendadas pelo GitHub.

O projeto Improov inicialmente acessava apenas dados públicos da API do GitHub, o que restringia nosso alcance em termos de monitoramento e personalização das interações com os usuários. Para tornar o sistema mais eficaz e ajudar os usuários de forma mais completa, percebemos que seria essencial acessar dados privados do GitHub, como informações mais detalhadas sobre as atividades nos repositórios.

Com isso em mente, exploramos alternativas para obter esses dados com segurança e eficiência. Após avaliar a documentação oficial do GitHub, identificamos que o sistema oferecia diferentes *roles* e *scopes* que poderiam ser configurados para conceder permissões adicionais ao nosso sistema, mantendo a privacidade e segurança dos usuários. Optamos por selecionar e aplicar os escopos que atendem, exatamente, as necessidades do projeto, uma escolha que resolveu a questão do acesso aos dados privados sem aumentar a complexidade.

Esse processo de reestruturação do Improov foi fundamental para possibilitar um monitoramento mais preciso, garantindo que o sistema pudesse continuar funcionando de forma otimizada e segura, respeitando os limites de acesso definidos e alinhados com as práticas recomendadas pelo GitHub.

2.3.4.3. Implementação de Webhooks

Durante o desenvolvimento, foi identificado que a atualização em tempo real dos dados dos usuários seria fundamental para a experiência do Improov. Inicialmente, buscou-se alcançar essa funcionalidade por meio do consumo direto da API REST do GitHub. No entanto, essa abordagem apresentou limitações. Para obter dados em tempo real, seriam necessárias requisições constantes à API. Essa prática revelou-se inviável devido ao rate limit da API do GitHub, que define um limite máximo de requisições por hora. Quando esse limite é atingido, a aplicação fica indisponível, impossibilitando o acesso aos dados de forma contínua.

Diante desse problema, consideramos cinco alternativas técnicas antes de implementar os Webhooks:

- **Polling com Intervalos Programados:** Configurar o sistema para realizar consultas à API em intervalos fixos. Essa solução reduziria o consumo de requisições, mas ainda estaria sujeita ao limite de requisições da API e não garantiria atualizações em tempo real.
- **Utilização da API GraphQL:** A API GraphQL do GitHub permite otimizar as consultas, consolidando as informações necessárias em uma única requisição. Apesar de aumentar a eficiência, essa abordagem ainda estaria limitada pelo rate limit, tornando inviável o monitoramento contínuo.
- **Cache Local:** Implementar um sistema de cache para armazenar temporariamente os dados, reduzindo o número de requisições. Contudo, essa abordagem comprometeria a precisão dos dados em tempo real, pois as atualizações dependeriam do tempo de atualização do cache.
- **Serviços de Terceiros para Sincronização Contínua:** Avaliamos também soluções de terceiros que oferecem sincronização contínua com o GitHub. Porém, esses serviços, além de terem custo elevado, não permitiam a customização necessária ao Improov.
- **Banco de Dados com Logs Incrementais:** Criar um banco de dados local com logs incrementais de atividades. Essa alternativa permitiria registrar modificações com base em horários específicos, mas traria complexidade adicional para sincronizar os eventos em tempo real sem uma integração nativa com a API.

Após a análise dessas alternativas, optamos pela implementação de Webhooks do GitHub, que se mostrou a solução mais apropriada para o objetivo do

projeto. Os Webhooks são mecanismos que permitem o envio automático de notificações para o sistema sempre que eventos específicos ocorrem no GitHub, como commits, pull requests ou issues. Conforme explicado na documentação oficial do GitHub, essa funcionalidade possibilita uma atualização em tempo real dos dados do usuário sem a necessidade de fazer múltiplas requisições à API, evitando, assim, o risco de ultrapassar o rate limit.

A implementação de Webhooks garantiu uma sincronização contínua das atividades dos desenvolvedores, proporcionando maior eficiência e reatividade ao Improov, ao mesmo tempo em que preservou o desempenho do sistema.

