



REPÚBLICA FEDERATIVA DO BRASIL
MINISTÉRIO DA ECONOMIA
INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL
DIRETORIA DE PATENTES, PROGRAMAS DE COMPUTADOR E TOPOGRAFIAS DE CIRCUITOS INTEGRADOS

Certificado de Registro de Programa de Computador

Processo Nº: **BR512022001663-0**

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 04/04/2022, em conformidade com o §2º, art. 2º da Lei 9.609, de 19 de Fevereiro de 1998.

Título: RepositORE

Data de publicação: 04/04/2022

Data de criação: 04/04/2022

Titular(es): FUNDAÇÃO UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE - FUERN

Autor(es): SEBASTIÃO EMÍDIO ALVES FILHO; THALIA KATIANE SAMPAIO GURGEL; KEFTON DAVID NUNES DE MELO; CLEYTON CARLOS SANTOS COSTA

Linguagem: HTML; JAVA SCRIPT; OUTROS

Campo de aplicação: ED-03; IF-02; IF-04

Tipo de programa: AP-01; GI-01

Algoritmo hash: SHA-512

Resumo digital hash:

680326749e0c6e3dc00a542ffe75ba67a79fe9500535c4291c2508941075fb13a38c7be1104e9be646941dacb74454d08497951e5b00ab0a6ead821b745d43c0

Expedido em: 12/07/2022

Aprovado por:

Joelson Gomes Pequeno

Chefe Substituto da DIPTO - PORTARIA/INPI/DIRPA Nº 02, DE 10 DE FEVEREIRO DE 2021

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE – UERN

FACULDADE DE CIÊNCIAS EXATAS E NATURAIS – FANAT

DEPARTAMENTO DE INFORMÁTICA – DI

CLEYTON CARLOS SANTOS COSTA

REPOSITORE: PERSPECTIVA DO FRONT-END

MOSSORÓ - RN

2022

CLEYTON CARLOS SANTOS COSTA

REPOSITORE: PERSPECTIVA DO FRONT-END

Relatório apresentado ao curso de Ciência da Computação da Universidade do Estado do Rio Grande no Norte como requisito da disciplina de Trabalho de Diplomação, sob a orientação do(a) Prof(a). Sebastião Emidio Alves Filho

MOSSORÓ - RN

2020

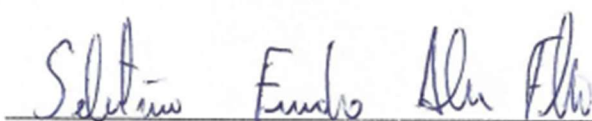
CLEYTON CARLOS SANTOS COSTA

REPOSITORE: PERSPECTIVA DO FRONT-END

Registro de software apresentado como pré-requisito para a obtenção do título de Bacharel em Ciência da Computação da Universidade do Estado do Rio Grande do Norte – UERN, submetida à aprovação da banca examinadora composta pelos seguintes membros:

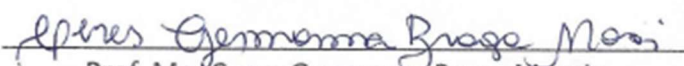
Aprovada em: 27/04/2022

Banca Examinadora



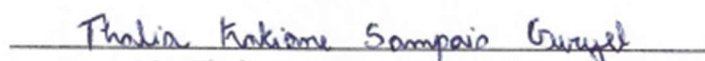
Prof. Dr. Sebastião Emídio Alves Filho

Universidade do Estado do Rio Grande do Norte - UERN



Prof. Me. Ceres Germana Braga Moraes

Universidade do Estado do Rio Grande do Norte - UERN



Me. Thalia Katiane Sampaio Gurgel

Universidade do Estado do Rio Grande do Norte - UERN

SUMÁRIO

1	INTRODUÇÃO	5
1.1	JUSTIFICATIVA	5
1.2	OBJETIVOS	5
1.3	TRABALHOS RELACIONADOS	6
2	ARQUITETURA DO SISTEMA	6
2.1	DIAGRAMA DE COMPONENTES	8
2.2	DIAGRAMA DE CLASSES	10
2.3	DIAGRAMAS DE CASO DE USO	12
2.4	DESCRIÇÃO REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS.....	13
3	IMPLEMENTAÇÃO	14
3.1	METODOLOGIA DE DESENVOLVIMENTO	16
3.2	FERRAMENTAS E BIBLIOTECAS UTILIZADAS	17
3.3	CAPTURAS DE TELA.....	18
4	TESTES E VALIDAÇÃO	23
5	PERSPECTIVAS FUTURAS.....	24
6	REFERENCIAS BIBLIOGRÁFICAS.....	25

1 INTRODUÇÃO

A aplicação da robótica ao ensino de alunos, por meio de estudos, programação, montagem e implementação da robótica através de kits de robótica, denomina-se robótica educacional. A robótica educacional tem como objetivo permitir o desenvolvimento do conhecimento teórico e prático do aluno, como também ajuda no desenvolvimento de outras habilidades cognitivas, tais como produtividade, interação, resolução de problemas, autonomia e etc.

Os diversos kits de robótica existentes são compostos de motores, sensores, linguagens de programação e etc, possibilitando uma grande variedade de tipos de atividades de robótica aplicáveis ao ensino.

