



Documento de Projeto de Sistema

ClassiWeb

Registro de Alterações:

Versão	Responsável	Data	Alterações
1.0	Ezequiel, Matheus, Thiago e Vinícius	24/02/2021	Levantamento dos requisitos e introdução
1.1	Thiago	25/02/2021	Inserção do Mini-mundo
1.2	Thiago e Vinícius	08/03/2021	Alterações no Mini-mundo
1.3	Matheus e Vinícius	11/03/2021	Descrição das plataformas, frameworks e bibliotecas utilizadas para o desenvolvimento.
1.4	Ezequiel, Matheus, Thiago e Vinícius	12/03/2021	Inserção dos modelos e finalização do documento

Vitória, ES

2021

1 Introdução

Este documento apresenta o projeto (*design*) do sistema *ClassiWeb*. O sistema consiste em uma aplicação web com o objetivo de promover o encontro entre seus usuários para venda de produtos. A solução proposta permitirá aos usuários divulgarem produtos para venda e gerenciar o estado do anúncio. O anúncio do produto deverá incluir um nome/título, descrição, quantidade, preço, estado (novo ou usado), fotos, localidade e categoria. Os estados para os anúncios são “oculto” ou “publicado” (visível a todos no sistema). Um usuário é considerado vendedor de um anúncio se ele for o autor do mesmo. Ademais, é denominado cliente de um anúncio o usuário que efetivou sua compra. O vendedor poderá a qualquer momento atualizar o estado do anúncio. Uma vez criado o anúncio estará no estado oculto, ou seja, ele não será divulgado na plataforma. Assim, o anúncio somente será disponibilizado e divulgado na plataforma quando o vendedor alterar seu estado para publicado. Quando um cliente realizar uma compra, o sistema irá sugerir tanto ao anunciante quanto para o cliente que avaliem como foi a experiência da venda. O usuário poderá atribuir uma nota (Muito insatisfeito, Insatisfeito, Indiferente, Satisfeito e Muito Satisfeito). Note que o sistema não deve permitir a venda de um produto para seu anunciante. A média das notas das avaliações será divulgada no perfil público dos usuários. O vendedor também poderá cancelar o anúncio, o que implicará em apagar todos os seus dados relacionados.

Além disso, as possíveis categorias dos anúncios serão pré-cadastradas pelo administrador do sistema.

Quanto aos usuários, o sistema deverá suportar dois perfis de usuários distintos: usuário comum e administrador. Os visitantes são os usuários não autenticados. Esses usuários somente poderão pesquisar por anúncios publicados. Já o perfil de usuários comuns são aqueles cadastrados e autenticados no sistema. Nesse perfil, são permitidas todas as funcionalidades do visitante, bem como, a criação e gestão dos anúncios. O perfil de administrador permitirá excluir e/ou banir anúncios e usuários que infringjam qualquer uma das políticas vigentes da plataforma e cadastrar as possíveis categorias de anúncio. São requeridos para o perfil de administrador nome, registro na empresa (matrícula), email e senha. Já para o usuário comum são necessários o nome, CPF, endereço, email e senha (email e senha serão utilizados como login).

A busca por anúncios engloba apenas os anúncios publicados que satisfaçam os filtros preenchidos de descrição, categoria, preço (mínimo e máximo) e localidade. Ao usuário comum será permitido criar uma lista de produtos desejados, na qual serão mantidos os produtos de interesse do mesmo. Ao incluir um produto nessa lista, o vendedor receberá uma notificação do usuário interessado naquele produto. Além disso, o usuário receberá notificações das atualizações dos produtos incluídos nessa lista. O sistema também

permitirá uma troca de mensagens básica entre um usuário comum e o vendedor de um produto.

Além desta introdução, este documento está organizado da seguinte forma: a Seção 2 apresenta a plataforma de software utilizada na implementação do sistema; a Seção 3 apresenta a arquitetura de software; por fim, a Seção 4 apresenta os modelos FrameWeb que descrevem os componentes da arquitetura.