2.3.4.4. Monitoramento em Tempo Real

Com a implementação de Webhooks, o monitoramento das atividades dos usuários no GitHub passou a ser contínuo e em tempo real. Esta funcionalidade proporcionou feedback imediato e preciso aos usuários, permitindo uma análise detalhada e atualizada de suas atividades no GitHub. O monitoramento em tempo real é fundamental para oferecer um suporte personalizado e eficaz ao aprendizado de programação.

3. IMPLANTAÇÃO E TESTES

Com o propósito de assegurar a qualidade do sistema e aplicar os conhecimentos adquiridos na disciplina de "Testes e Qualidade de Software", elaboramos o documento de plano de testes. Este processo envolveu o planejamento, redação e implementação de casos de teste com base nos requisitos do sistema.

A documentação desempenhou um papel fundamental ao manter os desenvolvedores do projeto informados sobre o cronograma e os recursos essenciais que deveriam ser implementados no software. Isso possibilitou a obtenção de resultados periódicos e automatizados durante o processo de desenvolvimento do sistema.

A seguir, apresentaremos as tabelas que contêm os requisitos a serem testados, também conhecidos como casos de teste, relacionados às principais ações dos usuários.

As tabelas apresentam casos de teste, estruturado com os seguintes campos: o **identificador** (CT00X) é o código único que diferencia cada caso de teste; a

Descrição explica o objetivo do teste; os **Requisitos Associados** indicam qual requisito funcional está sendo validado (como: RF00X); as **Pré-Condições** definem o que deve estar configurado antes de executar o teste; as **Pós-Condições** descrevem o estado esperado do sistema após a execução; os **Critérios de Sucesso** definem quando o teste é considerado aprovado; a **Entrada** lista os dados necessários para realizar o teste; o **Procedimento** descreve o passo a passo da execução; a **Técnica** indica o método utilizado para executar o teste; a **Abordagem Caixa Preta** significa que o teste avalia apenas o comportamento externo do sistema, sem considerar o código interno; e o **Estágio do Teste** indica em qual fase do processo de testes aquela funcionalidade está sendo verificada.

Tabela 3: Caso de Teste Realizar Cadastro via GitHub.

[CT001] REALIZAR CADASTRO VIA GITHUB	
Descrição: Para acessar as funcionalidades do sistema como as missões, análise de desempenho, ranking, etc. É necessário que o usuário realize o cadastro. O cadastro é feito através da plataforma do GitHub e ao se cadastrar você terá acesso ao sistema.	
Requisito(s) Associados: RF001	
Pré-Condições: Ter um e-mail válido.	Pós-Condições: Estar na tela inicial do sistema autenticado.
Critérios de sucesso: Após criar o cadastro, o sistema deve voltar para a página inicial autenticada. Deve-se permitir acesso total do sistema após o cadastro.	
Entrada: <ul style="list-style-type: none"> ● Usuário: Sequência de caracteres. ● E-mail: Sequência de caracteres que tenha o "@" e algum caractere depois do do "@". ● Senha: No mínimo 8 caracteres incluindo números e no mínimo uma letra maiúscula, podendo ter no máximo 15 caracteres. 	Procedimento: <ol style="list-style-type: none"> 1. <i>Acesse o sistema em um navegador .</i> 2. <i>Selecione no botão "Comece a programar".</i> 3. <i>Selecione no botão "Não tenho conta".</i> 4. <i>Insira a entrada no formulário e clique em "Create Account" .</i> 5. <i>Você será direcionado para a tela inicial do sistema.</i>
Técnica: Manual	Abordagem do teste: Caixa Preta
Responsáveis: Programadores (ou equipe subcontratada)	Estágio do Teste: Integração

Fonte: Própria Autoria, 2023

Tabela 4: Caso de Teste Realizar Autenticação via GitHub.

