Documento de Projeto de Sistema

VixCourts

Registro de Alterações:

Versão	Responsável	Data	Alterações
1.0	Gabriel/Silas	23/09/2019	Versão Final

Vitória, ES

2019

1 Introdução

Este documento apresenta o documento de projeto (*design*) arquitetural do sistema VixCourts. Este documento está organizado da seguinte forma: a Seção 2 apresenta a plataforma de software utilizada na implementação da ferramenta; por fim, a Seção 3 apresenta o projeto da arquitetura de software e suas subseções explicam cada uma de suas camadas.

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
Java EE	11	Conjunto de especificação de APIs e tecnologias, que são implementadas por programas servidores de aplicação.	Redução da complexidade do desenvolvimento, implantação e gerenciamento de aplicações Web a partir de seus componentes de infra-estrutura prontos para o uso.
Java	11	Linguagem de programação orientada a objetos e independente de plataforma.	Escrita do código-fonte das classes que compõem o sistema.
JSF	2.3	API para a construção de interfaces de usuários baseada em componentes para aplicações Web	Criação das páginas Web e sua comunicação com as classes Java.
EJB	4.0.9	API para construção de componentes transacionais gerenciados por *container*.	Implementação das regras de negócio em componentes distribuídos, transacionais, seguros e portáveis.
JPA	2.1	API para persistência de dados por meio de mapeamento objeto/-relacional.	Persistência dos objetos de domínio sem necessidade de escrita dos comandos SQL.
Jbutler	1.4	Um framework CRUD para ser utilizado em aplicações Webs que seguem a arquitetura de três camadas.	Facilitar a integração entre as três camadas do sistema.
CDI	2.0	API para injeção de dependências.	Integração das diferentes camadas da arquitetura.
Facelets	2.0	API para definição de decoradores (*templates*) integrada ao JSF.	Reutilização da estrutura visual comum às paginas, facilitando a manutenção do padrão visual do sistema.
PrimeFaces	7.0	Conjunto de componentes visuais JSF *open source*.	Reutilização de componentes visuais Web de alto nível.
MySQL Workbench	8.0.12	Sistema Gerenciador de Banco de Dados Relacional gratuito.	Armazenamento dos dados manipulados pela ferramenta.
WildFly	17	Servidor de Aplicações para Java EE.	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.
TeX Live	2019	Implementadão do LaTeX	Documentação do projeto arquitetural do sistema.
Overleaf	2019	Editor de LaTeX.	Escrita da documentação do sistema, sendo usado o *template abnTeX*.[1]
Eclipse Java EE IDE for Web Developers	4.8	Ambiente de desenvolvimento (IDE) com suporte ao desenvolvimento Java EE.	Implementação, implantação e testes da aplicação Web Java EE.
Apache Maven	3.5	Ferramenta de gerência/-construção de projetos de software.	Obtenção e integração das dependências do projeto.

3 Arquitetura de Software

A arquitetura de software do sistema VixCourts segue a arquitetura padrão sugerida pelo FrameWeb (SOUZA, 2007; SOUZA; FALBO; GUIZZARDI, 2009) baseada no padrão Camada de Serviço (FOWLER, 2002). A Figura 1 ilustra a arquitetura e indica onde atuam os *frameworks* para o desenvolvimento Web, listados na Tabela 1.

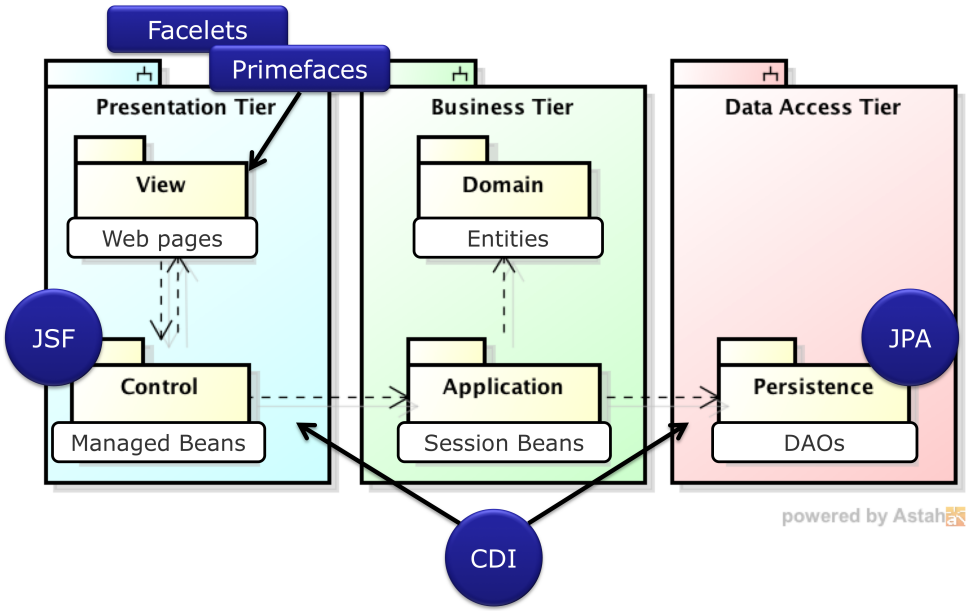

Figura 1 – Arquitetura padrão proposta pelo FrameWeb.