2 Plataforma de Desenvolvimento

A plataforma e os frameworks escolhidos para o desenvolvimento do sistema são apresentados na Tabela 1.

Tabela 1 – Plataforma de Desenvolvimento e Tecnologias Utilizadas.

Tecnologia	Versão	Descrição	Propósito
Node.js	14.16.0	Node.js é um software multiplataforma, que executa códigos JavaScript no backend/servidor e frontend/interface.	Redução da complexidade do desenvolvimento, implantação e gerenciamento de aplicações Web a partir de seus componentes e bibliotecas.
JavaScript	ES2020	Linguagem de programação multiparadigma com foco em desenvolvimento web.	Escrita do código-fonte dos componentes que compõem o sistema.
ReactJS	17.0.1	Biblioteca JavaScript para a criação de interfaces de usuários em aplicações Web.	Criação das páginas Web e comunicação com o back-end. Reutilização da estrutura visual comum às páginas, facilitando a manutenção do padrão visual do sistema.
TypeORM	0.2.31	API para persistência de dados por meio de mapeamento objeto/-relacional.	Persistência dos objetos de domínio sem necessidade de escrita dos comandos SQL.
TypeDI	0.10.0	API para injeção de dependências.	Integração das diferentes camadas da arquitetura.
JWT	8.5.1	Implementação do JsonWebToken para NodeJS.	Geração e validação de tokens para autenticação e autorização dos usuários do sistema.
MaterialUI	4.11.13	Conjunto de componentes visuais ReactJS <i>open source</i> .	Reutilização de componentes visuais Web de alto nível.
SQLite	3.34	Sistema Gerenciador de Banco de Dados Relacional gratuito.	Armazenamento dos dados manipulados pela ferramenta.
Express	4.17.1	Framework para construção de APIs RESTful	Utilizado para otimizar a construção de aplicações web e API's.

Já a Tabela 2 detalha as ferramentas de suporte ao desenvolvimento do projeto.

Tabela 2 – Softwares de Apoio ao Desenvolvimento do Projeto

Tecnologia	Versão	Descrição	Propósito
FrameWeb Editor	1.0	Ferramenta CASE do método FrameWeb.	Criação dos modelos de Entidades, Aplicação, Persistência e Navegação.
Overleaf	2021	Implementação e editor do \LaTeX	Documentação do projeto arquitetural do sistema, sendo usado o <i>template bsc-engsoft</i> . ¹
Visual Studio Code	1.54.1	Ambiente de desenvolvimento (IDE) com suporte ao desenvolvimento Typescript.	Implementação, implantação e testes da aplicação Web ReactJS.
Insomnia	2021.1	Ferramenta que auxilia nos testes de requisições a APIs.	Testar as APIs criadas para a aplicação.
npm	6.5	Ferramenta de gerência/construção de projetos de software.	Obtenção e integração das dependências do projeto.
Yarn	1.22	Ferramenta de gerência/construção de projetos de software.	Obtenção e integração das dependências do projeto.

3 Arquitetura de Software

O *ClassiWeb* consiste em um Sistema de Informação Baseado na Web (*Web-based Information Systems* - WIS), ou seja, é uma aplicação orientada a dados que será disponibilizada na Internet cujas informações e funcionalidades norteiam o seu desenvolvimento (SOUZA, 2020). Nesse contexto, o projeto do sistema foi elaborado embasado no FrameWeb (SOUZA, 2020) e segue o padrão arquitetônico proposto nessa metodologia. Assim, a arquitetura proposta é fundamentada no padrão de Camada de Serviço (FOWLER, 2002) que divide a aplicação em três camadas: a camada de apresentação (*Presentation Tier*), lógica de negócio (*Business Tier*) e acesso a dados (*Data Access Tier*). A camada de apresentação por sua vez é estruturada seguindo o padrão Modelo-Visão-Controlador (*Model-View-Controller*) (FOWLER, 2002). A Figura 1 apresenta a arquitetura do *ClassiWeb* explicitando os *frameworks* que serão utilizados no desenvolvimento de cada camada.