1.1 JUSTIFICATIVA

Uma pesquisa realizada por Gurgel(2019) acabou revelando que existem muitos conteúdos de robótica na internet, porém a grande maioria desses conteúdos estão dispersos de forma descentralizada e desorganizada, sendo constatado também que existe um déficit de plataformas que possuam descrição de campos através da utilização de metadados, o que dificulta mais ainda o reconhecimento de conteúdos de robótica por parte dos motores de busca, que por sua vez obscurece o ato de achar conteúdos de robótica por parte dos usuários.

Levando em consideração a escassez de plataformas especializadas com descrição de metadados em campos de objetos de aprendizagem de robótica educacional, originou-se a ideia de criar a plataforma RepositORE.

1.2 OBJETIVOS

O RepositORE é uma plataforma que tem como principal objetivo ajudar alunos, professores e apreciadores da robótica a compartilhar e encontrar conteúdos relevantes de robótica mais facilmente, ou seja, o RepositORE visa suprir a carência de plataformas com descrição de metadados, e assim suprir as necessidades dos usuários de encontrar conteúdo específico de robótica.

1.3 TRABALHOS RELACIONADOS

O RepositORE utiliza uma descrição de campos dos Objetos de Aprendizagem com metadados. Durante o estudo de Gurgel (2019) foi feito um levantamento sobre diferentes padrões de metadados, sendo que dentre os padrões selecionados para comparação, foi selecionado o padrão de metadados Dublin Core, devido a sua simplicidade, flexibilidade e extensibilidade. No decorrer do estudo, foi feita uma pesquisa e comparação de diferentes trabalhos que aplicavam o padrão de descrição de metadados Dublin Core.

Foi feito por Gonçalves *et al.*(2013) uma adaptação do padrão de metadados Dublin Core ao caso da Biblioteca Digital da Assembleia Legislativa do Estado de Minas Gerais. O padrão Dublin Core foi escolhido pelos autores devido à capacidade de adaptação e extensibilidade do padrão, permitindo uma adaptação de metadados ao Código de Catalogação Anglo Americano 2 (AACR2).

Roseto & Nogueira(2002) realizaram um trabalho que consistiu no uso do padrão Dublin Core para a descrição de conteúdos na biblioteca digital de teses da USP. O padrão acabou sendo escolhido pelos autores devido sua extensibilidade, o que permitiu a personalização do Dublin Core para a aplicação desejada pelos autores.

O padrão Dublin Core foi utilizado para a Descrição de Obras Raras na Web: A Coleção da Biblioteca Brasileira Digital, proposto por Pires(2012). Também foi feita uma adaptação do padrão às necessidades do autor.

2 ARQUITETURA DO SISTEMA

Principais funcionalidades do sistema:

- Página principal: Essa página dá uma pequena introdução ao sistema, e exibe os cinco últimos objetos de aprendizagem de robótica educacional postados por usuários no sistema. A página principal é exibida para qualquer usuário, mesmo que ele não tenha uma conta no RepositORE.
- Login: Nesta página, o usuário pode fazer login no sistema através de uma conta Google ou com uma conta de e-mail criada no sistema. Caso não possua conta

de e-mail, ele pode selecionar a opção de criar uma conta e ser redirecionado para a página de criação de conta.

- Criação de conta: Na página de criação de conta, o usuário pode criar sua conta usando um e-mail qualquer.
- Cadastro de objetos: É nessa página que o usuário pode cadastrar e postar um objeto de aprendizagem de robótica educacional com descrição de metadados. Essa página só é disponibilizada para usuários logados no sistema.
- Lista de objetos: A listagem de objetos mostra para o usuário que esteja logado no sistema todos os objetos postados por ele mesmo e por outros usuários. É possível buscar um objeto na lista por busca textual, como também é possível ordenar a listagem por objetos mais recentes ou mais antigos. Cada objeto de aprendizagem postado possui três interações disponíveis, sendo elas: mostrar detalhes, atualizar objeto e excluir objeto
- Pesquisar objeto: Permite ao usuário usar a busca textual para pesquisar um objeto de robótica do seu interesse.
- Mostrar detalhes: Ao clicar em mostrar detalhes, o usuário pode visualizar o objeto com todos os metadados inseridos durante o cadastro do objeto.
- Atualizar objeto: Permite ao usuário a edição de um objeto já existente postado. Ao clicar em “atualizar objeto”, o usuário é redirecionado para a página de cadastro de objeto, porém agora com os campos a serem editados.
- Deletar objeto: O usuário pode usar essa funcionalidade para excluir um objeto da listagem.
- Curtir ou descurtir objeto: O usuário pode deixar uma pequena avaliação em forma de curtir ou descurtir no objeto de seu interesse.
- Contate-nos: Nesta página, o usuário pode enviar um e-mail diretamente para os desenvolvedores com uma mensagem personalizada através de um formulário. O formulário contém um campo para o usuário colocar o seu nome, um para o e-mail e outro campo para a mensagem que ele escreverá para os desenvolvedores. Ao clicar em “enviar mensagem” é enviado um e-mail para o e-mail do GTEC, com o que foi digitado nos campos.
- Lista de objetos relacionados: O usuário pode visualizar os objetos relacionados com o objeto no qual se está vendo os detalhes.