[CT002] REALIZAR AUTENTICAÇÃO VIA GITHUB		
Descrição: Para acessar as funcionalidades do sistema como as missões, análise de desempenho, ranking, etc. É necessário que o usuário esteja autenticado. A autenticação é feita através da plataforma do GitHub e ao efetuar-la o usuário terá acesso ao sistema.		
Requisito(s) Associados: RF001, RF002		
Pré-Condições: Estar cadastrado no sistema.	Pós-Condições: Estar na tela inicial autenticada do sistema.	
Critérios de sucesso: Após efetuar a autenticação o sistema deve voltar para a página inicial autenticada. Deve-se permitir acesso total do sistema após o login.		
Entrada: <ul style="list-style-type: none"> • E-mail: Inserir e-mail já cadastrado no sistema. • Senha: Inserir a senha que está vinculada ao e-mail. 	Procedimento: <ol style="list-style-type: none"> 1. <i>Acesse o sistema em um navegador.</i> 2. <i>Selecione no botão de "Comece a programar".</i> 3. <i>Selecione no botão "Já tenho conta".</i> 4. <i>Insira a entrada no formulário e clique em "Sign In".</i> 5. <i>Você será direcionado para a tela inicial do sistema.</i> 	
Técnica: Manual	Abordagem do teste: Caixa Preta	
Responsáveis: Programadores (ou equipe subcontratada)	Estágio do Teste: Integração	

Fonte: Própria Autoria, 2023

Tabela 5: Caso de Teste Criar Missões

[CT003] CRIAR MISSÃO		
Descrição: Nesta iteração será testado a criação de uma missão personalizada do usuário onde o principal objetivo é engajar o usuário na plataforma e ajudá-lo a gerenciar suas tarefas.		
Requisito(s) Associados: RF002, RF010		
Pré-Condições: Estar autenticado no sistema.	Pós-Condições: Estar na tela de missões e ter a missão criada na parte de "Missões Pendentes" e a mesma deve estar com o status de ativo.	
Critérios de sucesso: Após Criar a missão deve aparecer uma mensagem de sucesso, adicionar missão na parte de "Missões pendentes" e o seu status deve estar como ativo.		

Entrada: <ul style="list-style-type: none"> • Nome: Sequência de caracteres. • Repetir: Selecionar uma das opções. 	Procedimento: <ol style="list-style-type: none"> 1. Efetue o login no sistema. 2. Selecione no botão de “Olá, {seu nome}”. 3. Selecione “Minhas Missões”. 4. Listará todas as missões, dividindo em partes de missões pendentes e concluídas. 5. Clique em “Criar Missão”. 6. Insira a entrada no formulário e clique em “Salvar”. 7. Deve aparecer uma mensagem de sucesso e ter a missão criada na parte de “Missões Pendentes” e a mesma deve estar com o status de ativo.
Técnica: Manual	Abordagem do teste: Caixa Preta
Responsáveis: Programadores (ou equipe subcontratada)	Estágio do Teste: Unidade

Fonte: Própria Autoria, 2023

Tabela 6: Caso de Teste Realizar Missões

[CT004] REALIZAR MISSÃO	
Descrição: Nesta iteração deve ser realizado o teste do fluxo principal do sistema, concluir uma missão ativa.	
Requisito(s) Associados: RF002, RF010, RF013, RF016	
Pré-Condições: Estar autenticado no sistema e criado uma missão.	Pós-Condições: Estar na tela de missões e ter a missão realizada na parte de “Missões concluídas” com a data de conclusão.
Crterios de sucesso: Após realizar a missão deve aparecer uma mensagem de sucesso, desativar a missão na parte de “Missões pendentes” e adicionar a missão na parte de “Missões concluídas” com a data de conclusão.	
Entrada: <ul style="list-style-type: none"> • N/A 	Procedimento: <ol style="list-style-type: none"> 1. Efetue o login no sistema. 2. Selecione no botão de “Olá, {seu nome}”. 3. Selecione “Minhas Missões”. 4. Listará todas as missões, dividindo em partes de missões pendentes e concluídas. 5. Escolha uma missão pendente e clique no botão de confirmação verde. 6. A missão selecionada deve ficar

