



Documento de Projeto de Sistema

API CNPq

Vitória, ES

2023

Registro de Alterações:

Versão	Responsável	Data	Alterações
1.0	Júlia de Souza Borges	29/04/2023	Versão inicial.
1.1	Vilker Zucolotto Pessin	13/06/2023	Ajustes no modelo inicial.

1 Introdução

Este documento apresenta o projeto *API CNPq*, que é um produto desenvolvido na disciplina Desenvolvimento Web e Web semântica, ministrada pelo professor Dr. Vítor Estevão Silva Souza, na Universidade Federal do Espírito Santo (UFES).

O CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) disponibiliza informações relevantes sobre suas bolsas de estudo e pesquisa. ([CNPq, 2023](#))

Para o desenvolvimento da solução API CNPq utilizaremos algumas ferramentas populares, como PHP ([The PHP Group, 2023](#)), Composer ([The PHP Group, 2023](#)), Laravel ([Laravel LLC, 2023](#)), NodeJS ([OpenJS Foundation, 2023](#)) e Eloquent ([Marijn Haverbeke, 2023](#)). O objetivo será construir um sistema básico com desenvolvimento web com recursos CRUD (Criar, ler, atualizar e deletar) e outras funções demonstradas na disciplina. A solução foi criada para aplicar conhecimentos relacionados ao Desenvolvimento Web e Web Semântica em um projeto de código aberto, permitindo que todos os códigos estejam disponíveis para consulta e contribuição da comunidade de desenvolvedores.

Também utilizaremos o FrameWeb, solução que auxilia os desenvolvedores na construção de Sistemas de Informação na Web (WISs) por meio de métodos, linguagens e ferramentas que aproveitam a base arquitetônica oferecida por frameworks, além de facilitar a publicação de dados vinculados desses WISs ([SOUZA, 2020](#))

É importante ressaltar que os dados disponibilizados pelo CNPq através de sua API estão disponíveis em dados abertos do governo. Isso significa que essas informações são de livre acesso e podem ser utilizadas por qualquer pessoa, sem restrições de uso. Além disso, os dados abertos são um importante instrumento de transparência e controle social, pois permitem que a sociedade tenha acesso às informações públicas e possa fiscalizar as ações governamentais.

Dessa forma, a API CNPq pode ser um recurso valioso para o desenvolvimento de soluções que utilizem esses dados abertos para gerar informação sobre os projetos de pesquisa, desenvolvimento de aplicativos e outras atividades que possam contribuir para o avanço da ciência e tecnologia no Brasil.

O objetivo principal do API CNPq será de processar informações vindas de um arquivo csv e popular um banco de dados que possa ser facilmente acessado e manipulados, em desenvolvimento web, com o objetivo de aplicar o aprendizado da disciplina Desenvolvimento Web e Web Semântica. Com essa solução, será possível apresentar informações detalhadas e precisas sobre diferentes áreas de pesquisa, fornecendo aos usuários uma experiência de navegação que apresente dados.

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

Na Tabela 1 são listadas as tecnologias utilizadas no desenvolvimento da ferramenta, bem como o propósito de sua utilização.

Tabela 1 – Plataforma de Desenvolvimento e Tecnologias Utilizadas.

Tecnologia	Versão	Descrição	Propósito
Laravel	10.8	Framework web PHP gratuito e de código aberto, destinado ao desenvolvimento de aplicações web seguindo o padrão arquitetônico model-view-controller (MVC).	Redução da complexidade do desenvolvimento, implantação e gerenciamento de aplicações Web a partir de seus componentes de infra-estrutura prontos para o uso.
PHP	8.2.4	Linguagem de programação orientada a objetos e independente de plataforma.	Escrita do código-fonte das classes que compõem o sistema.
Blade	2.2.12	Manipulação do layout.	Principal ferramenta do Framework destinada a manipular dados para o front-end com um layout dinâmico e flexível.
Bootstrap	5.2.3	Conjunto de componentes visuais <i>open source</i> .	Reutilização de componentes visuais Web de alto nível.
Laravel MIX	6.0.49	Webpack.	Pré-processamento scss, sass, less, minificação de css, js e otimização de imagens.
Eloquent	10.8	API para persistência de dados por meio de mapeamento objeto-relacional.	Persistência dos objetos de domínio sem necessidade de escrita dos comandos SQL.
Composer	2.5.5	API para injeção de dependências.	Integração das diferentes camadas da arquitetura.
PostgreSQL	14	Sistema Gerenciador de Banco de Dados Relacional gratuito.	Armazenamento dos dados manipulados pela ferramenta.
Built-in web server	8.2.4	Servidor embutido de aplicações para PHP.	Fornecimento de implementação das APIs citadas acima e hospedagem da aplicação Web, dando acesso aos usuários via HTTP.