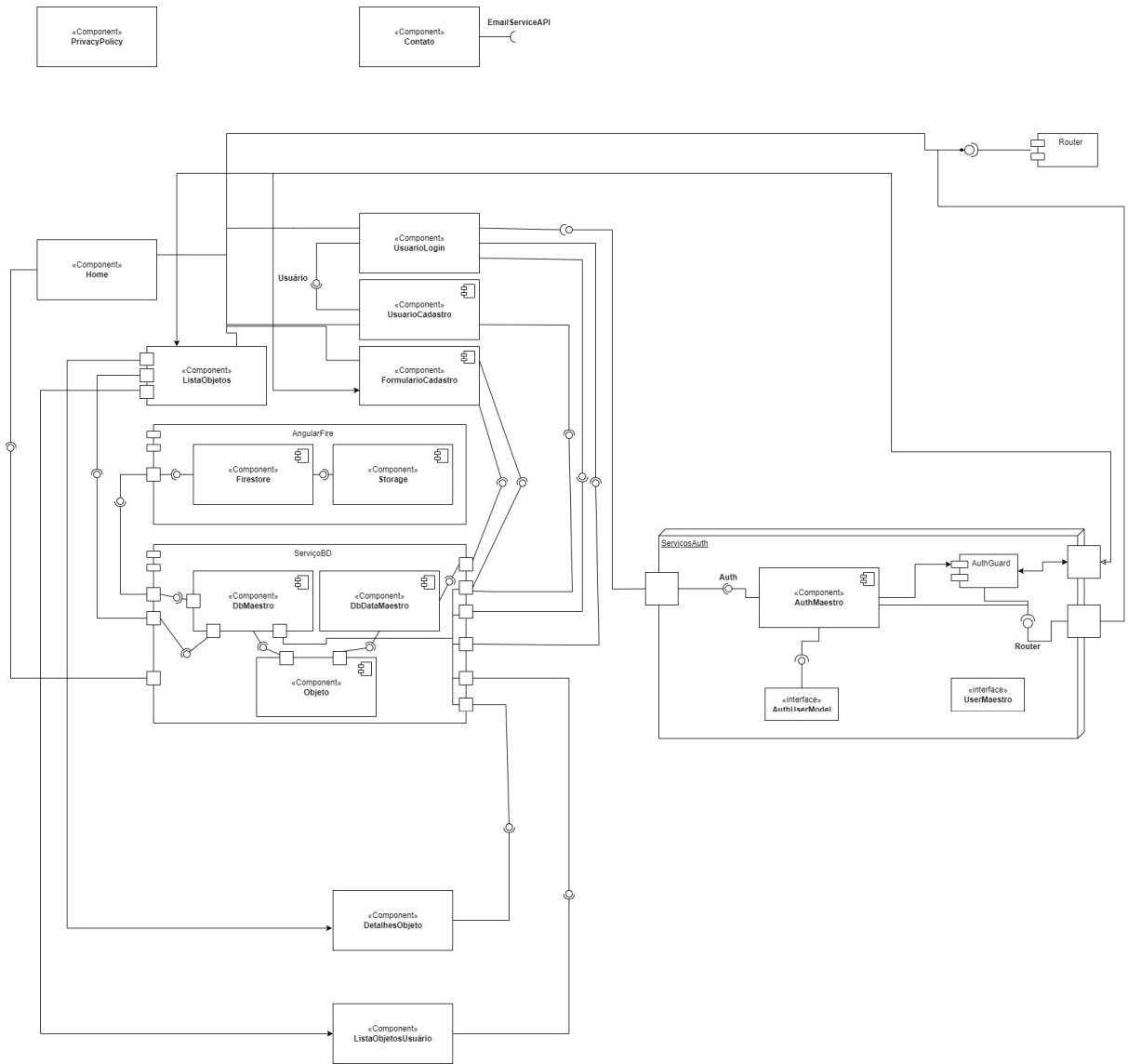
- Política de Privacidade: Onde o usuário pode ver o documento da política de privacidade do RepositORE.

2.1 DIAGRAMA DE COMPONENTES

A Figura 1 apresenta o diagrama de componentes, onde descreve como as partes diferentes interagem entre si.

Entre as partes mais importantes estão os núcleos de sistema ServiçosAuth e ServiçoBD (autenticação e banco de dados respectivamente), que são os blocos onde ocorrem as transações com os serviços do AngularFire uma implementação do Google Firebase. Também temos a interface do Router que provê navegação no plano de fundo. O restante dos blocos representa as outras páginas acessíveis do sistema, como por exemplo as páginas de exibição de objetos, a página principal e a página de contato, que tem uma comunicação com a API que é responsável pelo envio de e-mails através do SendGrid.

Figura 1 – Diagrama de Componentes.