¹ [Repositório do Template - Bitbucket](#)

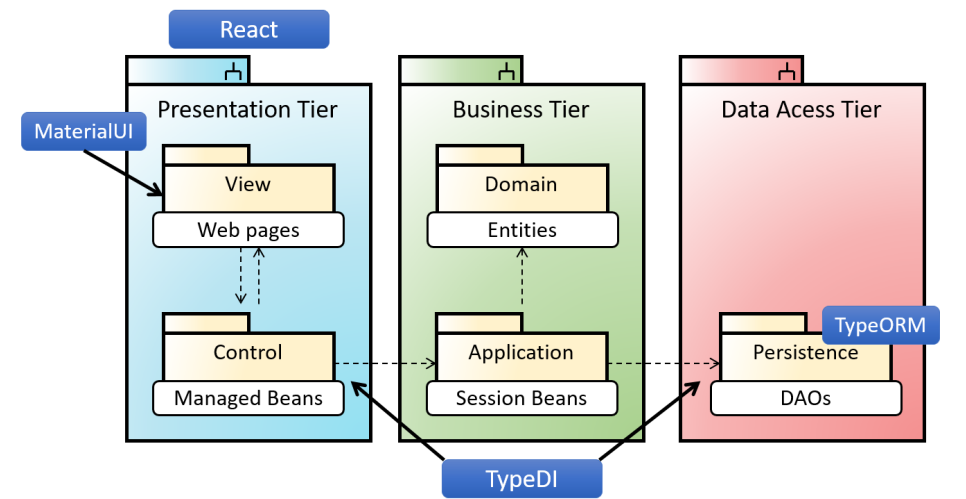


Figura 1 – Arquitetura do Sistema *ClassiWeb*

4 Modelagem FrameWeb

A metodologia FrameWeb propõe a estruturação da aplicação em quatro componentes principais distribuídos em três camadas distintas.

A camada de negócio possui o componente de entidades/pacote do domínio e o de aplicação, apresentados na seção 4.1. O primeiro refere-se à modelagem conceitual do retrato da realidade do domínio do problema no qual o sistema se propõe a resolver. Já o segundo, estrutura o sistema em serviços que implementam as regras de negócio manipulando os objetos do domínio do problema e manipulando também a persistência dos mesmos.

O componente de aplicação faz a interface com a camada de persistência que engloba o componente responsável por persistir os objetos fazendo o mapeamento objeto-relacional, apresentado na seção 4.2.

Por último, a camada de apresentação, detalhada na seção 4.3), inclui o componente de Navegação que é composto pela Visão e o Controlador, conforme proposto pelo padrão MVC. O pacote de Visão inclui os artefatos relacionados a interface com o usuário, como as páginas web. Já o Controlador, possui as classes de controle que gerenciam o fluxo de navegação entre as páginas da Visão, bem como, realiza a chamada às funcionalidades do sistema através da interface com o componente de aplicação.

¹ <<https://github.com/nemo-ufes/FrameWeb/issues/31>>

² <<https://github.com/nemo-ufes/FrameWeb/issues/32>>

³ <<https://github.com/nemo-ufes/FrameWeb/issues/24>>

Ressalta-se que houveram algumas limitações na elaboração dos diagramas dos componentes devido às questões identificadas na ferramenta de modelagem do FrameWeb conforme reportado no github [1](#) [2](#) [3](#).

Além disso, o escopo para a modelagem dos componentes considerou os três principais casos de uso do sistema: cadastrar e filtrar anúncio; e finalizar a compra de um anúncio.

O Caso de Uso para cadastrar anúncio deverá permitir que um usuário cadastrado e autenticado na plataforma inclua um novo anúncio informando os dados requeridos.

