

COLÉGIO ESTADUAL DO PARANÁ
CURSO TÉCNICO EM INFORMÁTICA

WILLIAM CLAUDE TUMÉO RAUSIS

SCOUT CAMP: Gerenciamento de grupos

CURITIBA
2016

WILLIAM CLAUDE TUMÉO RAUSIS

SCOUT CAMP: Gerenciamento de grupos

Trabalho de Conclusão de Curso Técnico, apresentado à disciplina de Análise e Projetos, do Curso Técnico em Informática, do Colégio Estadual do Paraná, como requisito para obtenção do título de Técnico em Informática.

Orientador: Prof. Fernando Emrich Pinto Maia

CURITIBA

2016

Dedico este trabalho a minha querida Gabrielle Lira, que (quase) sempre me atura.

AGRADECIMENTOS

Ao meu amigo Lucas Flicky que foi o responsável pela idealização do sistema do grupo, criando a ideia do sistema e ajudando nos recursos da página.

Ao grupo de amigos “Os Mitadores” que me ajudou a crescer como pessoa e programador.

Ao meu amigo Hugo José, que foi um dos primeiros a me incentivar na carreira do desenvolvimento, principalmente o desenvolvimento de jogos.

A Pedro Casanova, criador do portal Cogumelando, que me deu uma oportunidade, ajudando no meu reconhecimento como profissional.

A Tom Rothamel, criador da Visual Novel Engine Ren'Py, que me fez conhecer a linguagem Python e me ajudou a aprendê-la.

Ao meu amigo Emanuel Victor, por ajudar em projetos que me proporcionaram reconhecimento internacional.

A minha mãe, por todos os ensinamentos e por ter apoiado sempre.

E a todos os meus professores que foram os maiores responsáveis por tudo o que aprendi e ainda aprenderei.

A maioria dos bons programadores programam não porque esperam receber o pagamento ou obter bajulação por parte do público, mas porque é divertido programar.

Linus Trovalds

RESUMO

TUMEO, William. Scout Camp: Gerenciamento de grupos. 2016. 38 f. Monografia de Trabalho de Conclusão de Curso – Curso Técnico em Informática, Colégio Estadual do Paraná. Curitiba, 2016.

Esta monografia trata de um projeto de framework para o gerenciamento de grupos de forma local, sem utilizar banco de dados.

Inspirado no conceituado Jekyll, com o qual você cria os dados de forma local e o programa exporta uma página web pronta a partir de templates. A diferença sendo que o mesmo é voltado especificamente para blogs.

Palavras-chave: python, framework, html, template, grupos.

ABSTRACT

TUMEO, William. Scout Camp: Group management. 2016. 38 f. Monografia de Trabalho de Conclusão de Curso – Curso Técnico em Informática, Colégio Estadual do Paraná. Curitiba, 2016.

This monograph describes a project for a framework built to manage groups locally, without using a database.

Inspired on the well-known Jekyll, which is used to create data locally and the tool exports a ready-to-go webpage from templates. The difference here being that Jekyll is designed to be used with blogs.

Keywords: python, framework, html, template, groups.

LISTA DE FIGURAS

Figura 1 - Exemplo de painel para grupos.....	23
Figura 2 - Método de desenvolvimento Extreme Programming.....	25
Figura 3 - Diagrama de classes.....	26
Figura 4 - Diagrama de caso de uso (Render).....	27
Figura 5 - Diagrama de caso de uso (Server).....	28
Figura 6 - Diagrama de atividade.....	30
Figura 7 - Programa sendo executado em CLI.....	31
Figura 8 - Teste usando o programa pronto.....	32

LISTA DE TABELAS

Tabela-1--Etapas-do-trabalho.....	16
-----------------------------------	----

LISTA DE SIGLAS

PHP	Hypertext Preprocessor (processador de hipertexto).
HTML	Hyper Text Markup Language (linguagem de marcação de hipertexto).
YAML	Ain't Markup Language (não é uma linguagem de marcação).
CSS	Cascading Style Sheets (folha de estilo em cascata)
SQL	Structured Query Language (linguagem de consulta estruturada)
XML	eXtensible Markup Language (linguagem de marcação extensível)
C	linguagem de programação
MVC	Model-View-Controller (modelo-visão-controlador)
JSON	JavaScript Object Notation (notação de objetos do JavaScript)
INI	
IDE	Integrated development environment (ambiente de desenvolvimento integrado)
VCS	Version Control System (sistema de controle de versão)
SCM	Source Control Management (gerenciamento de controle do fonte)
CASE	Computer-aided software engineering (engenharia de software assistida por computador)
CLI	Command-Line Interface (interface da linha de comando)
SQLite	Versão alternativa e leve do SQL