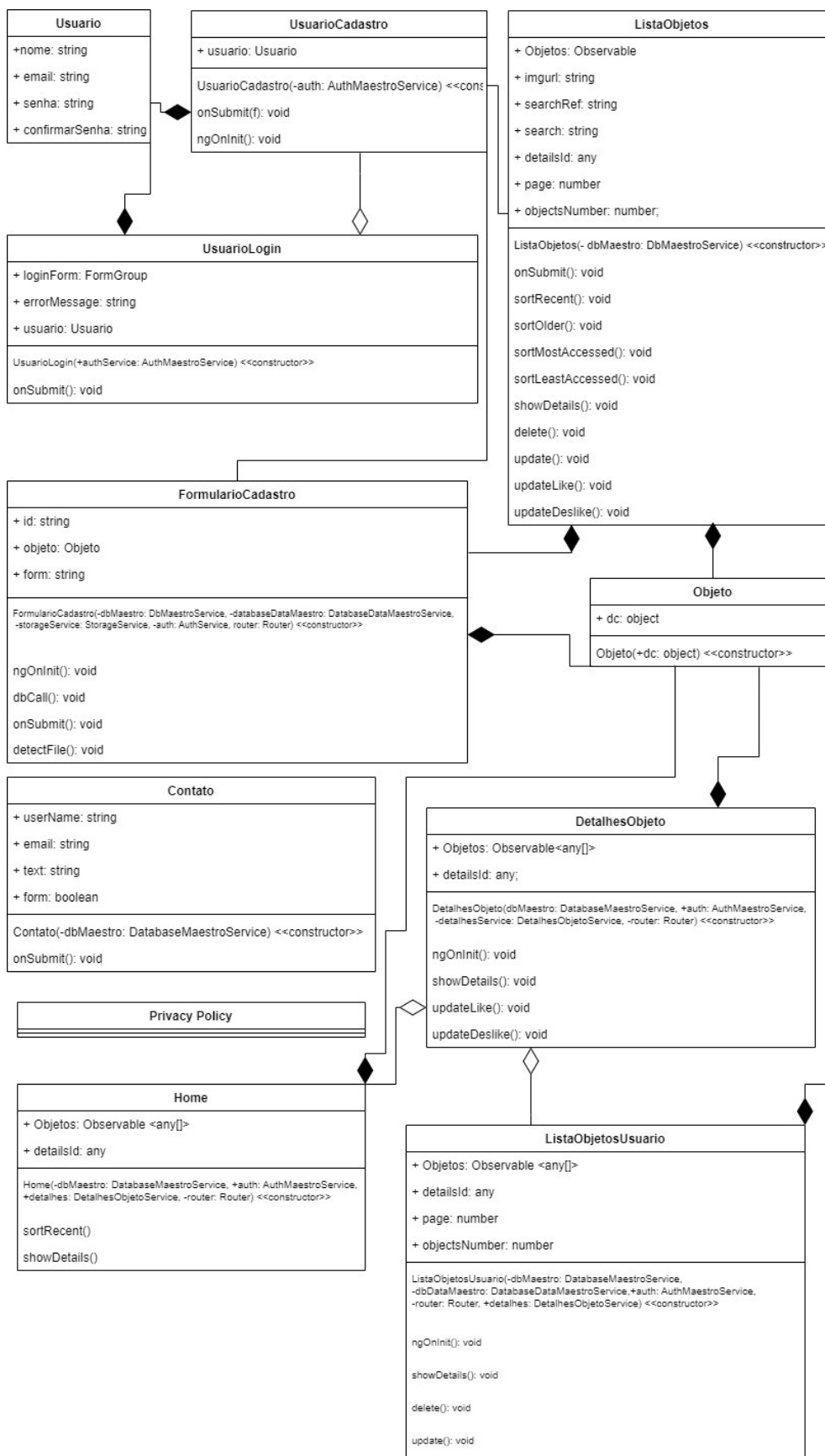
Fonte: Autoria Própria

2.2 DIAGRAMA DE CLASSES

A Figura 2 apresenta o diagrama de classes, que descreve as estruturas internas de cada uma das classes do sistema, e suas interações a nível local.

Dentre essas classes:

- Objeto é uma interface que define o que um objeto é para o sistema.
- ListaObjetos é a classe responsável por mostrar os objetos cadastrados no banco.
- FormularioCadastro classe responsável pela criação de um objeto e por coloca-lo no banco de dados.
- DetalhesObjeto classe que gera a pagina de detalhes de cada objeto.
- ListaObjetosUsuario classe semelhante a ListaObjetos, mas específica em exibir somente os objetos criados pelo próprio usuário.
- Home pagina inicial do sistema, que mostra os últimos objetos cadastrados no sistema
- Contato classe geradora da pagina de contato, que permite um portal de acesso fácil de comunicação com a gestão do sistema.
- UsuarioCadastro e UsuarioLogin paginas de cadastro de usuário por e-mail e pagina de login de cadastro por e-mail ou por conta do google.
- PrivacyPolicy uma classe que gera uma pagina simples com a política de privacidade.