Por outro lado, o Caso de Uso para filtrar anúncios permitirá o usuário pesquisar por anúncios de interesse parametrizando a busca. Os dados necessários para o cadastro do anúncio e os possíveis parâmetros para o filtro são explicitados na Seção [1](#).

O último Caso de Uso retratado permitirá ao Cliente finalizar a compra de um anúncio e também a avaliação da compra, tanto pelo Cliente quanto pelo vendedor.

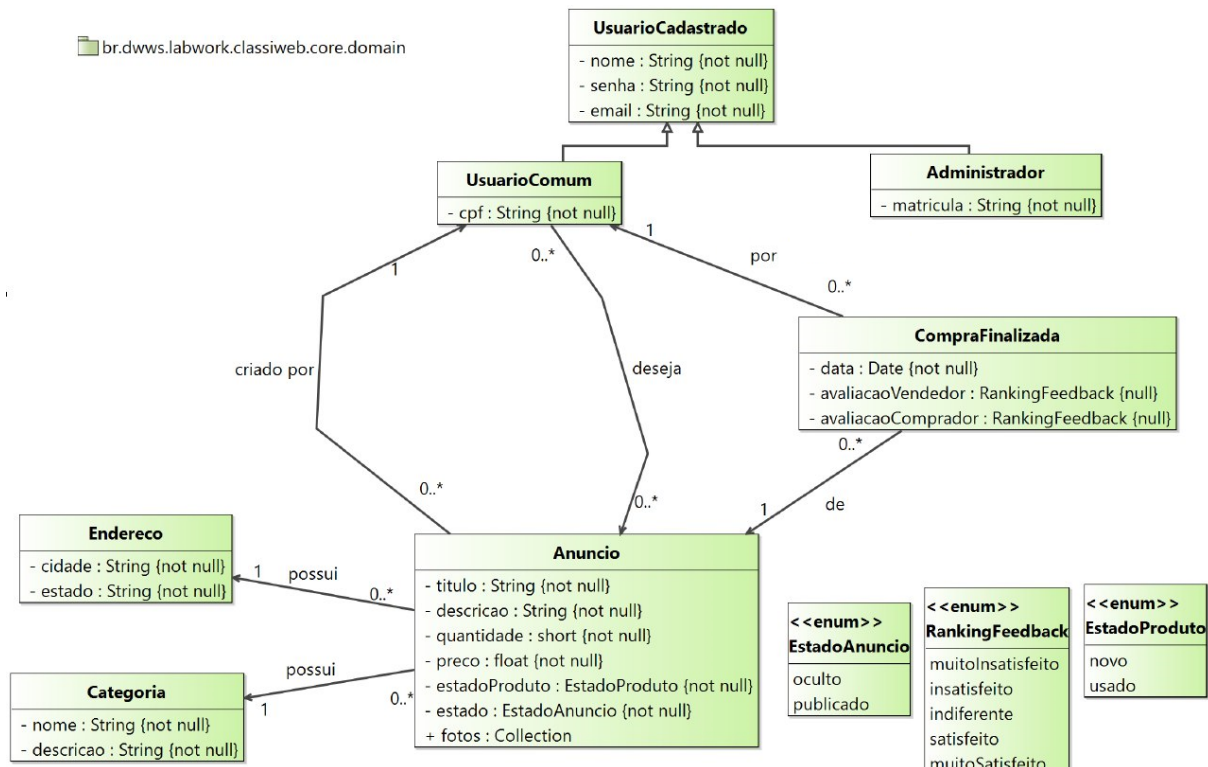
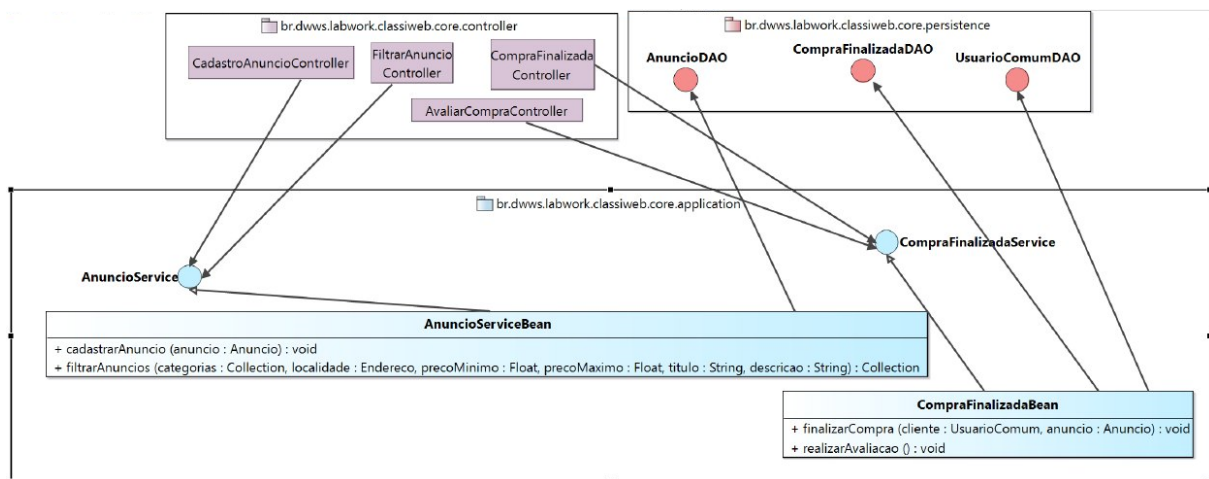
A Tabela [3](#) apresenta os *frameworks* que serão utilizados no desenvolvimento do sistema e sua correspondência análoga aos utilizados pelo Frameweb.