Na Tabela 2 vemos os softwares que apoiaram o desenvolvimento de documentos e também do código fonte.

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	v1	Implementação e editor de LaTeX.	Escrita da documentação do sistema, sendo usado o <i>template abnTeX</i> . ¹
Visual Studio Code	1.77.3	Ambiente de desenvolvimento (IDE) com suporte ao desenvolvimento PHP.	Implementação, implantação e testes da aplicação Web PHP.
Composer	2.5.5	Ferramenta de gerência/construção de projetos de software.	Obtenção e integração das dependências do projeto.

3 Arquitetura de Software

A Figura 6 mostra a arquitetura do sistema *API CNPq*. Como é possível observar utilizamos uma arquitetura Model View Controller (MVC) que foi criada em 1978 por Reenskaug. Esse modelo se baseia de 3 componentes principais (REENSKAUG, 1979).

O componente View é responsável por apresentar a aplicação, cuidando do que será mostrado ao cliente, se importando com como os dados serão representados, sem se importar de que maneira essas informações foram obtidas.

Por sua vez, o Model se responsabiliza por gerenciar o acesso ao banco de dados, manipulando, armazenando e atualizando os dados mantidos nele. É nesse modelo que ocorre a lógica de negócios da aplicação, processando e validando elas.

Por fim, o Controller fica responsável pela comunicação entre o View e o Model, assim, movendo toda a aplicação. Ele recebe as entradas do usuário, requisita os dados ao Model e atualiza o View para refletir às mudanças,

Desse modo, o fluxo de dados funciona da seguinte maneira. O cliente faz uma requisição por meio da interface do modelo View, essa entrada é então gerenciada pelo Controller que socilita ao Model os dados necessários do banco de dados, então ao receber essas informações, o Controller atualiza o View que responde modificando a interface do cliente.

¹ <<http://www.abntex.net.br>>.

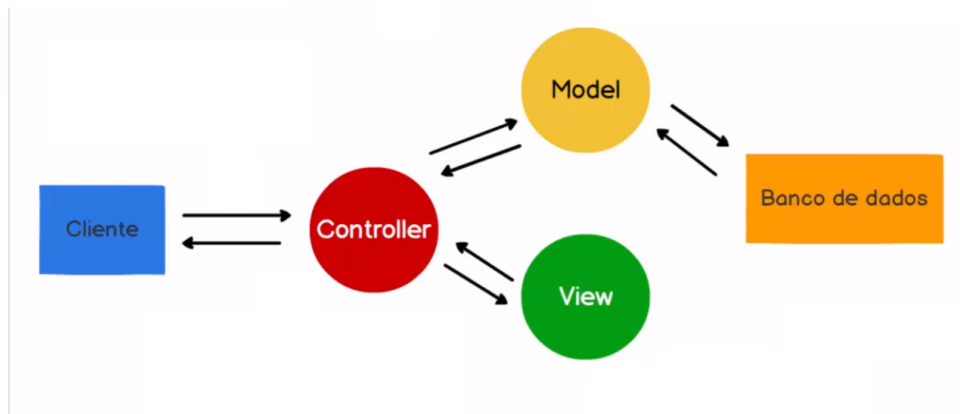


Figura 1 – Arquitetura de Software.

4 Modelagem FrameWeb

API CNPq é um sistema Web cuja arquitetura utiliza *frameworks* comuns no desenvolvimento para esta plataforma. Desta forma, o sistema pode ser modelado utilizando a abordagem FrameWeb (SOUZA, 2020).