Sumário

1. INTRODUÇÃO.....	13
2. OBJETIVOS.....	14
2.1.1 Objetivo Geral.....	14
2.1.2 Objetivos Específicos.....	14
3. JUSTIFICATIVA.....	15
4. ESTRUTURA DO TRABALHO.....	16
5. REFERENCIAL TEÓRICO.....	17
5.1 PROCESSADORES DE HTML.....	17
5.2 LINGUAGEM PYTHON.....	18
5.3 LINGUAGENS DE TEMPLATE.....	19
5.1 LINGUAGENS DE NOTAÇÃO DE OBJETOS.....	19
6. MATERIAIS E MÉTODO.....	21
6.1 MATERIAIS.....	21
6.1.1 Linguagem Python.....	21
6.1.2 Linguagem e biblioteca YAML.....	21
6.1.3 Biblioteca Pystache.....	21
6.1.4 Linguagens HTML, CSS e JavaScript.....	21
6.1.5 Bootstrap.....	22
6.1.6 Editor Atom.....	22
6.1.7 Git.....	22
6.1.8 Linguagens Batch e Shell.....	22
6.1.9 Astah Community.....	22
6.1.10 LibreOffice.....	22
6.2 MÉTODO.....	23
6.2.1 Levantamento de requisitos.....	23
6.2.2 Método de desenvolvimento.....	24
6.2.3 Modelagem do sistema.....	25
6.2.4 Funcionamento do programa.....	30
7. RESULTADOS E DISCUSSÕES.....	32
7.1 IMPLEMENTAÇÃO DO SISTEMA.....	32

7.2 DADOS OBTIDOS E DISCUSSÕES.....	33
8. CONCLUSÃO.....	34
9. REFERÊNCIAS.....	35

1. INTRODUÇÃO

Hoje em dia, surgem muitos grupos pela internet, podendo ser grupos grandes ou pequenos e tendo ou não algum tipo de objetivo.

Muitos desses grupos tem um certo objetivo, como grupos de estudos, ou grupos de jogos. Como o sistema é focado nesses grupos com esse tipo de objetivo, o maior exemplo para representar o sistema são os grupos de escoteiros.

O sistema terá algumas características que tem em grupos de escoteiros, como medalhas e patentes, entre outras características.

Como ele tem o objetivo de ser acessado a qualquer momento, a melhor forma de se fazer é com uma página web. Essa página web poderia ser facilmente criada para ser gerenciada por um banco de dados usando PHP. Porém, o sistema tem o foco em grupos que não tem o interesse em gastar dinheiro comprando host, muito menos host com suporte a banco de dados.

O sistema será criado para ser implementado da mesma forma que o famoso Jekyll, que é um Framework escrito em Ruby que funciona como um processador de HTML com suporte a postagens baseado em dados locais (sem utilizar banco de dados), utilizando-se de linguagens de marcação como YAML e Markdown.

2. OBJETIVOS

O objetivo geral apresenta o resultado principal da realização deste trabalho e os objetivos específicos complementam este objetivo geral.

2.1.1 Objetivo Geral

Implementar um framework para gerar páginas HTML com o objetivo de gerenciar grupos.

2.1.2 Objetivos Específicos

- Proporcionar o gerenciamento de grupos do tipo “escoteiro” de forma que não seja necessário utilizar um servidor pago.
- **Implementar** a personalização dos grupos usando o método de templates.
- Facilitar as configurações para que possa subir os arquivos em uma página do Facebook.

3. JUSTIFICATIVA

A ideia do sistema se deu pela necessidade própria de gerenciar um grupo de amigos que realiza eventos e que possui essa distribuição de medalhas de acordo com realizações dentro do grupo, como um grupo de escoteiros mesmo.

A ideia de se fazer o sistema de forma local foi pelo motivo de que o sistema teria que ser implementado principalmente em uma página do Facebook, que só suporta páginas estáticas usando somente HTML, CSS e JavaScript.

O sistema atual já poderia ser implementado, mas só poderia ser usado por pessoas que já entendem de programação, então a justificativa desse sistema é deixá-lo o mais simples de usar possível.

4. ESTRUTURA DO TRABALHO

A estrutura do trabalho contém uma relação dos capítulos e uma descrição sucinta do que cada um deles contém. Esta seção fornece uma visão geral do trabalho no sentido da sua estrutura em capítulos.

Tabela 1 – Etapas do trabalho

Etapas	Agosto	Setembro	Outubro	Novembro	Dezembro	Fevereiro
Levantamento de requisitos	X					
Especificações	X					
Arquitetura de software		X				
Teste/Documentação			X	X	X	X
Apresentação						X
Correção/ Manutenção						X

5. REFERENCIAL TEÓRICO

5.1 PROCESSADORES DE HTML

São programas específicos para gerar páginas web a partir de templates de forma local e funcionam como Frameworks. Dentre esses programas, o mais usado é o Jekyll, que deu a origem da ideia para o sistema em questão, justamente por ser muito personalizável e oferecer um sistema de base de dados local muito bem organizado.