Tabela 3 – *Frameworks* da arquitetura do sistema separados por categoria.

Categoria de <i>Framework</i>	<i>Framework</i> Utilizado
Controlador Frontal	React
Injeção de Dependências	TypeDI
Mapeamento Objeto/Relacional	TypeORM
Segurança	JWT

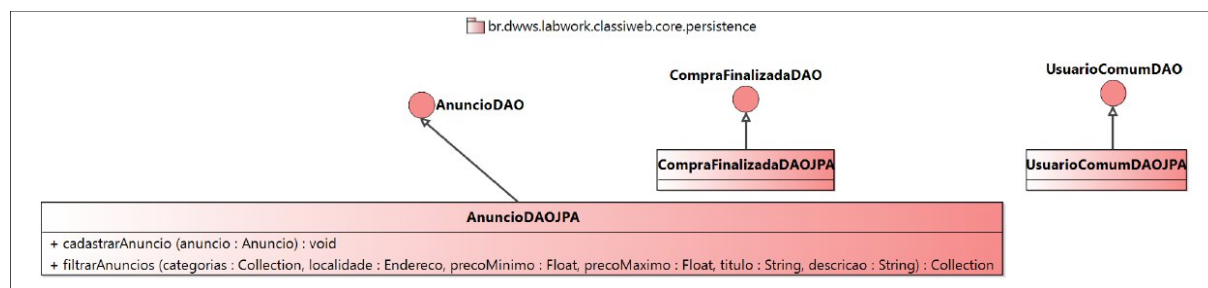
4.1 Camada de Negócio

A Figura [2](#) apresenta o componente de entidades da camada de negócio visando retratar o contexto descrito na Seção [1](#) - pacote *br.dwvs.labwork.classiweb.core.domain*. Ademais, o diagrama do componente de Aplicação referente às funcionalidades descritas na Seção anterior são apresentadas na Figura [3](#) - pacote *br.dwvs.labwork.classiweb.core.application*. Note que são apresentados as interfaces dos serviços e suas respectivas implementações somente das funcionalidades do escopo deste documento. Também são explicitadas as dependências desses serviços aos Controladores (na camada de Apresentação) e a camada de persistência.