A Tabela 3 indica os *frameworks* presentes na arquitetura do sistema que se encaixam em cada uma das categorias de *frameworks* que FrameWeb dá suporte. Em seguida, os modelos FrameWeb são apresentados para cada camada da arquitetura.

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

Categoria de <i>Framework</i>	<i>Framework</i> Utilizado
Controlador Frontal	Blade
Injeção de Dependências	Composer
Mapeamento Objeto/Relacional	Eloquent

4.1 Camada de Negócio

Nesta seção vamos descrever a camada de negócios do API CNPq apresentando as camadas que compõem a arquitetura deste sistema, mostrando as principais regras e a lógica de negócios da aplicação. Vamos dar enfoque aos processos que incluem a manipulação de dados, validação de entrada de dados, processamento de informações e outras tarefas relacionadas às operações, sobretudo apresentando os modelos desenvolvidos no frameweb e que lançaram os alicerces para o desenvolvimento da aplicação das regras que governam o comportamento do sistema.

Utilizando o frameweb foram desenvolvidos os modelos de aplicação e entidade relacionados à camada de Negócios.

O diagrama de entidade contém a UML de base de projetos, determinando a relação de todas as classes e atributos relacionados, bem como a relação de cardinalidade desses elementos. Este modelo também já foi concebido para integrar a solução de banco de dados, como por exemplo, as relações pelo identificador das chaves primárias e externas, permitindo gerar um superdomínio de informações.