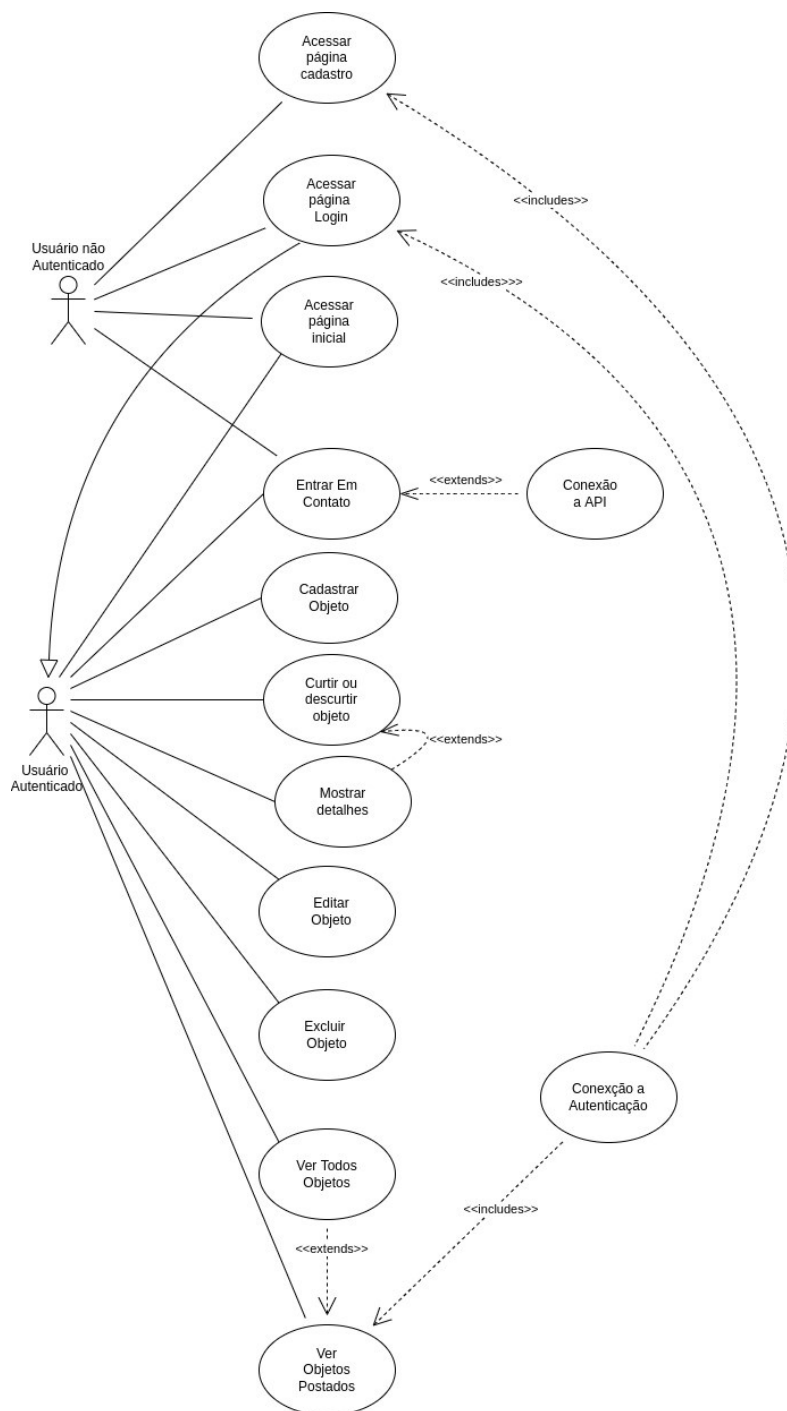


Fonte: Autoria Própria

2.3 DIAGRAMAS DE CASO DE USO

A figura 3 apresenta o diagrama de casos de uso onde mostra os acontecimentos experienciados por usuários que estão acessando a página. Isso inclui usuários não cadastrados que tem acesso bem limitado aos recursos do sistema e usuários cadastrados, que tem o acesso completo aos recursos do sistema.

Figura 3 – Diagrama de Caso de Uso.



Fonte: Autoria Própria

2.4 DESCRIÇÃO REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS

Os principais requisitos funcionais do sistema são:

- Cadastrar Objetos: É possível cadastrar um objeto, postando-o.
- Atualizar objetos: É possível atualizar um objeto já existente.
- Listar objetos: Os objetos podem ser visualizados através de uma listagem.
- Pesquisar objetos: Os objetos da lista podem ser filtrados por meio de busca textual.
- Excluir objetos: Os objetos da lista podem ser excluídos.
- Ver detalhes de objetos: Uma página que exibe todos os campos de um objeto é revelada
- Curtir objeto: A quantidade de curtidas em um objeto é atualizada.
- Descurtir o objeto: A quantidade de descurtidas em um objeto é atualizada.
- Enviar e-mail de contato para os desenvolvedores: É enviado um e-mail com as informações preenchidas pelo usuário.
- Fazer login com e-mail criado: O usuário pode se autenticar na plataforma com uma conta de e-mail criada.
- Fazer login com conta Google: O usuário pode se autenticar na plataforma com uma conta Google existente.
- Fazer logout: O usuário pode encerrar a sessão de sua conta.
- Criar conta: É possível criar uma conta de e-mail no sistema.
- Filtrar por objetos mais antigos primeiro: A listagem é ordenada, do objeto mais antigo ao mais recente.
- Filtrar por objetos mais recentes primeiro: A listagem é ordenada, do objeto mais recente ao mais antigo.
- Filtrar por objetos mais acessados primeiro: A listagem é ordenada, do objeto mais acessado ao menos acessado.
- Filtrar por objetos menos acessados primeiro: A listagem é ordenada, do objeto menos acessado ao mais acessado.

Os principais requisitos não funcionais são:

- **Confiabilidade:** O software deve apresentar consistência e estabilidade, estando sempre disponível para os usuários.
- **Eficiência:** O software deve tratar os dados com a maior eficiência e velocidade o possível.
- **Usabilidade:** O software deve ser intuitivo, isto é, fácil de utilizar, não deixando brechas para mal uso ou dificuldade de uso.
- **Desempenho:** O software deve ter o melhor desempenho possível, não possuindo problemas de carregamento e tratamento de dados, mesmo com um número crescente e grande de usuários e dados.
- **Escalabilidade:** O software deve ser escalável, isto é, deve permitir a implementação de novas funcionalidades com facilidade, sem que o código já existente necessite ser refeito.
- **Privacidade:** O software deve seguir um conjunto de regras de políticas de privacidade, nos quais são informados ao usuário quais dados são públicos ou não, e como esses dados são abordados e armazenados.
- **Segurança:** O software deve prover uma proteção sólida aos dados dos usuários, mantendo os mesmos seguros mesmo em caso de ataques.

