



Documento de Projeto de Sistema

Social Meet Scheduler

Registro de Alterações:

Versão	Responsável	Data	Alterações
1.0	Ádler Oliveira Silva Neves	22/09/2019	Versão Inicial

Vitória, ES

2019

1 Introdução

Este documento apresenta o documento de projeto (*design*) arquitetural do sistema Social Meet Scheduler. 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
Python	3.7	Linguagem de programação orientada a objetos e independente de plataforma.	Escrita do código-fonte das classes que compõem o sistema.
Django	2.2.5	Conjunto de APIs e tecnologias voltadas para a construção rápida de aplicações rápidas, seguras e escaláveis.	Redução da complexidade do desenvolvimento, implantação e gerenciamento de aplicações Web a partir de seus componentes de infra-estrutura prontos para o uso.
Django Admin Site	2.2.5	Aplicação drop-in que provê um gerenciamento facilitado dos dados do sistema por administradores.	Redução da complexidade de administração do sistema.
Django Rosetta	0.9.3	Aplicação drop-in que provê um gerenciamento facilitado da tradução de páginas da interface.	Redução do esforço de traduzir o sistema para diferentes idiomas.
Django Templates	2.2.5	API para a construção de páginas baseadas em templates	Criação das páginas Web do sistema, reutilizando a estrutura visual comum às páginas, facilitando a manutenção do padrão visual do sistema.
Django Page Components	0.1	API para criação de componentes reutilizáveis	Criação e reutilização de componentes visuais Web de alto nível, componentes customizados de forma a facilitar a manutenção do padrão visual do sistema.
Django Forms	2.2.5	API para definição e tratamento de formulários	Criação de formulários e seu posterior tratamento, facilitando o desacoplamento das camadas.
Django ORM	2.2.5	API para persistência de dados por meio de mapeamento objeto-/relacional.	Persistência dos objetos de domínio sem necessidade de escrita dos comandos SQL.
PyCDI	1.1	API para injecção de dependências.	Integração entre diferentes camadas da arquitetura.
PostgreSQL	11.5	Sistema Gerenciador de Banco de Dados Relacional gratuito.	Armazenamento dos dados manipulados pela ferramenta.
WSGIServer	0.2	Servidor de Aplicações e arquivos estáticos para Python WSGI.	Execução das APIs citadas acima em ambiente de desenvolvimento.
Gunicorn	19.9.0	Servidor de Aplicações para Python WSGI.	Execução das APIs citadas acima em ambiente de produção e hospedagem da aplicação Web, dando acesso aos usuários via HTTP.
NGINX	1.17.3	Servidor de arquivos estáticos e proxy-reverso.	Servir arquivos estáticos e realizar o proxy-reverso no Gunicorn, adicionando a camada TLS em ambiente de produção.

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	1.0.0. 201908 181134	Ferramenta CASE do método FrameWeb.	Criação dos modelos de Entidades, Aplicação, Persistência e Navegação.
PlantUML	1.2019.11	Renderizador de diagramas.	Renderização gráfica de diagramas a partir de texto que o FrameWeb não gera.
TeX Live	2019. 51075-3	Implementação do L ^A T _E X	Documentação do projeto arquitetural do sistema.
GNOME L ^A T _E X	3.32.0	Editor de LaTeX.	Escrita da documentação do sistema, sendo usado o <i>template abn-TeX</i> . ¹
Eclipse IDE for Enterprise Java Developers	4.12.0	Ambiente de desenvolvimento (IDE) com suporte ao desenvolvimento Java EE.	Ambiente de execução do plug-in do FrameWeb Editor.
VSCodium	1.37.1	Editor de texto com suporte a extensões com suporte a Python.	Desenvolvimento do código-fonte da aplicação.
pip	19.0.3	Ferramenta de gerência de dependências de software python.	Obtenção das dependências do projeto.
virtualenv	16.1.0	Ferramenta para isolar ambientes de desenvolvimento python.	Isolar as dependências do software do restante do sistema.
Automake	1.16.1	Ferramenta de automação em script	Desenvolver atalhos convenientes para várias tarefas.

3 Arquitetura de Software

A arquitetura de software do sistema Social Meet Scheduler segue uma arquitetura análoga à padrão sugerida pelo FrameWeb (SOUZA, 2007; SOUZA; FALBO; GUIZZARDI, 2009) baseada no padrão Camada de Serviço (FOWLER, 2002). A Figura 1 ilustra tal arquitetura sugerida e indica onde atuariam os *frameworks* para desenvolvimento Web Java com *JSF*, *JPA*, *CDI*, *PrimeFaces* e *Facelets*.