Um framework naturalmente facilita a organização do código. A maioria das decisões sobre padrões já foi tomada pelos desenvolvedores do framework. Você está utilizando o conhecimento de outras pessoas para agilizar o seu processo de desenvolvimento. (BOTELHO, Bernardo)

A ideia é que você crie páginas e até mesmo um blog de forma estática, usando HTML, junto com algumas ferramentas que irão ajudá-lo a converter seus dados e templates em arquivos estáticos, pronto para ser publicado. Ele é baseado em vários formatos como Markdown para formatação de textos e posts e um padrão de template chamado Liquid com um pouco de YAML para exibir e guardar os dados das variáveis.

O conteúdo do seu site fica guardado nos arquivos de cada página. Você não precisa levantar um servidor de MySQL. Todas as informações do site estarão nos arquivos que você criar para cada página.

O formato YAML é conhecido pela facilidade de leitura. Ele foi criado para ser fácil da gente entender e também escrever. Ou seja, ele é um formato simples para escrevermos manualmente, mas também para manipularmos via programação.

O formato YAML pode ser implementado nas seguintes linguagens:

- JavaScript
- Objective-C
- Perl
- PHP
- Python

- Ruby
- Java
- Haskell
- XML
- SYMFONY

O formato Liquid é um formato de template muito simples. A sua sintaxe é muito parecida com outros tipos de padrões de templates. Você cria a sua página HTML usando variáveis que depois serão substituídas por textos obtidos de alguma base de dados.

O Jekyll foi criado em parceria com o GitHub, para ser usado com a ferramenta GitHub Pages, que serve para hospedar um site usando os próprios repositórios do GitHub. A ideia surgiu pois o GitHub Pages não tem suporte a banco de dados, somente HTML, CSS e JavaScript.

5.2 LINGUAGEM PYTHON

Python é uma linguagem de programação orientada a objetos criada por Guido Van Rossum durante o projeto “Amoeba”.

Percebi que o desenvolvimento de utilitários para administração de sistema em C (do Amoeba) estava tomando muito tempo. Além disso, fazê-los em shell Bourne não funcionaria por diversas razões. O motivo mais importante foi que, sendo um sistema distribuído de microkernel com um design novo e radical, as operações primitivas do Amoeba eram diferentes muito (além de serem mais refinadas) das operações primitivas disponíveis no shell Bourne. Portanto, havia necessidade de uma linguagem que ‘preencheria o vazio entre C e o shell’. Por um tempo longo, esse foi o principal objetivo do Python. (VAN ROSSUM, Guido)

A linguagem Python é muito simples e versátil de se utilizar, mas também é muito poderosa. Devido a sua simplicidade, o processamento de arquivos de texto é muito mais fácil de se programar.

5.3 LINGUAGENS DE TEMPLATE

São linguagens que são utilizadas para processar texto e substituir palavras chaves por dados pré-definidos. São muito utilizadas em aplicações que usam o sistema MVC (Model View Controller), onde os templates são escritos somente na camada View.

Essas linguagens são processadas por um compilador específico de cada linguagem, transformando as variáveis e os processamentos em texto pronto, a partir de dados que já foram processados nas camadas anteriores da aplicação.

Segue um exemplo bem comum em uma aplicação MVC:

1. A camada Model obtém e prepara as informações que vêm do banco de dados.
2. A camada Controller recebe as informações que vêm da camada Model e faz algum tipo de ação, por exemplo, exibir os dados na tela do usuário.
3. O Controller manda os dados específicos em variáveis para a camada View.
4. A camada View recebe os dados e somente exibe-os, podendo ter alguns tipo de processamento, como laços de repetição.

Alguns exemplos dessas linguagens são:

- Liquid;
- Mustache;
- Blade;
- Handlebars.

5.1 LINGUAGENS DE NOTAÇÃO DE OBJETOS

São linguagens de transferência de dados com o objetivo de serem fáceis e

amigáveis de usar e simples de processar. Elas tem uma estrutura simples para que possam ser utilizadas a partir de várias linguagens diferentes.

Dentre todas, a mais utilizada atualmente é o JSON (JavaScript Object Notation), que é oriunda do JavaScript mas que já é usada em muitas linguagens pela semelhança com a notação de objetos de outras linguagens e por isso tem uma fácil conversão.

Outra linguagem de notação muito utilizada é a conhecida como arquivo INI, que é utilizada no Windows e é muito mais simples que as outras, em relação aos tipos básicos suportados.

Atualmente vem crescendo o uso dessas linguagens, e uma em ascensão é a linguagem YAML (Ain't Markup Language), cujo objetivo é tornar a escrita dos dados mais amigável possível, sendo também convertida para diversas linguagens.

6. MATERIAIS E MÉTODO

A ênfase deste capítulo está em reportar o que e como será feito para alcançar o objetivo do trabalho.