3 IMPLEMENTAÇÃO

O software possui um total de oito diferentes páginas que são exibidas ao usuário, sendo que três dessas páginas são exibidas apenas para usuários autenticados. Visto que o software foi desenvolvido utilizando o *framework* Angular, que segue o modelo *single page application*, em todas as páginas são carregadas dentro de um único arquivo, sob a demanda em que o usuário requer acessar a página (ANGULAR, 2022).

Cada página é formada por um conjunto de arquivos, esse conjunto de arquivos é chamado de componente. A aplicação possui nove componentes.

O Frontend de todo o sistema consome as classes visuais do Bootstrap, isso inclui a página principal, em que o resto do sistema exibe as páginas a ser carregadas

e renderizadas pelo navegador. Além disso, também é utilizado os ícones disponibilizados também pelo Bootstrap.

- HomeComponent: Este componente gera a página principal da aplicação, sendo acessível por todos os usuários. Este componente gera uma página que mostra os últimos cinco objetos publicados no RepositORE.
- ListaObjetosComponent: Este componente gera a página de lista de objetos, sendo acessível somente pelos usuários autenticados, sendo alimentada pelo serviço DatabaseMaestroService, essa lista também conta com botões de ação todos os cartões ligados com cada um dos objetos. A paginação é provida por uma biblioteca ngx-pagination disponibilizada na rede npm. Essa página só mostra propriedades específicas de cada Objeto.
- DetalhesObjetoComponent: Este componente gera uma página com todos os campos de um objeto específico que foi postado, sendo acessado somente por usuários autenticados. Essa página lista todas as propriedades cadastradas de um objeto específico.
- PrivacyPolicyComponent: Este componente gera a página de políticas de privacidade da plataforma, sendo exibida para qualquer usuário. É uma página simples, que pode ser acessada em qualquer ponto do sistema, através de um link de rodapé no fim da página.
- FormularioCadastroComponent: Este componente gera a página de formulário de cadastro de objetos, sendo exibido somente para usuários autenticados, esse formulário contém diversas propriedades que podem ser adicionadas em um objeto, mas possui poucas propriedades obrigatórias. Também é nessa página que ocorre o uso do serviço StorageService, que será invocado quando o usuário anexar uma imagem ao Objeto.
- ContatoComponent: Esta página gera o formulário de contato com os desenvolvedores, sendo exibida para qualquer usuário, essa página utiliza uma API desenvolvida para o projeto, que contacta os serviços do SendgridJs.

- **UsuárioLoginComponent:** Este componente gera a página em que o usuário usa para iniciar sua sessão autenticada no sistema. Existe a opção de fazer login com um endereço de e-mail confirmado e senha ou então usando uma conta do Google. Caso o usuário esteja autenticado, será renderizado em vez o componente **ListaObjetosUsuarioComponent**, servindo então como página de perfil de usuário. Essa página consome o serviço **AuthMaestroService**.
- **UsuarioCadastroComponent:** Este componente gera a página que o usuário utiliza para criar uma conta utilizando um e-mail de sua escolha. Essa página consome o serviço **AuthMaestroService**.

No campo de serviços temos:

- **DatabaseMaestroService:** Este serviço é o responsável por fazer a conexão entre componentes e banco de dados.
- **StorageService:** É o serviço responsável por fazer o envio de arquivos de imagem/vídeo para o banco de dados no Firebase.
- **AuthMaestroService:** Este serviço é responsável pela autenticação de contas de usuário e controle de sessões de usuário.
- **Index.js:** Este serviço é responsável por fazer o envio de e-mails de contato com os desenvolvedores, utilizando do serviço de envio de e-mails proveniente do Sendgrid.

3.1 METODOLOGIA DE DESENVOLVIMENTO

Para o desenvolvimento do código, foi utilizada a metodologia ágil Scrum. Esta metodologia foi escolhida devido aos seus benefícios e fluxo de trabalho se adequarem bem ao estilo de desenvolvimento e relacionamento da equipe. Os principais benefícios que se mostraram de suma importância durante o desenvolvimento foram: agilidade, adaptabilidade e colaboração. O projeto foi dividido em *sprints*, que é dividir o processo de construção do software em pedaços, criando uma melhor visualização da evolução do projeto, e tornando o processo de desenvolvimento mais claro (SCRUM, 2022).

As funcionalidades a serem implementadas, divisão de tarefas e erros a serem resolvidos foram definidos através de *issues* no GitHub. Uma *issue* é uma

marcação de funcionalidade a ser implementada, ou seja, o desenvolvedor utiliza *issues* para monitorar uma funcionalidade que ele precisa implementar.

Foram realizadas reuniões semanais, geralmente contendo no mínimo 40 minutos de duração. Durante as reuniões, eram discutidas as funcionalidades implementadas, a forma na qual foram implementadas, melhorias que poderiam ser feitas nessas funcionalidades, quais funcionalidades seriam desenvolvidas e estipulação de prazos.