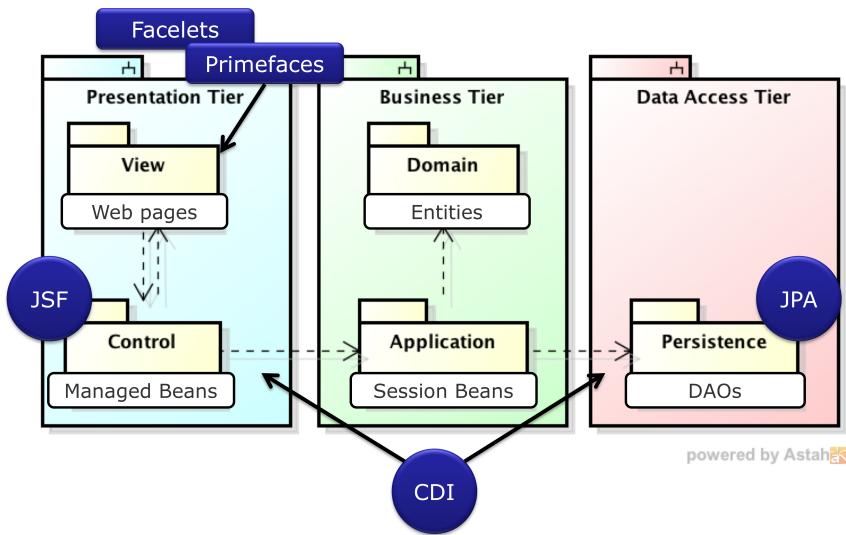


Figura 1 – Arquitetura padrão proposta pelo FrameWeb.

Entretanto, como tal arquitetura não se mantém no padrão Django, que é composta de *models*, *views* e *templates* com responsabilidades diferentes da dos elementos de mesmo nome na arquitetura Java EE. Portanto é necessário fazer aproximações para ambos os modelos dizerem a mesma coisa com palavras diferentes. Se adicionarmos uma camada de serviços entre os *models* e *views*, podemos obter uma arquitetura como a da figura 2, que é bastante similar à 1. Como Django usa Active Record e não Data Access Objects, a camada de acesso ao dado foi ignorada.

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

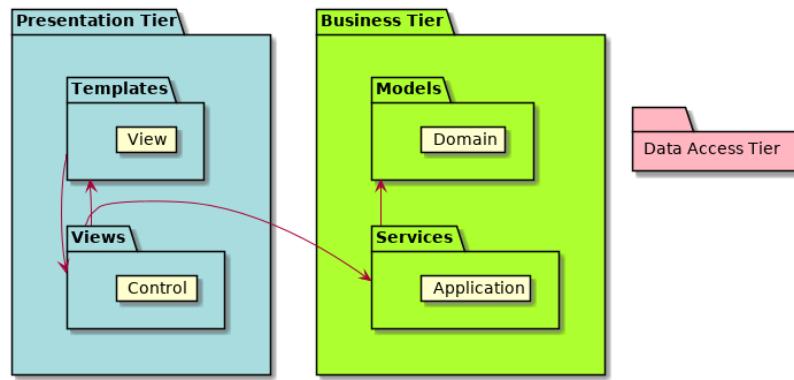


Figura 2 – Arquitetura que busca aproximar o modelo do Django à do FrameWeb.