Nas próximas seções, serão apresentados diagramas FrameWeb relativos a cada uma das camadas da arquitetura do sistema.

3.1 Camada de Apresentação

A Figura 2 apresenta o modelo de navagação da funcionalidade **Agendar Quadra**. Na página do mapa é possível visualizar as quadras cadastradas na região. Ao selecionar um quadra um formulário para realizar o agendamento de uma quadra é aberto. Outra forma de selecionar uma quadra é utilizando a funcionalidade **Buscar Quadras Disponíveis**, desta forma o sistema irá recomendar as quadras disponíveis que estejam próximas do usuário de acordo com o tipo de quadra desejada e o horário especificado pelo usuário. Com a quadra selecionada o usuário pode realizar o agendamento.

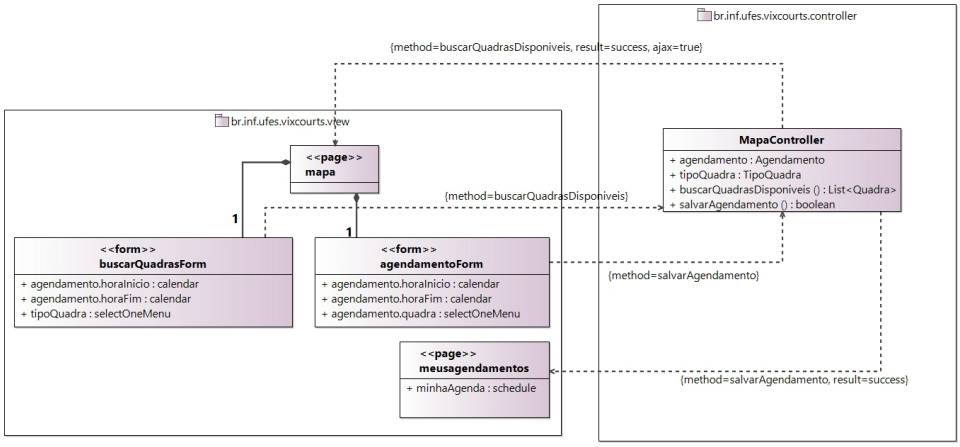

Figura 2 – Modelo de Navegação - Buscar Quadra

A Figura 3 é o modelo de navegação para buscar partidas. Ao acessar o mapa o usuário tem a opção de buscar por uma partida que já foi agendada. Para realizar a busca é preciso fornecer a modalidade e o endereço, dessa forma o sistema irá recomendar uma partida que está agendada em alguma quadra na região próxima ao usuário. Com a partida recomendada, é necessário enviar uma solicitação para participar da partida.

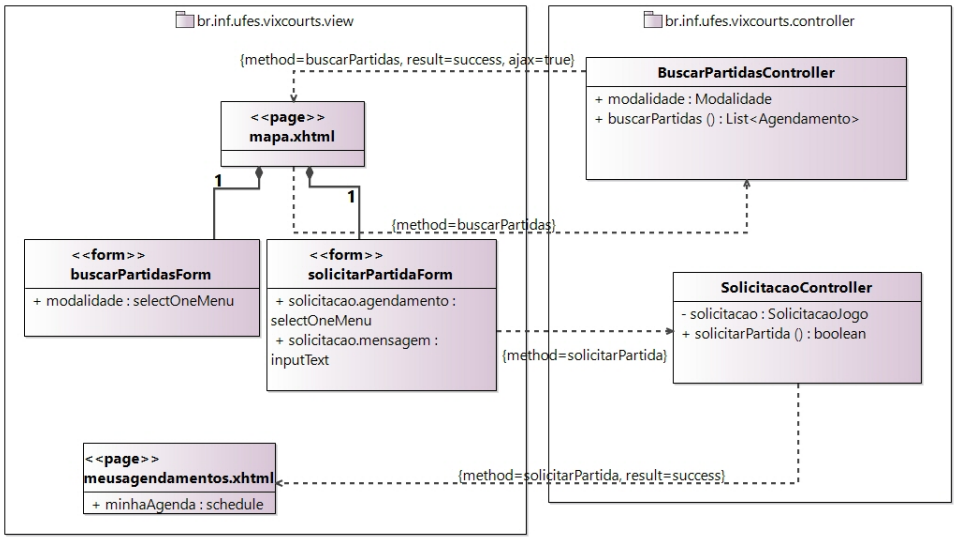

Figura 3 – Modelo de Navegação - Socilitar Partida

Por fim, a Figura 4 apresenta o modelo de navegação para criação de usuário. Na página de criação de usuário é apresentado um formulário com os campos que devem ser preenchidos. Em caso de sucesso o sistema irá redirecionar o usuário para a página inicial do sistema.