3.2 FERRAMENTAS E BIBLIOTECAS UTILIZADAS

Foram utilizadas diversas ferramentas e tecnologias, sendo as principais:

- Google Firebase: Um BaaS (Backend as a Service) disponibilizada pelo google que simplifica a implantação de qualquer serviço ou site na web (FIREBASE, 2022).
- Angular: Um framework mantido pela Microsoft, Google e outras empresas que agiliza a construção de qualquer webapp através da automatização e simplificação dos métodos e bibliotecas do JavaScript.
- Bootstrap: um framework de desenvolvimento front-end de código aberto para desenvolvimento de interfaces em aplicativos web, permitindo aos desenvolvedores a criação de interfaces responsivas e personalizadas mais facilmente (BOOTSTRAP, 2022).
- Github: ferramenta poderosíssima de controle de código fonte e de versão de aplicativo.
- Visual Studio Code: Editor de Código especializado em desenvolvimento cloud, mobile e web. “O Visual Studio Code é um editor de código-fonte leve, mas poderoso, que é executado em sua área de trabalho e está disponível para Windows, macOS e Linux. Ele possui JavaScript, TypeScript e Node, com suporte de C, integrado e outras linguagens (como vem C++) e outras linguagens (como vem com C++, Java, Python, PHP, Go e outras linguagens de execução)” (VISUAL STUDIO CODE, 2022).

- NodeJS: Um Runtime Javascript desenvolvido sobre o motor de Javascript V8 do Google.
- Twilio SendGrid: O SendGrid é uma plataforma que proporciona uma *api* para comunicação com usuários através de e-mails transacionais. O SendGrid utiliza um serviço de computação em nuvem.

3.3 CAPTURAS DE TELA

A Figura 4 apresenta a vista da tela de página inicial da plataforma, onde o usuário pode visualizar as cinco últimas postagens feitas por usuários. Esta página pode ser vista por usuários autenticados e não autenticados.


Figura 4 – Pagina Inicial.



Fonte: Autoria Própria

A Figura 5 apresenta a página de objetos postados pelo próprio usuário.


Figura 5 – Pagina do Usuário.

RepositORE  Perfil [Cadastrar Objetos](#) [Lista de Objetos](#) [Contate-nos](#)

Kefton David Nunes de Melo

Você está logado!
Seja bem vindo ao RepositOre!


[Sair](#)



Título: **Robô lego**

Subtítulo: Tutorial de montagem de um robô de lego

Descrição Geral: Montagem de robô lego, básica e



Título: **Legu Mindstorm**

Subtítulo: Robô EV3

Descrição Geral: Demonstração básica dos aspectos técnicos

Fonte: Autoria Própria

A Figura 6 apresenta o formulário de cadastro de objetos de aprendizagem. No cadastro, existem um conjunto de campos a serem preenchidos, sendo que apenas cinco campos são obrigatórios. Cada campo possui um botão de ajuda, que mostra um resumo sobre o campo específico.

Figura 6 – Pagina de Cadastro.

RepositORE Perfil Cadastrar Objetos Lista de Objetos Contate-nos

Cadastro de Objetos

campos com (*) são obrigatórios

*Titulo: Ajuda Faltam 50 caracteres! Subtítulo: Ajuda Faltam 50 caracteres!

Autor: Ajuda Faltam 50 caracteres! Aprendiz: Ajuda

Organizador: Ajuda Faltam 50 caracteres!

Assunto: Ajuda

*Descrição geral: Ajuda Faltam 250 caracteres!


Palavra-chave: Ajuda Faltam 50 caracteres! Construção: Ajuda Faltam 50 caracteres!

Tipo de programação: Ajuda Objetivo do programa: Ajuda Faltam 100 caracteres!

Fonte: Autoria Própria

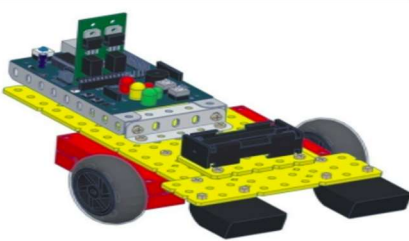
A Figura 7 apresenta a tela de listagem de objetos de robótica educacional, postados pelos usuários autenticados. Na lista de objetos, é possível ver algumas funcionalidades, tais como: a barra de pesquisa textual por objetos; filtragem por postagens mais recentes, mais antigas, mais acessadas e menos acessadas; botões de curtir e descurtir um objeto de aprendizagem; atualizar, deletar e ver detalhes dos objetos.

Figura 7 – Lista de Objetos.

RepositORE  Perfil [Cadastrar Objetos](#) [Lista de Objetos](#) [Contate-nos](#)

Pesquisa por Objeto

Mais recentes Mais antigos Mais acessados Menos acessados



Título: Carrinho Modelix


Subtítulo: Tutorial de montagem de um carrinho robô

Descrição Geral: Tutorial básico de montagem de carrinho robô, para iniciantes

Autor: Kefton David e Sebastião Alves

Palavra Chave: robótica

Formato:



Título: Robô lego

Subtítulo: Tutorial de montagem de um robô de lego

Descrição Geral: Montagem de robô lego, básica e simplificada para iniciantes.

Autor: Cleyton Carlos


Palavra Chave: robótica

Formato:

Fonte: Autoria Própria

A Figura 8 apresenta a lista de opções disponível para cada um dos objetos, incluindo opções de “Curtir” e “Descurtir”, mostrar mais detalhes, editar informações do Objeto e Excluir permanentemente o mesmo.

Figura 8 – Opções do Objetos.



Título: Carrinho Modelix

Subtítulo: Tutorial de montagem de um carrinho robô

Descrição Geral: Tutorial básico de montagem de carrinho robô, para iniciantes

Autor: Kefton David e Sebastião Alves

Palavra Chave: robótica


Formato:

Data de Disponibilização: 2-4-2022 17:15:50

Postado por **você**

Número de acessos: 5

Mostrar detalhes



Título: Robô lego

Subtítulo: Tutorial de montagem de um robô de lego

Descrição Geral: Montagem de robô lego, básica e simplificada para iniciantes.