3.1 Camada de Apresentação

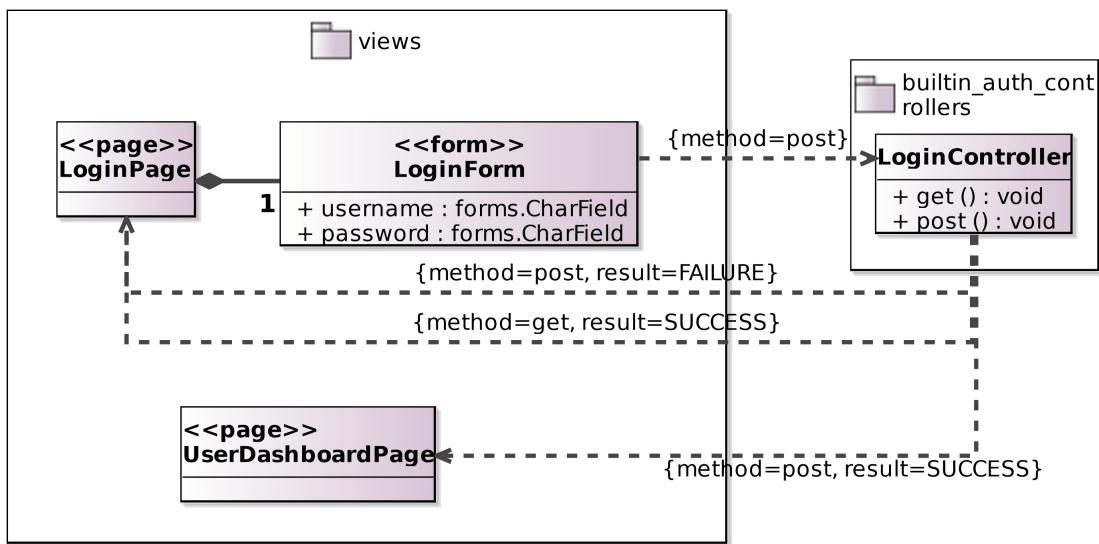


Figura 3 – Modelo de Navegação do Social Meet Scheduler – Login

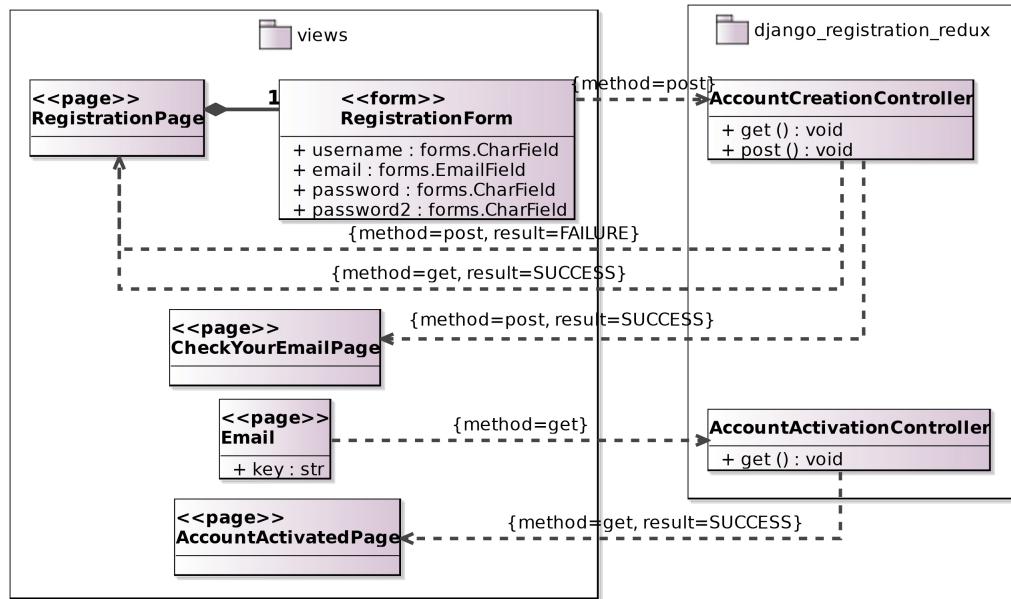


Figura 4 – Modelo de Navegação do Social Meet Scheduler – Criar conta

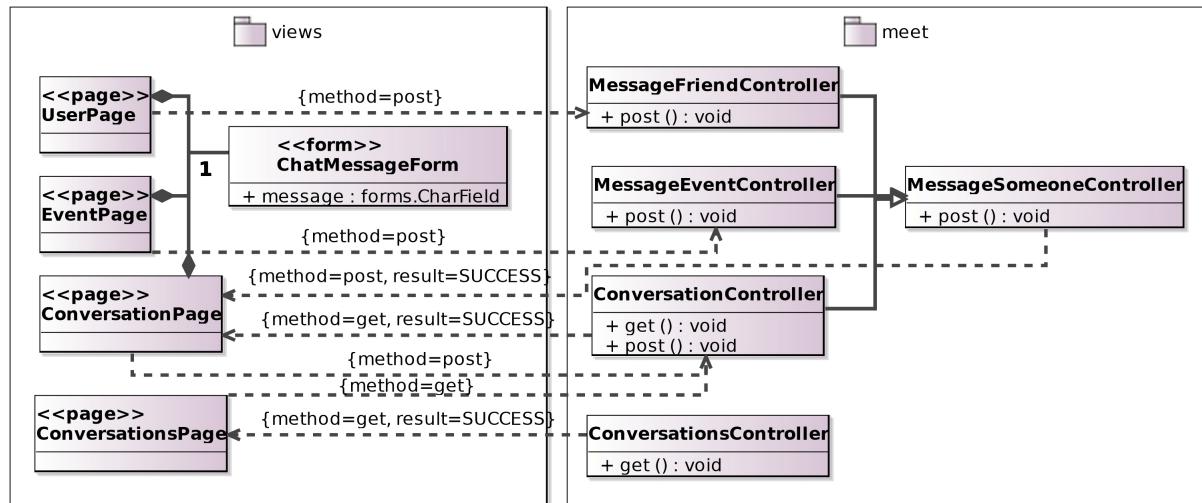


Figura 5 – Modelo de Navegação do Social Meet Scheduler – Iniciar e continuar conversas