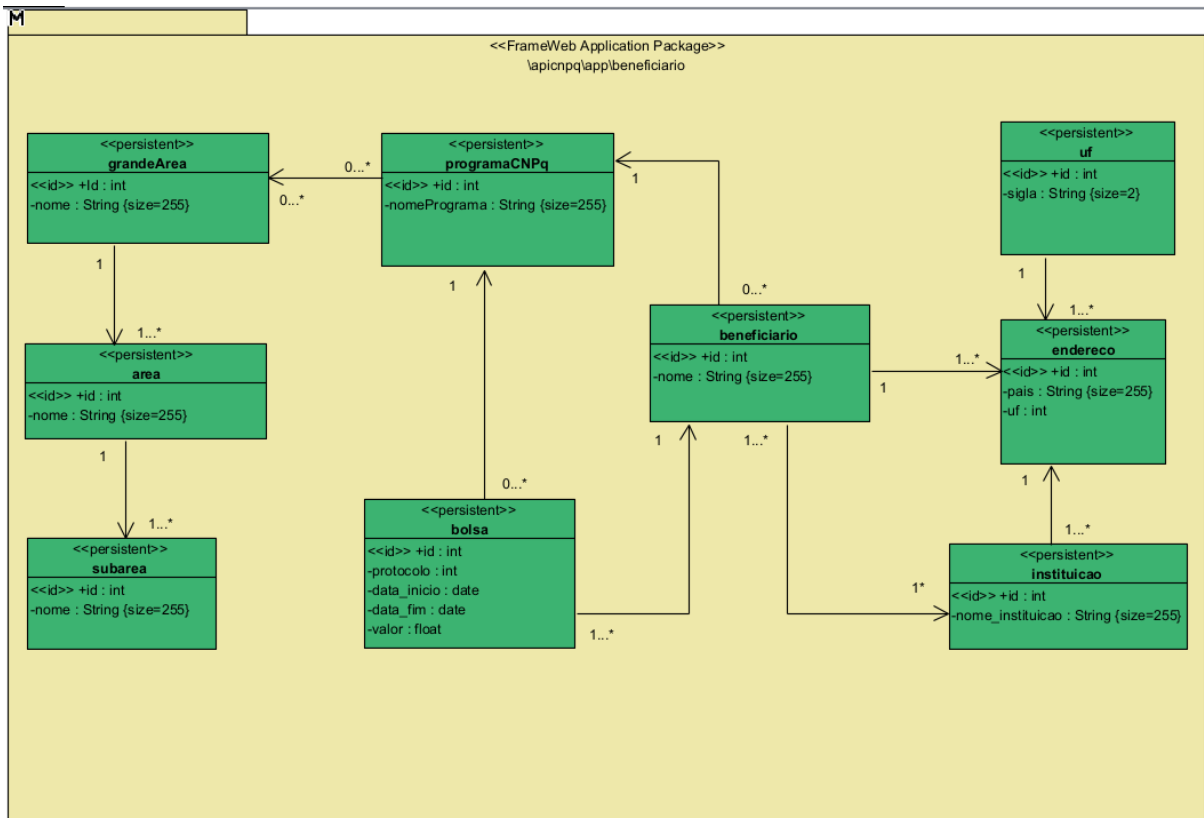


Figura 2 – Modelo de Entidade

Assim, na Camada de Negócio, o Beneficiário figura como a classe central do sistema, sendo desenvolvida inclusive uma opção de (C) 'Cadastro' de 'beneficiários' no banco de dados. Esta tabela se relaciona todas as outras classes, como por exemplo, 'programa', 'bolsa', 'endereco', 'uf', 'instituição', 'grandeArea', 'area', 'subarea', permitindo a integração de todos os elementos.

Por sua vez, o modelo de aplicação mostra como o controller gerencia o cadastro de novos beneficiários, trazendo as regras em relação à inclusão e alteração dos dados do banco de dados, sendo previstas regras e restrições para esse processo.

4.2 Camada de Acesso a Dados

Nesta camada de acesso de dados foi desenvolvida pensando na comunicação entre o software e o banco de dados. Como principais função, o desenvolvimento prevê métodos para armazenar e recuperar dados do banco de dados, bem como gerenciar a conexão com

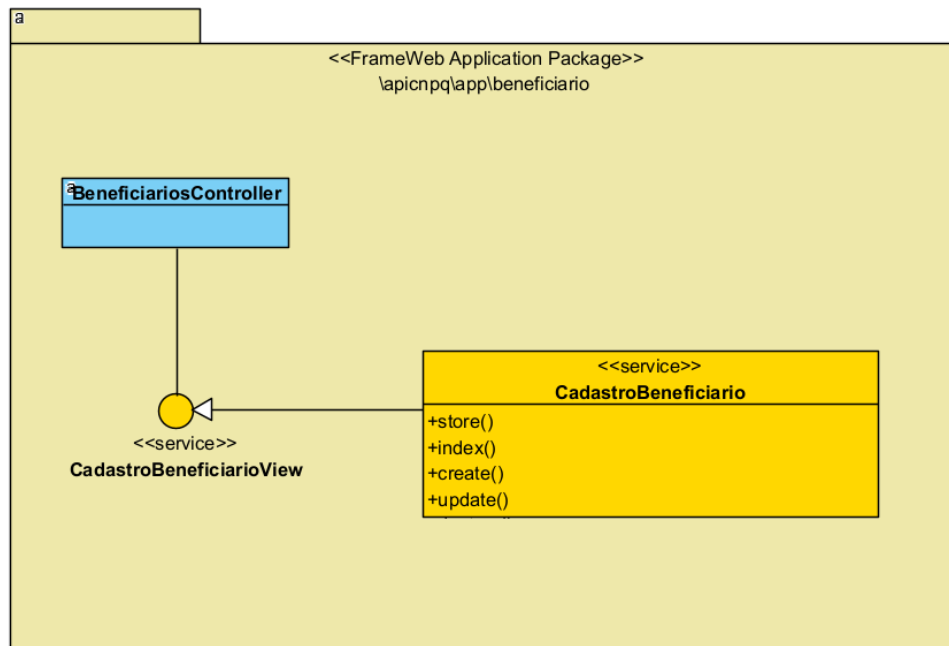


Figura 3 – Modelo de Aplicação

o banco de dados, a partir do Modelo de Persistência.

É importante destacar que o framework Laravel não utiliza o modelo DAO (Data Access Object) separadamente, pois adota uma abordagem mais flexível e orientada a objetos com base no padrão do Eloquent ORM. O Eloquent ORM fornece uma forma conveniente de mapear objetos para tabelas de banco de dados e oferece métodos para criar, recuperar, atualizar e excluir registros de forma intuitiva.