Figura 2 – Modelo de Entidades do *ClassiWeb*.Figura 3 – Modelo de Aplicação do *ClassiWeb*.

4.2 Camada de Acesso a Dados

Optou-se por utilizar o padrão *Data Access Object* (DAO) (ALUR; CRUPI; MALKS, 2003) no componente de persistência visando tornar transparente a persistência dos objetos com relação ao *framework* de mapeamento objeto relacional escolhido. A Figura 4 descreve o modelo desse componente.

Figura 4 – Modelo de Persistência do *ClassiWeb*.

4.3 Camada de Apresentação

Os componentes de Visão e seus Controladores para os Casos de Uso cadastrar, filtrar anúncio e concluir compra são detalhados nas Figuras 5, 6 e 7, respectivamente. O pacote *br.dwws.labwork.classiweb.core.controller* inclui os Controladores e o pacote */core/-NomeCasoDeUso* as páginas web do Sistema referente ao Caso de Uso "NomeCasoDeUso".

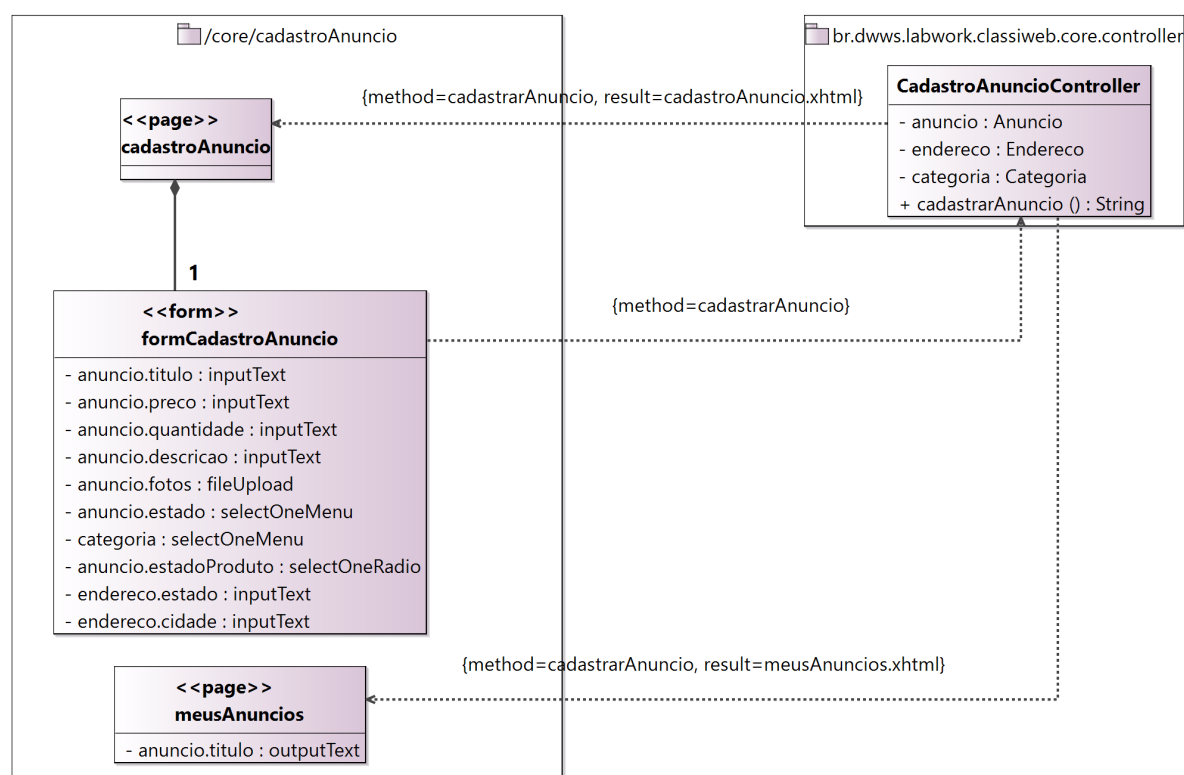


Figura 5 – Modelo de navegação da funcionalidade "Cadastrar Anúncio".