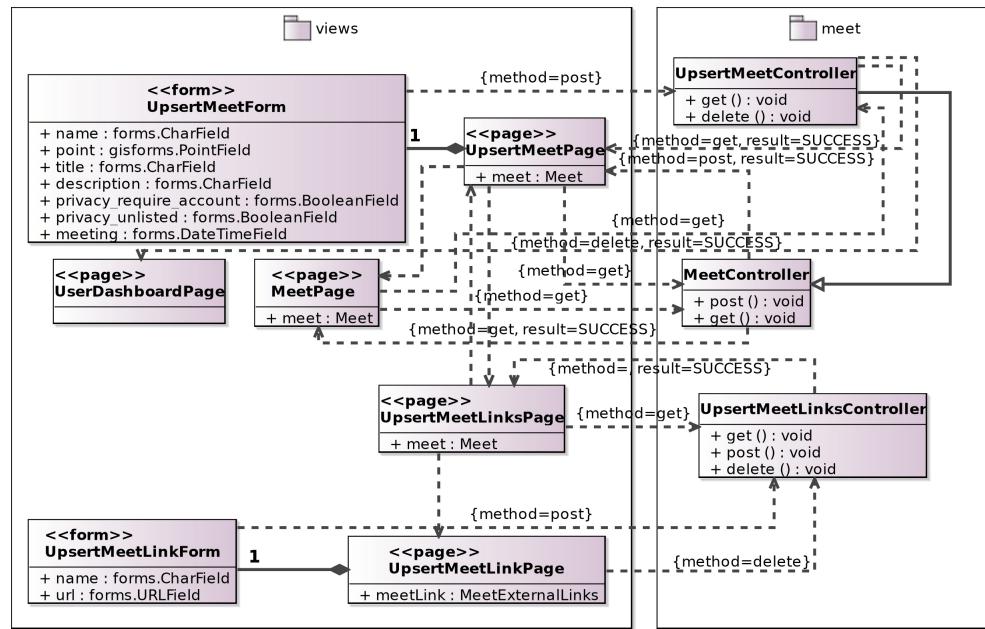


Figura 6 – Modelo de Navegação do Social Meet Scheduler – Gerenciar encontros

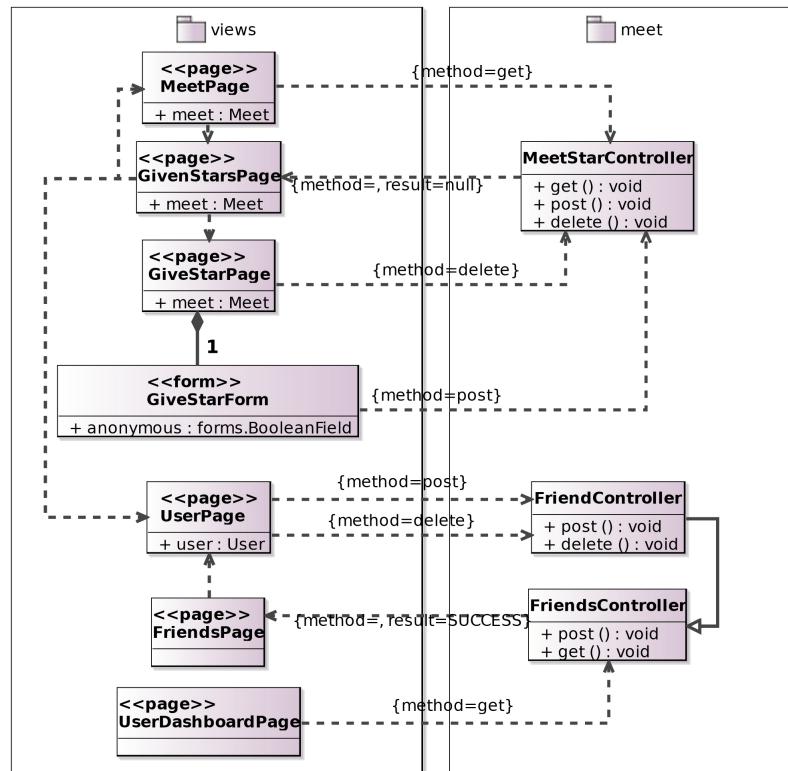


Figura 7 – Modelo de Navegação do Social Meet Scheduler – Participar de encontros e adicionar/aceitar/rejeitar/desistir pedidos de amizade