4.3 Camada de Apresentação

Nesta camada de apresentação vamos mostrar as regras relacionadas à interface do usuário. Essa camada é responsável por apresentar informações e interações do sistema para o usuário final, para tanto, segue informações básicas sobre como foi estruturado o sistema sob a ótica do front-end, onde os usuários podem visualizar e interagir com as funcionalidades do sistema

O modelo de navegação de cadastro foi utilizado um formulário que utiliza uma tag input com a determinação do tipo de entrada para cada atributo, sendo determinadas regras de restrição na entrada desses dados. Foi desenvolvida, então, uma página que está vinculada ao beneficiário a partir de uma pasta pública, e esta página contém um formulário de cadastro que vai interagir com o controller que gerencia o front end com os campos obrigatórios e acessar o back-end quando todos os dados forem devidamente cadastrados pelo usuário. Desta forma, a função store, por meio do Laravel, recebe um request por meio do método post, performando uma transformação interna que gera um

arquivo no formato Json para alimentar o banco de dados com as informações preenchidas pelo usuário.

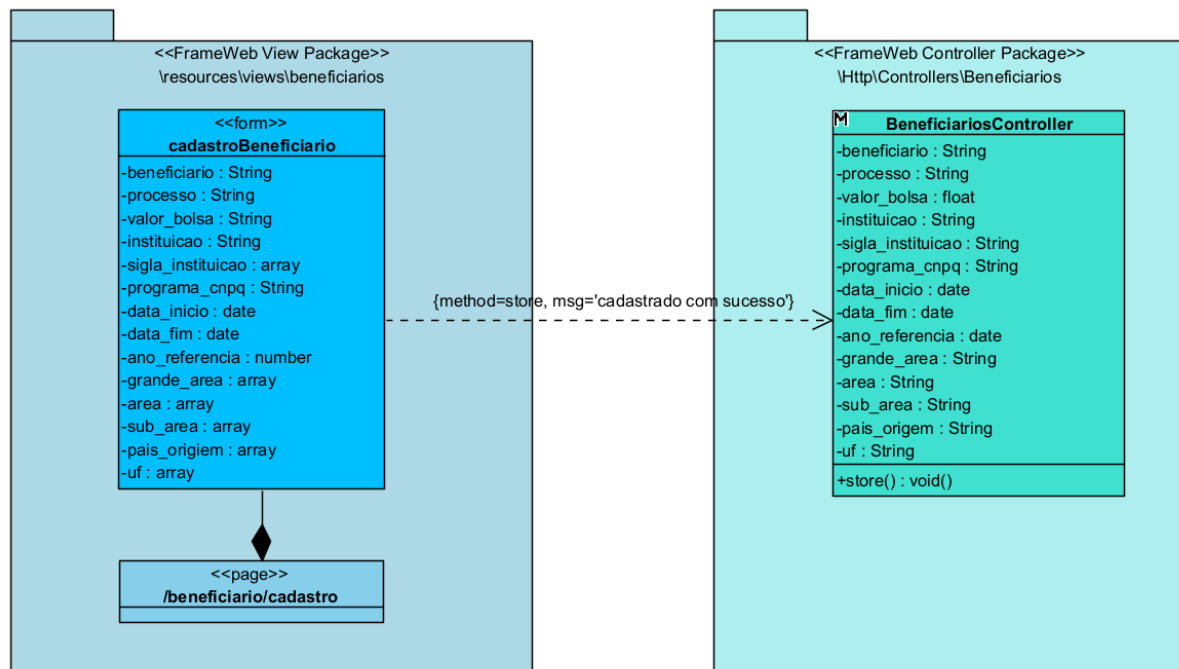


Figura 4 – Modelo de Navegação - cadastro de beneficiário

Como estratégia, esses input de dados foram armazenados em variáveis, de tal forma que é possível fazer uma verificação com características e regras de verificação para que esses dados sejam devidamente inseridos.