6.1 MATERIAIS

Materiais são as ferramentas, as tecnologias, os ambientes de desenvolvimento e outros que são utilizados para realizar as atividades desde a definição dos requisitos à implantação do sistema.

6.1.1 Linguagem Python

A linguagem Python foi escolhida pela sua facilidade em lidar com processamento de arquivos de texto, tanto na leitura quanto na escrita.

6.1.2 Linguagem e biblioteca YAML

A linguagem YAML foi escolhida por ser muito fácil de compreender e usar para pessoas que não estão familiarizadas com linguagens de programação.

6.1.3 Biblioteca Pystache

Pystache é uma versão para Python da linguagem de template Mustache, que será usada para os templates locais.

6.1.4 Linguagens HTML, CSS e JavaScript

São as linguagens utilizadas na programação voltada para o mercado de sites (programação web), sendo o HTML a linguagem de marcação estática que

delimita toda a formatação geral em uma página web, o CSS uma linguagem de estilização que define os estilos e outros efeitos que uma página web pode executar e o JavaScript é a linguagem de programação orientada a objetos voltada especificamente para a manipulação da página em tempo de execução.

6.1.5 Bootstrap

Bootstrap é um framework de CSS e JavaScript que facilita a personalização da página web utilizando menos código.

6.1.6 Editor Atom

Atom é uma ferramenta CASE IDE, destinada a editar os códigos-fonte do projeto.

6.1.7 Git

Git é uma ferramenta de VCS (ou SCM) utilizada para organizar e manter os códigos-fonte entre outras coisas seguras utilizando backups manuais.

6.1.8 Linguagens Batch e Shell

São linguagens de script utilizadas em rotinas no prompt de comando do sistema operacional. Batch é utilizada no Windows e Shell no Linux.

6.1.9 Astah Community

Astah é uma ferramenta CASE destinada à criação de diagramas de diversos tipos, como diagrama de classes, caso de uso, atividade, entidade-relacionamento entre outros.

6.1.10 LibreOffice

LibreOffice é uma ferramenta de escritório com utilitários para edição de

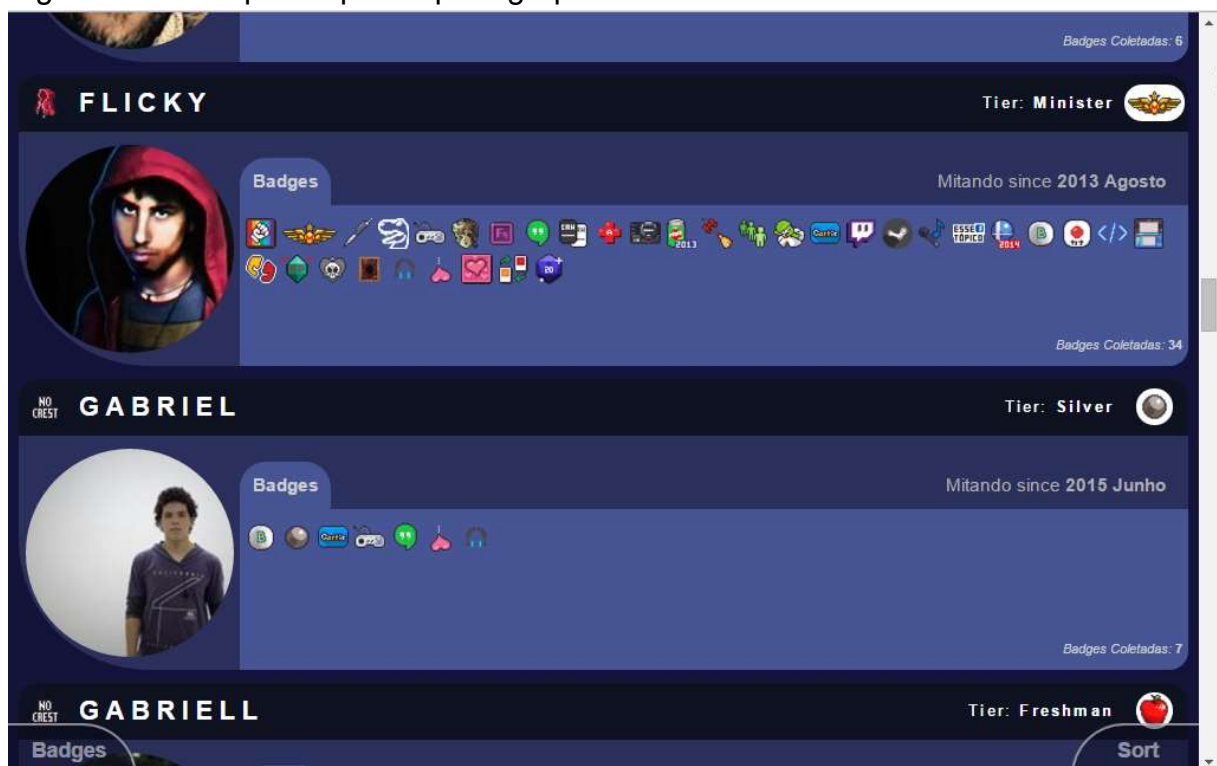
texto WYSIWYG, edição de planilhas, edição de formas e figuras, base de dados entre outros.