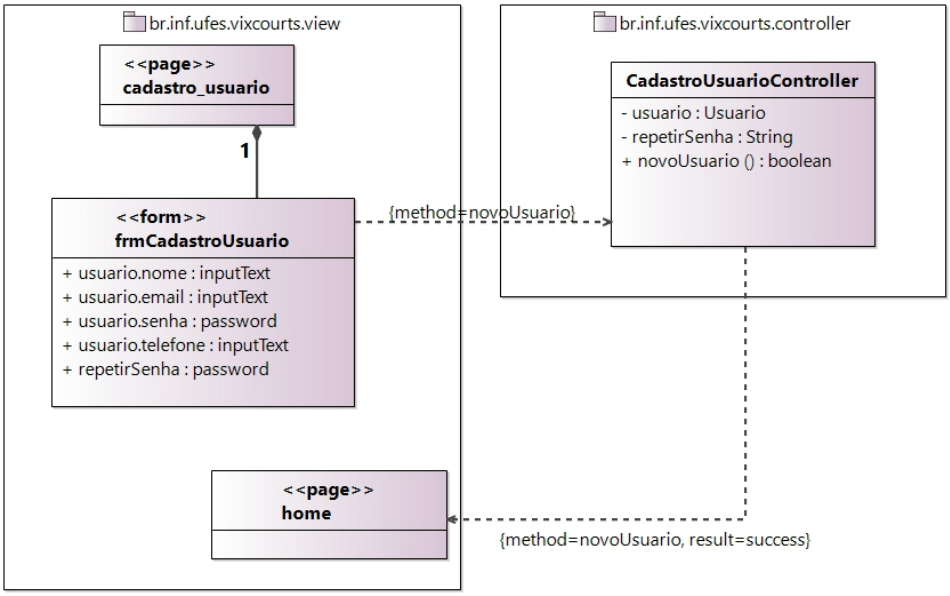

Figura 4 – Modelo de Navegação - Criação de Usuário

3.2 Camada de Negócio

Na Figura 5 o modelo de entidade do sistema, sendo possível visualizar as entidades que compõem o sistema e suas relações. Um **Usuário** *realiza* **Agendamento** de um determinada **Quadra**. Esse agendamento possui um hora de inicio e fim. Além de realizar agendamento, o **Usuário** pode administrar e integrar um ou mais **Times**.

Um usuário também pode solicitar participar de uma partida. Essa relação é representada pela classe **SolicitacaoJogo**, que relaciona o um usuário com um agendamento.

O modelo apresentado não conta com todas as definições do mapeamento para o banco de dados, isso se deve ao fato da ferramenta não exibir visualmente as opções de null ou not null selecionadas (Issue 22).

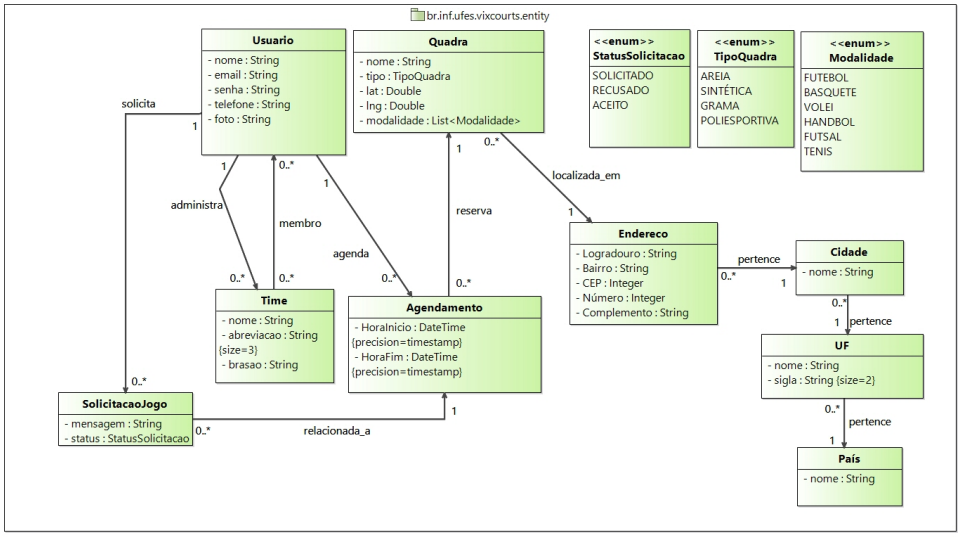

Figura 5 – Modelo de Entidade

A Figura 6 apresenta o modelo de aplicação do projeto. Os controladores se comunicam com suas respectivas camadas de aplicação, tais camadas utilizam os DAOs, que realizam a comunicação entre a aplicação e o banco de dados.