Foi aplicada a mesma abordagem em que uma página contém um relatório com filtros para permitir que o usuário insira informações e, ao ser processado no backend, seja possível realizar uma consulta ("select") no banco de dados e obter os dados desejados. Uma nova funcionalidade foi adicionada para que, ao buscar os valores do banco de dados, seja feita uma operação de atualização dos dados, calculando a soma dos valores pagos como Bolsa. Isso amplia os recursos do sistema para incluir as operações de leitura e atualização dos dados, resultando na criação de uma tela agregada por ano. Com essa nova funcionalidade, o sistema utiliza três métodos principais: "index", "create" e "store", para lidar com a exibição, criação e armazenamento dos dados, respectivamente.

Por fim, será gerado pelo método delete para fazer as alterações de exclusões do banco de dados.

O consumo de dados será feito a partir da leitura de um arquivo com extensão csv para popular algumas tabela do banco de dados, tais como UF e grande área. Em resumo, para criarmos um modelo de persistência de banco de dados, começaremos obtendo arquivo csv de consulta pública do CNPq. Faremos chamadas para obter informações sobre pesquisadores, projetos, publicações, entre outros. Em seguida, iremos adicionar esse

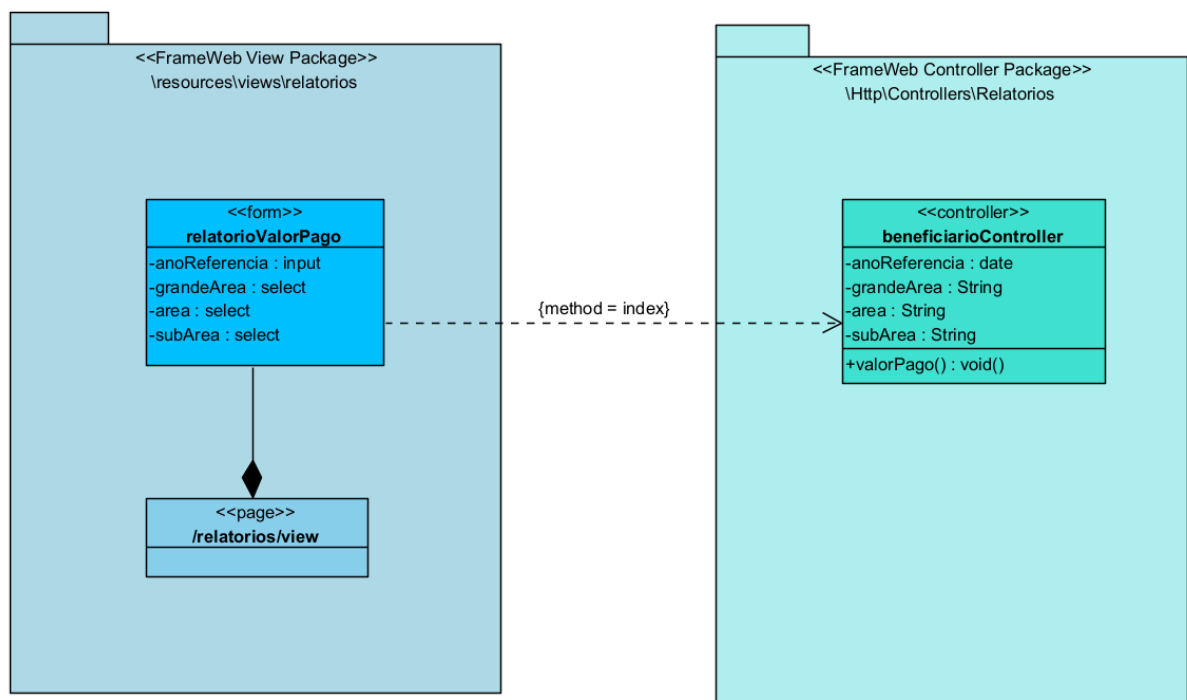


Figura 5 – Modelo de Navegação - relatório de bolsas

arquivo em nossa aplicação para popular algumas tabelas do banco de dados visando a facilidade de preenchimento dos cadastros dos bolsistas.