	<i>desativada na parte de “Missões Pendentes” e na parte de “Missões Concluídas” deve ter a missão e sua data de conclusão.</i>	
Técnica: Manual	Abordagem do teste: Caixa Preta	
Responsáveis: Programadores (ou equipe subcontratada)		Estágio do Teste: Sistema

Fonte: Própria Autoria, 2023

Tabela 7: Caso de Teste Favoritar Projeto

[CT005] VISUALIZAR UM PROJETO FAVORITO		
Descrição: Nesta iteração será testado a visualização de todos os projetos/repositórios presentes na conta do GitHub do usuário onde o principal objetivo é destacar os projetos mais relevantes do mesmo.		
Requisito(s) Associados: RF002, RF003, RF005		
Pré-Condições: Estar autenticado no sistema.	Pós-Condições: Ter os projetos favoritos em destaque na tela de "Início".	
Critérios de sucesso: Após selecionar um projeto como favorito o mesmo deve aparecer na tela de "Início" na parte de “Projetos Favoritos”.		
Entrada: N/A	Procedimento: <ol style="list-style-type: none"> 1. <i>Efetue o login no sistema.</i> 2. <i>Selecione no botão “Olá, {seu nome}”.</i> 3. <i>Selecione “Meus projetos”.</i> 4. <i>Listará todos os projetos com estrelas cinzas.</i> 5. <i>Selecione um projeto para adicioná-lo aos seus projetos favoritos, e clique na estrela.</i> 6. <i>Deve aparecer uma mensagem de sucesso e a estrela deve ficar na cor amarela. E apenas poderá ter 3 projetos com a estrela amarela.</i> 7. <i>Selecione no botão “Olá, {seu nome}”.</i> 8. <i>Selecione “Inicio”.</i> 9. <i>Listará os seus projetos favoritos selecionados no item 6 na parte “Projetos Favoritos”</i> 	
Técnica: Manual	Abordagem do teste: Caixa Preta	
Responsáveis: Programadores (ou equipe subcontratada)		Estágio do Teste: Unidade