6.2 MÉTODO

6.2.1 Levantamento de requisitos

Nesta etapa os requisitos foram obtidos a partir de um projeto de página web já implementado, que seria um painel listando os membros do grupo e suas medalhas, como na imagem abaixo.

Figura 1 – Exemplo de painel para grupos



Fonte: <https://osmitadores.github.io> (2015)

Os requisitos básicos para cada integrante do painel são:

- Nome

- Medalhas
- Insígnia
- Nível

Outros requisitos poderão ser adicionados a partir de uma personalização manual.

A ideia do projeto é criar uma ferramenta que possa ser utilizada por usuários com pouco conhecimento em programação, sendo necessário somente o conhecimento de programação web.

Existe a ideia de se criar uma aplicação em PHP para facilitar ainda mais para que possa ser utilizado por usuários sem conhecimento de programação mas foi descartada neste projeto pois o objetivo é gerar uma página estática capaz de ser hospedada em qualquer servidor, sem depender de um banco de dados ou qualquer outra transferência que não seja das imagens que serão usadas no painel.

A linguagem escolhida nesta etapa foi Python pois ela tem uma facilidade maior de processar arquivos de texto. A versão escolhida foi a 2.7 pois é a mais compatível com os sistemas operacionais Linux, que já possuem esta versão instalada. O único problema em usar a versão 2.7 é que ela possui um suporte reduzido a caracteres unicode, que são os caracteres de idiomas com algum tipo de acentuação ou caracteres especiais.

Nesta etapa também foi escolhida a linguagem YAML para fazer a parte de banco de dados local pois é facilmente utilizada sem ter muito conhecimento de linguagem de notação de objetos e foi escolhida a linguagem de template Mustache (que a biblioteca Python se chama PyStache) para ser utilizada nos templates HTML.

6.2.2 Método de desenvolvimento

O método escolhido para a execução do projeto foi a metodologia ágil Extreme Programming (ou XP). Isso porque o projeto sofre uma série de alterações pois necessita de testes constantes.

Este método mantém um ciclo de desenvolvimento focado em lançamentos frequentes, visando novos recursos e correção de falhas mais rapidamente.