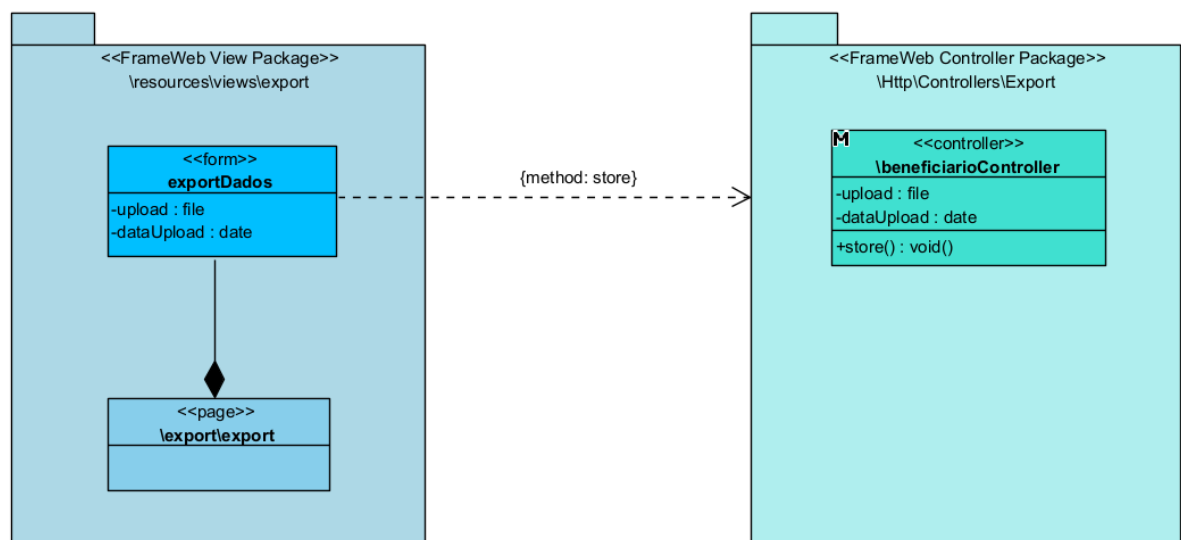


Figura 6 – Modelo de Navegação - relatório de bolsas

Com o banco de dados populado, poderemos criar um formulário para que os usuários possam realizar operações de CRUD (Criar, Ler, Atualizar e Excluir) nos dados. Esse formulário será a interface pela qual os usuários poderão inserir novos registros, visualizar informações existentes, atualizar dados ou excluir registros indesejados. Vale ressaltar que nesta etapa é importante aplicar validações adequadas no formulário para

garantir que os dados inseridos pelos usuários sejam consistentes e corretos. Isso ajudará a evitar erros e manter a integridade dos dados armazenados no banco de dados.

Com esse modelo de persistência de banco de dados em vigor, será possível aproveitar os dados extraídos do arquivo csv do CNPq e permitir que os usuários gerenciem esses dados de forma eficiente e intuitiva por meio do formulário.

Referências

Marijn Haverbeke. *Eloquent JavaScript*. 2023. Disponível em: <<https://eloquentjavascript.net/>>. Acesso em: 25 de abril 2023. Citado na página 2.

CNPq. *Conselho Nacional de Desenvolvimento Científico e Tecnológico*. 2023. Disponível em: <<https://www.gov.br/cnpq/pt-br>>. Acesso em: 25 de abril 2023. Citado na página 2.

Laravel LLC. *The PHP Framework for Web Artisans*. 2023. Disponível em: <<https://laravel.com/>>. Acesso em: 25 de abril 2023. Citado na página 2.

OpenJS Foundation. *Node.js v20 is now available!* 2023. Disponível em: <<https://nodejs.org/en>>. Acesso em: 25 de abril 2023. Citado na página 2.

REENSKAUG, T. M. H. The original mvc reports. 1979. Citado na página 4.

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 2 vezes nas páginas 2 e 5.

The PHP Group. *PHP Documentation*. 2023. Disponível em: <<https://www.php.net/docs.php>>. Acesso em: 25 de abril 2023. Citado na página 2.