Autor: Cleyton Carlos

Palavra Chave: robótica

Formato:

Data de Disponibilização: 27-3-2022 22:50:54

Postado por **você**

Número de acessos: 32

Mostrar detalhes

Fonte: Autoria Própria

A Figura 9 apresenta a página de contato com a equipe administrativa do RepositORE através de e-mail.

Figura 9– Pagina de Contato.



RepositORE Perfil Cadastrar Objetos Lista de Objetos Contate-nos

Envie uma mensagem para a nossa equipe!

Seu nome:

Seu e-mail:

Sua mensagem:

Restam 250 caracteres!

Fonte: Autoria Própria

A Figura 10 apresenta pagina para Login no sistema, essa página só está disponível enquanto não estiver autenticado.

Figura 10 – Página de Login.

The screenshot shows a web interface with a red navigation bar at the top containing the text: 'RepositORE', 'Login', 'Crie sua conta', 'Cadastrar Objetos', 'Lista de Objetos', and 'Contate-nos'. Below the navigation bar is a white box with the title 'Login com Email ou Conta Google'. Inside this box, there are two input fields: 'E-mail' with the placeholder 'nome@email.com' and 'Senha' with the placeholder 'senha'. Below the password field is a note: 'A senha deve ter ao menos 8 caracteres'. At the bottom of the form are three buttons: a grey button labeled 'Login com E-mail', a white button labeled 'Crie sua conta', and a blue button labeled 'Login com Google'.

Fonte: Autoria Própria

4 TESTES E VALIDAÇÃO

Foi realizada uma série de testes e etapas de teste de software, para garantir a usabilidade, e que a aplicação possua somente os comportamentos desejados.

Testes realizados:

- Testes unitários: Foram feitos diversos testes unitários durante o desenvolvimento do software, em todos os arquivos que foram utilizados para criação de código. Na data da criação desse documento, todos os componentes foram amplamente testados separadamente.
- Testes de integração: Todos os componentes da aplicação foram testados, e as interações entre os componentes que necessitam de interoperabilidade entre si se mostraram satisfatórias. Entretanto, como

é utilizado o formato de cobrança gratuito do google firebase, a plataforma tem um limite operacional associado com os limites impostos pelo Google Firebase, isso inclui o limite de trafego simultâneo, a capacidade máxima de armazenamento total do banco de dados, entre outros limites.

- Teste operacional: A funcionalidade da aplicação foi extensamente colocada a prova em seu estado hospedado em um provedor de serviços web, como a aplicação foi desenhada para ser utilizada, e sua funcionalidade de todos os casos de uso foi avaliada como funcional. Todos os módulos não apresentaram nenhum tipo de mal comportamento e as respostas da aplicação está como foi planejada. Não foi possível realizar um teste de estresse da plataforma pois para isso é necessário de mais de 20 pessoas a utilizando ao mesmo tempo.

5 PERSPECTIVAS FUTURAS

O estado atual do RepositORE deve servir como prenúncio para a implementação de diversas funcionalidades novas e melhorias de qualidade de vida e usabilidade do software. Algumas melhorias a serem implementadas são: melhorias no sistema de recomendação de conteúdos, implementação de página de administrador, melhorias no perfil do usuário, entre outros.

Deste momento em diante, em que a aplicação já está estruturada, deve ser feita a manutenção, monitoramento, implementação de novas funcionalidades e teste das funcionalidades novas, levando em consideração a escalabilidade de código, para que o aperfeiçoamento da plataforma seja crescente e coeso.

6 REFERENCIAS BIBLIOGRÁFICAS

GURGEL, Thalia K. S. **RepositORE: Um Repositório de Objetos de Aprendizagem para Robótica Educacional**, Trabalho de Conclusão de Curso de Ciências da Computação, Faculdade de Ciências Naturais e Exatas, Universidade Estadual do Rio Grande do Norte, Mossoró, 2019.

ROSETTO, Marcia; NOGUEIRA, Adriana Hipólito. **Aplicação de elementos metadados DUBLIN CORE para descrição de dados bibliográficos on-line da biblioteca digital de teses da USP**. Anais.. Recife: SNBU, 2002.

GONÇALVES, Paulo de Castro. et al. **Adequação do Dublin Core ao AACR2: o caso da Biblioteca Digital da Assembleia Legislativa do Estado de Minas Gerais**. XXV.

DANIELA, Pires. **USO DO DUBLIN CORE NA DESCRIÇÃO DE OBRAS RARAS NA WEB: A COLEÇÃO DA BIBLIOTECA BRASILIANA DIGITAL**, 2012.

ANGULAR. **Angular**. 2022. Disponível em: <https://angular.io>. Acesso em: 8 abr 2022.

SCRUM. **Scrum**. 2022. Disponível em: <https://www.scrum.org/resources/what-is-scrum>. Acesso em: 8 abr 2022.

FIREBASE. **Firestore**. 2022. Disponível em: <https://firebase.google.com/docs>. Acesso em: 9 abr 2022.

BOOTSTRAP. **Bootstrap**. 2022. Disponível em: <https://getbootstrap.com>. Acesso em: 9 abr 2022.

VISUAL STUDIO CODE. **Visual Studio Code**. 2022. Disponível em: <https://code.visualstudio.com/docs>. Acesso em: 9 abr 2022.