Figura 2 – Método de desenvolvimento Extreme Programming

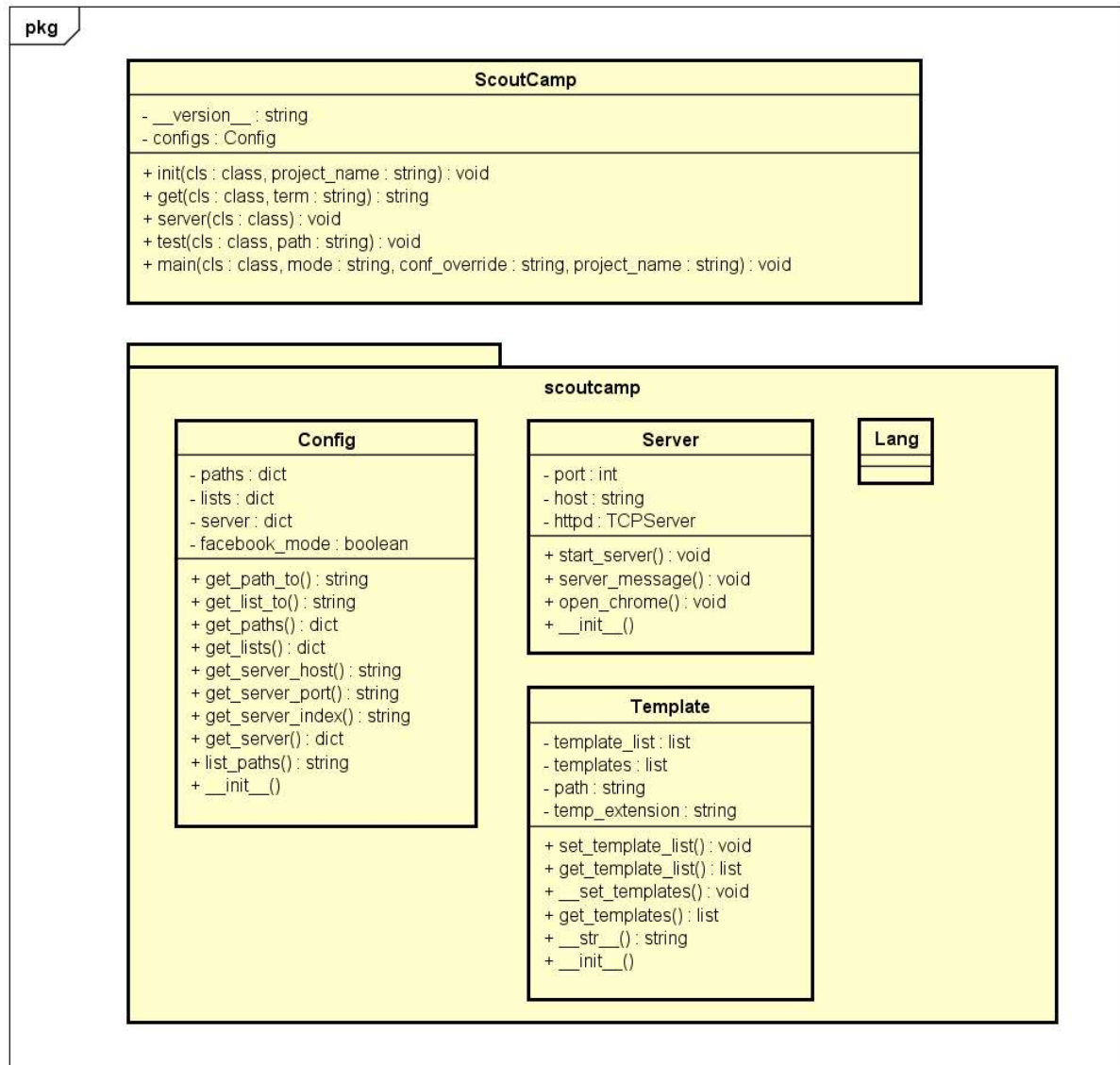


Fonte: <http://www.itnerante.com.br/> (2013)

6.2.3 Modelagem do sistema

Foi feita uma base geral do projeto com um diagrama de classes inicial como na imagem abaixo, e durante os testes outros diagramas foram sendo feitos em paralelo.

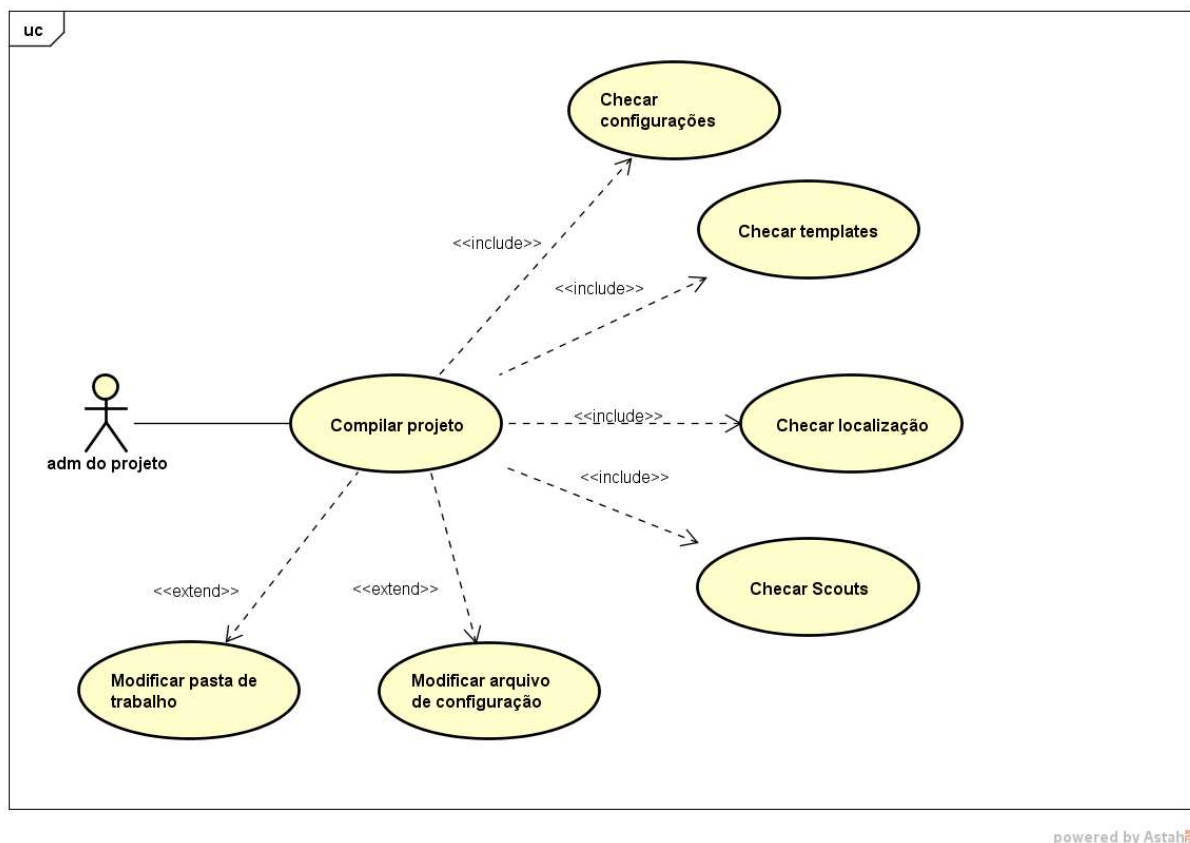
Figura 3 - Diagrama de classes



Fonte: Próprio autor (2016)