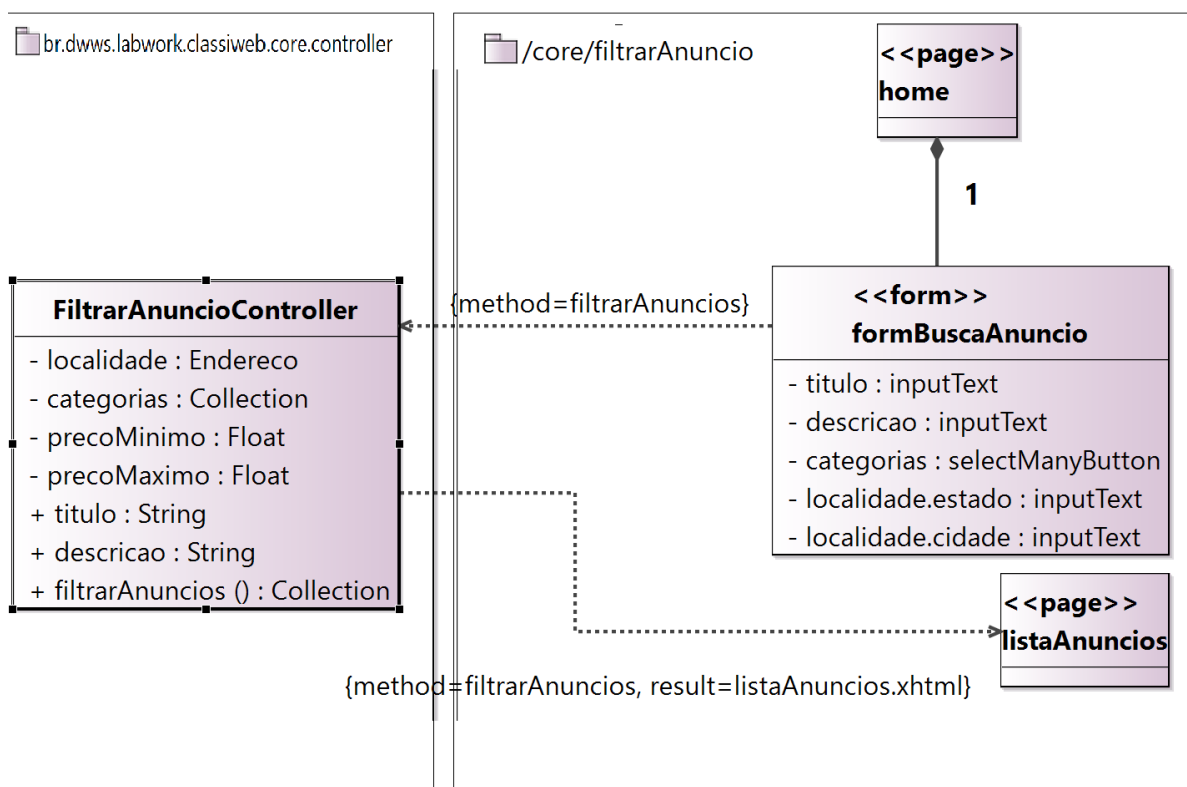


Figura 6 – Modelo de navegação da funcionalidade "Filtrar Anúncio".