Fonte: Própria Autoria, 2023

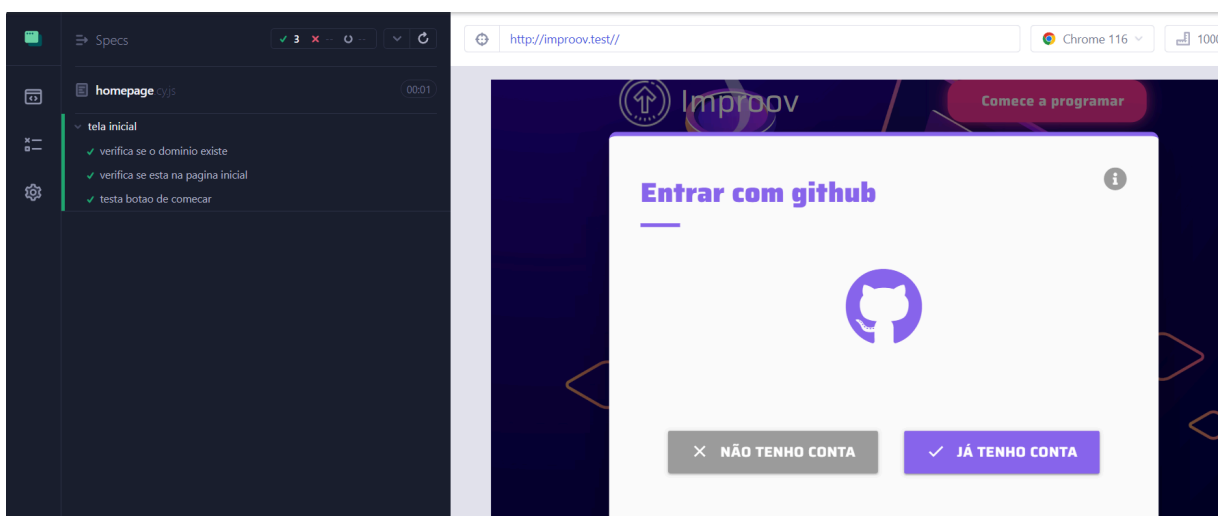
Tabela 8: Caso de Teste Ver Desempenho

[CT006] VISUALIZAR DESEMPENHO		
Descrição: Nesta iteração serão testados a visualização de gráficos informativos das principais ações do usuário, onde o principal objetivo é verificar o desempenho.		
Requisito(s) Associados: RF002, RF027, RF028, RF029, RF030, RF031, RF032		
Pré-Condições: Estar autenticado no sistema.	Pós-Condições: Visualizar a análise com o seu desempenho.	
Critérios de sucesso: Ver gráficos informativos sobre quantidade de projetos criados e linguagens utilizadas, frequência de <i>commits</i> e quantidade de missões realizadas.		
Entrada: N/A	Procedimento: <ol style="list-style-type: none"> 1. Efetue o login no sistema. 2. Selecione no botão "Olá, {seu nome}". 3. Selecione "Desempenho". 4. Você irá visualizar o seu desempenho. 	
Técnica: Manual	Abordagem do teste: Caixa Preta	
Responsáveis: Programadores (ou equipe subcontratada)		Estágio do Teste: Sistema

Fonte: Própria Aatoria, 2023

Para fazer a implementação dos testes automatizados foi utilizado o Cypress, ferramenta de teste fim-a-fim (E2E), pois possui um funcionalidades que auxiliam na inspeção de erros e gerar relatórios e evidências de teste. A Figura 3 mostra um exemplo da execução de um teste automatizado responsável por verificar a disponibilidade dos casos de teste 1 e 2.

Figura 3: Execução de teste automatizado com Cypress



Fonte: Própria Aatoria, 2023

Dessa forma, podemos garantir que as funcionalidades essenciais estivessem dentro do padrão aceitável para o uso do usuário, evitando que correções e novas funcionalidades quebrassem o sistema já disponível em ambiente de produção.

4. REGISTRO DE SOFTWARE

O processo de registro de um programa de computador no Instituto Nacional da Propriedade Industrial (INPI) é uma forma crucial para garantir a proteção legal e de direitos autorais de softwares desenvolvidos no Brasil. De acordo com o Manual do Usuário para o Registro Eletrônico de Programas de Computador publicado pelo Ministério da Economia em 2022, os passos para realização do processo são:

1. Para dar início ao registro, o titular do software ou seu procurador deve criar uma conta no portal do INPI para poder acessar os serviços online disponíveis pelo instituto. Esta criação da conta será fundamental para iniciar o processo de registro e acompanhar o andamento do processo ao longo do tempo.
2. Na etapa seguinte, o solicitante deve emitir e realizar o pagamento da Guia de Pagamento União (GRU), que são as taxas associadas ao registro do programa, e fazer o *download* e assinar digitalmente a Declaração de Veracidade (DV), seguindo o padrão de assinatura PAdES (PDF Advanced Electronic Signatures). Para assinar a Declaração de Veracidade corretamente, é necessário ter previamente um certificado digital de pessoa física (e-CPF) ou de pessoa jurídica (e-CNPJ). Estes certificados podem ser adquiridos através das autoridades certificadoras (ACs) credenciadas pela ICP-Brasil. A lista de ACs credenciadas encontra-se disponível no portal do Instituto Nacional de Tecnologia da Informação – ITI; talvez a autoridade certificadora mais acessível seja os Correios, para adquirir o e-CPF ou e-CNPJ é necessário ir presencialmente a uma agência portando os seus documentos de identificação, o processo de agendamento de visita a agência pode ser feito pelo site dos Correios.
3. Após efetuar o pagamento da GRU, acesse o petição eletrônico e preencha o formulário de Pedido de Registro de Programa de Computador. Nesta etapa, é necessário fornecer informações detalhadas sobre o titular, autores do programa e o software, também é necessário informar um resumo digital *hash*, a função criptográfica hash é um algoritmo utilizado para garantir a integridade de um documento eletrônico, de modo que qualquer modificação no texto original gera um resumo hash completamente diferente, o que permite que esse mecanismo de verificação de integridade de arquivos seja largamente utilizado como prova eletrônica na forense computacional.