Em seguida foram feitos alguns diagramas de caso de uso:

Figura 4 - Diagrama de caso de uso (Render)



Fonte: Próprio autor (2016)

Nome do caso de uso: Render

Ator principal: Administrador do projeto

Resumo: Descreve o comando “render” do programa, que servirá para transformar os dados em HTML estático.

Pré-condições:

- O administrador deverá ter, pelo menos, um arquivo de configuração
- As pastas devem estar organizadas conforme o arquivo de configuração

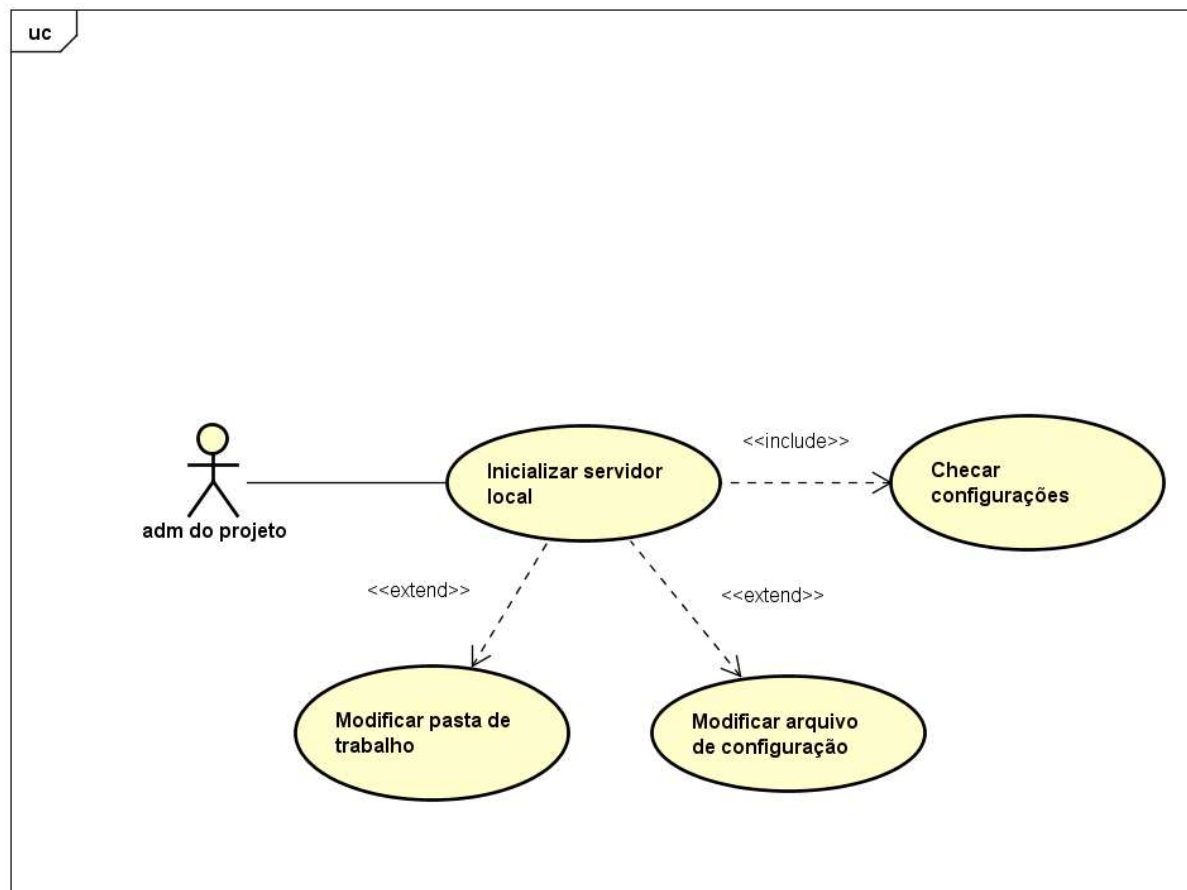
Pós-condições:

- O resultado será salvo na pasta especificada no arquivo de configuração

Fluxo principal:

1. Checar o arquivo de configuração.
2. Checar os templates.
3. Checar as strings de localização.
4. Checar os membros.
5. Processar as informações para o HTML

Figura 5 - Diagrama de caso de uso (Server)



powered by Astah

Fonte: Próprio autor (2016)

Nome do caso de uso: Server

Ator principal: Administrador do projeto

Resumo: Descreve o comando “server” do programa, que servirá para inicializar o um servidor local para testes.