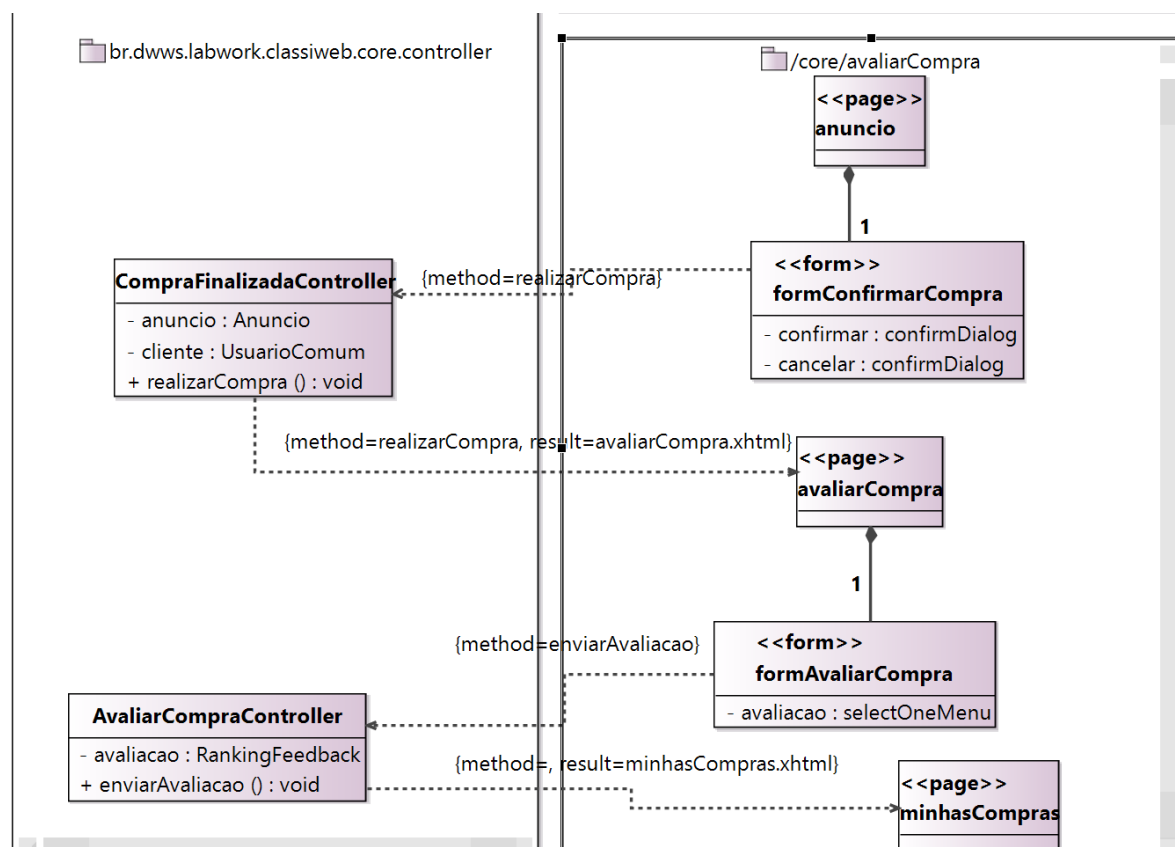


Figura 7 – Modelo de navegação da funcionalidade "Avaliar Compra".

Referências

ALUR, D.; CRUPI, J.; MALKS, D. *Core J2EE patterns: best practices and design strategies*. [S.l.]: Prentice Hall Professional, 2003. Citado na página 6.

FOWLER, M. *Patterns of Enterprise Application Architecture*. 1. ed. [S.l.]: Addison-Wesley, 2002. ISBN 9780321127426. Citado na página 3.

SOUZA, V. E. S. The FrameWeb Approach to Web Engineering: Past, Present and Future. In: ALMEIDA, J. P. A.; GUIZZARDI, G. (Ed.). *Engineering Ontologies and Ontologies for Engineering*. 1. ed. Vitória, ES, Brazil: NEMO, 2020. cap. 8, p. 100–124. ISBN 9781393963035. Disponível em: <<http://purl.org/nemo/celebratingfalbo>>. Citado na página 3.