Antes de criptografar e compactar o código-fonte, procure acrescentar no arquivo o máximo de documentação técnica e os dados que demonstrem a inovação e exclusividade do programa. Adicionalmente, é necessário anexar ao peticionamento a DV assinada eletronicamente, que deverá ser autenticado com certificado digital certificado pela ICP-Brasil.

4. Antes de submeter o formulário, é importante verificar se o pagamento da sua GRU foi processado corretamente. Após verificação, o pedido será submetido e o solicitante terá de aguardar a publicação da Expedição do Certificado de Registro na Revista da Propriedade Industrial; após a emissão do certificado, a consulta pode ser feita diretamente junto à Base de Dados do INPI.

Seguindo os passos descritos acima, fizemos o registro de *software* do Improov. A expedição do certificado foi feita na revista de número 2766, publicada em 09/01/2024, seção VII - programa de computador. O Certificado de Registro de Programa de Computador constitui uma confirmação oficial de documentação e garante a proteção legal de um programa de computador, concedendo ao seu titular o direito exclusivo sobre a reprodução, distribuição e comercialização do *software*. Apesar deste processo exigir algumas etapas, é necessário proteger os direitos autorais desta criação intelectual no âmbito dos direitos de propriedade industrial brasileira.

5. CONSIDERAÇÕES FINAIS

O desenvolvimento deste projeto proporcionou a aplicação prática de grande parte do conteúdo teórico e das habilidades técnicas adquiridas em sala de aula. Além disso, possibilitou o desenvolvimento de habilidades não técnicas cruciais no mercado de trabalho de desenvolvimento de sistemas. Isso incluiu a compreensão dos rituais e das rotinas diárias de uma equipe de desenvolvimento, o aprendizado sobre como conduzir o levantamento de requisitos e a elaboração de documentação, a gestão eficaz do tempo, o trabalho colaborativo e a priorização de atividades, entre outras competências essenciais.

Além de adquirir conhecimento sobre o ritmo de trabalho, este projeto proporcionou experiências valiosas na criação de um *pitch* e na apresentação do *software* ao público como um empreendimento real, em mais de um idioma (inglês e português). Em várias situações, a busca por ajuda em professores, fóruns, documentações online e livros foi necessária para enfrentar desafios inesperados que não haviam sido abordados anteriormente nas disciplinas.

Como resultado final, criamos um produto gamificado que fornece ferramentas para usuários se desenvolverem e se conectarem profissionalmente, em um ambiente que o auxilia a manter o foco em desenvolver habilidades necessárias para o mercado de trabalho e o incentiva a compartilhar conhecimento e se conectar com outros profissionais da área, facilitando seu processo de construção de conhecimento, portfolio e *networking*.

No geral, este projeto representou uma oportunidade enriquecedora para integrar conhecimento teórico com experiências práticas, preparando-nos de forma abrangente para os desafios e as demandas do mercado de trabalho no desenvolvimento de *software*.

6. BIBLIOGRAFIA

BRASSCOM. Relatório de Inteligência e Informação: **Formação Educacional e Empregabilidade em TIC Achados e Recomendações**. São Paulo, 2019.

Disponível em: <https://brasscom.org.br/wp-content/uploads/2021/10/BRI2-2019-010-P02-Formacao-Educacional-e-Empregabilidade-em-TIC-v83.pdf>. Acesso em: 2 out. 2023.

BRASIL. Ministério da Economia. **Manual do Usuário para o Registro Eletrônico de Programas de Computador**. [Rio de Janeiro]: Instituto Nacional Da Propriedade Industrial, 28 jul. 2022. Disponível em: <https://www.gov.br/inpi/pt-br/servicos/programas-de-computador/arquivos/manual/manual-e-software-2022.pdf>. Acesso em: 21 nov. 2023.