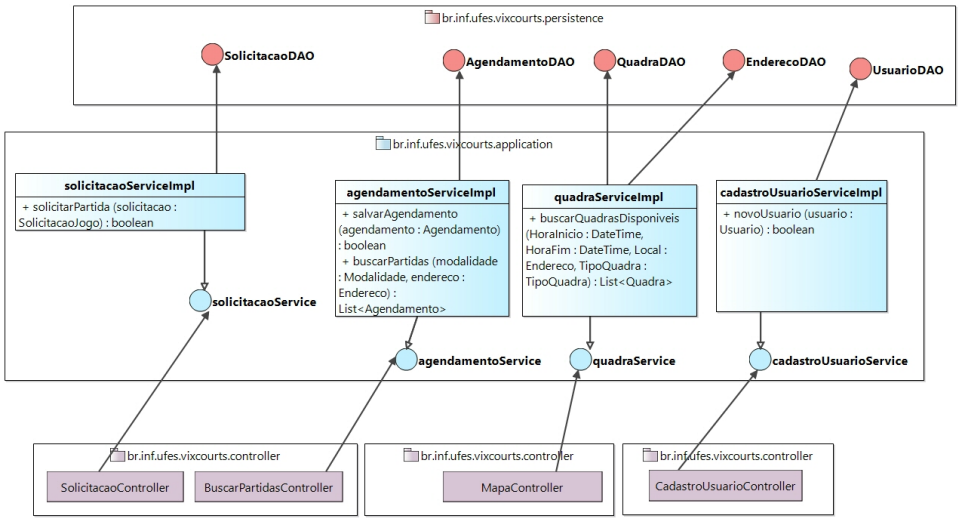

Figura 6 – Modelo de Aplicação

3.3 Camada de Acesso a Dados

Na camada de persistência foi feito um modelo para representar os DAOs responsáveis pelas respectivas entidades que compõe o sistema.

Na implementação deste trabalho foi utilizado o Jbutler, um utilitário que fornece os métodos básicos para manipulação das entidades (Criar, Alterar, Remover e Visualizar). Cada DAO possui métodos que proveem informações relevantes para a aplicação. Como exemplo, no *AgendamentoDAO* existe métodos para recuperação de *Agendamentos* dado

uma modalidade e seu endereço. A Figura 7 e **??**, apresenta o modelo de persistência da aplicação.

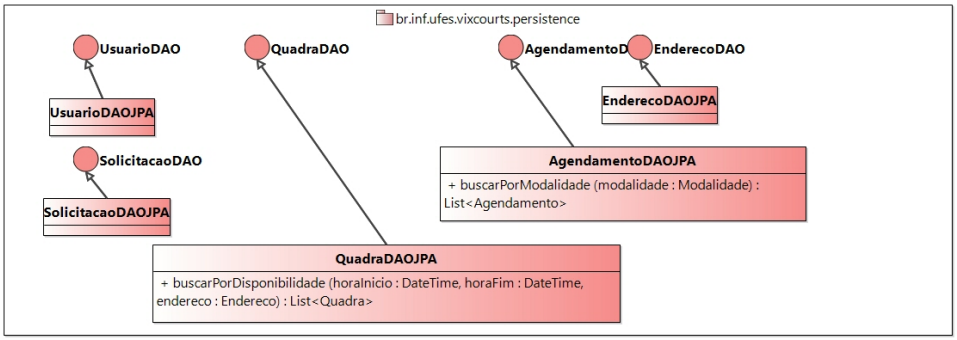

Figura 7 – Modelo de Persistência

Referências

FOWLER, M. *Patterns of Enterprise Application Architecture*. 1. ed. [S.l.]: Addison-Wesley, 2002. ISBN 9780321127426. Citado na página 4.

SOUZA, V. E. S. *FrameWeb: um Método baseado em Frameworks para o Projeto de Sistemas de Informação Web*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, 2007. Citado na página 4.

SOUZA, V. E. S.; FALBO, R. A.; GUIZZARDI, G. Designing Web Information Systems for a Framework-based Construction. In: HALPIN, T.; PROPER, E.; KROGSTIE, J. (Ed.). *Innovations in Information Systems Modeling: Methods and Best Practices*. 1. ed. IGI Global, 2009. cap. 11, p. 203–237. ISBN 9781605662787. Disponível em: <http://www.igi-global.com/reference/details.asp?id=33232>. Citado na página 4.