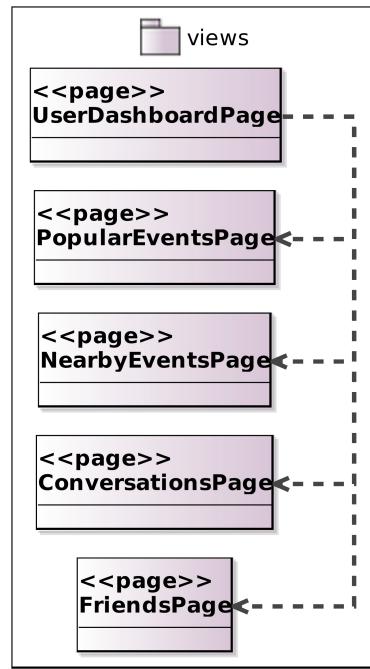


Figura 8 – Modelo de Navegação do Social Meet Scheduler – O caminho do usuário de sua homepage até um ponto em comum com as funcionalidades mencionadas anteriormente; eventuais controladores foram omitidos

3.2 Camada de Negócio

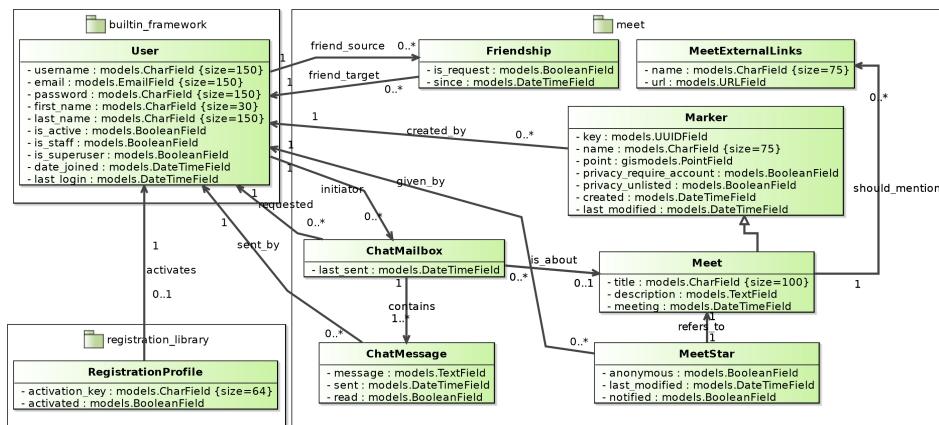


Figura 9 – Modelo de Entidades do Social Meet Scheduler.

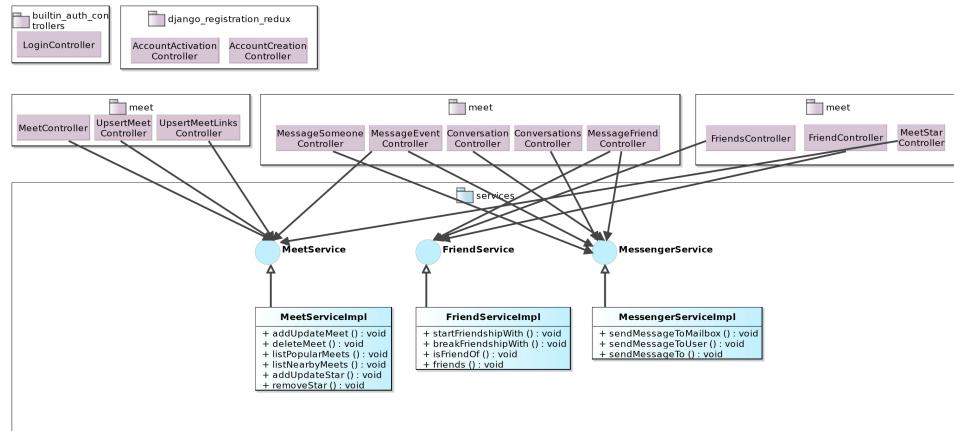


Figura 10 – Modelo de Aplicação do Social Meet Scheduler.

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](#).