CARNEVALE, B. **Recrutamento e Seleção especializado em TI: Entrevista com Patrícia Duchnicky**. Disponível em: <https://factorialhr.com.br/blog/recrutamento-e-selecao-ti-entrevista/>. Acesso em: 2 out. 2023.

Centre for Excellence in Universal Design. **The 7 Principles**. Disponível em: <https://universaldesign.ie/about-universal-design/the-7-principles>. Acesso em: 09 jan. 2024.

IBGE – INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Painel PNAD Contínua**. Mercado de Trabalho, Brasil, 2021. Rio de Janeiro: IBGE, 2021. Disponível em: <https://painel.ibge.gov.br/pnadc/>. Acesso em: 30 mar. 2023.

INPI. **Revista Propriedade Industrial: Programas de Computador**. N° 2766, Seção VII. 09 jan. de 2024. Disponível em: http://revistas.inpi.gov.br/pdf/Programa_de_computador2766.pdf. Acesso em: 09 jan. 2024.

KNOWLES, M. S. **Self-directed learning: A guide for learners and teachers**. Chicago: Association Press, 1975.

MANPOWERGROUP. **Pesquisa de Escassez de Talentos**. 2023. Disponível em: <https://blog.manpowergroup.com.br/pesquisa-escassez-de-talentos-2023>. Acesso em: 01 mar. 2024.

MIT Technology Review Insights. **New approaches to the tech talent shortage**. 2023. Disponível em: <https://www.technologyreview.com/2023/09/21/1079695/new-approaches-to-the-tech-talent-shortage/>. Acesso em: 22 abr. 2024.

MOMA, G. **10 heurísticas de Nielsen para o design de interface**. Medium: UX Collective, 2017. Disponível em: <https://brasil.uxdesign.cc/10-heur%C3%ADsticas-de-nielsen-para-o-design-de-interfa-ce-58d782821840>. Acesso em: 30 mar. 2023.

NORMAN, D. A. **The Design of Everyday Things**. Nova Iorque: Basic Books, 2013.

SOMMERVILLE, I. **Engenharia De Software**. 9 ed. São Paulo: Pearson, 2011.

ANEXO A – CERTIFICADO DE REGISTRO DE SOFTWARE EMITIDO PELO INPI

IPI
Assinado
Digitalmente

REPÚBLICA FEDERATIVA DO BRASIL
MINISTÉRIO DO DESENVOLVIMENTO, INDÚSTRIA, COMÉRCIO E SERVIÇOS
INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL
DIRETORIA DE PATENTES, PROGRAMAS DE COMPUTADOR E TOPOGRAFIAS DE CIRCUITOS

Certificado de Registro de Programa de Computador

Processo Nº: **BR512024000002-0**

O Instituto Nacional da Propriedade Industrial expede o presente certificado de registro de programa de computador, válido por 50 anos a partir de 1º de janeiro subsequente à data de 25/11/2023, em conformidade com o §2º, art. 2º da Lei 9.609, de 19 de Fevereiro de 1998.

Título: Improov

Data de publicação: 25/11/2023

Data de criação: 25/02/2022

Titular(es): MORGANA FERNANDES DE ALBUQUERQUE

Autor(es): MORGANA FERNANDES DE ALBUQUERQUE; ERICKSON FERREIRA DA SILVA NASCIMENTO

Linguagem: PHP

Campo de aplicação: ED-04

Tipo de programa: AV-01; DS-06; FA-04

Algoritmo hash: SHA-512

Resumo digital hash:

8985310BA4EC38EA2A3CE2BBA6DB89004F3909B0A842AB25D6B8F5099671574806EFF2EE14CA77C698925CCA
AC5B0B4F5AC4E54C11F996BB80A01BDF159D18C8

Expedido em: 09/01/2024

Aprovado por:

Carlos Alexandre Fernandes Silva
Chefe da DIPTO