Pré-condições:

- O administrador deverá ter, pelo menos, um arquivo de configuração
- As pastas devem estar organizadas conforme o arquivo de configuração

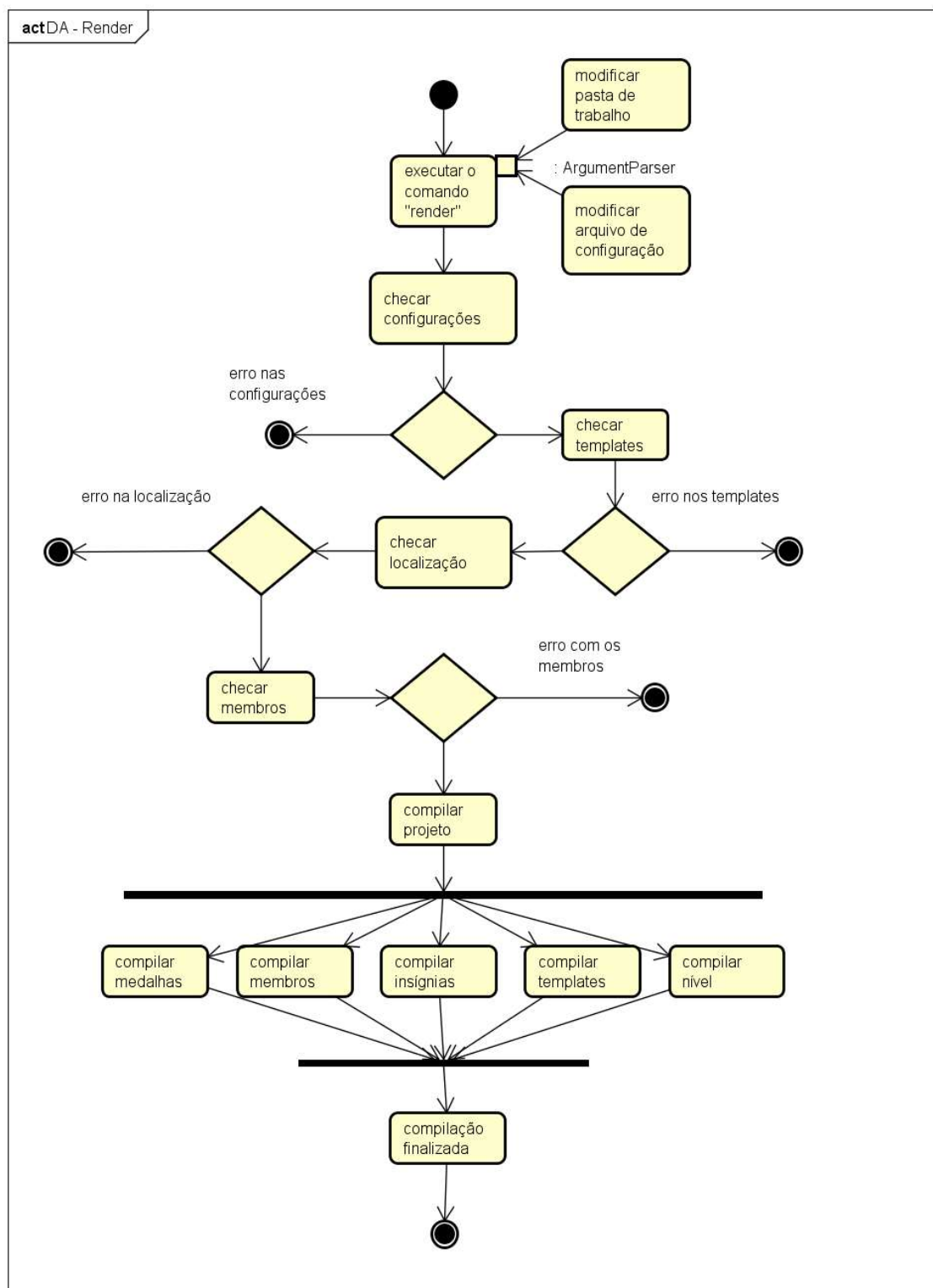
Pós-condições:

- Será iniciado um servidor local e o navegador será aberto

Fluxo principal:

1. Checar o arquivo de configuração.
2. Iniciar servidor.
3. Abrir navegador.

E também um diagrama de atividade mostrando o processo de como o programa funciona no seu comando principal:



powered by Astah

Fonte: Próprio autor (2016)

6.2.4 Funcionamento do programa

O programa não possui uma interface gráfica, logo ele é executado por CLI.

Figura 7 – Programa sendo executado em CLI

```

MINGW64; c:/Users/tumeo/Documents/GitHub/scout-camp (tcc)/TCC
tumeo@NWSEA-PC MINGW64 ~/Documents/GitHub/scout-camp (tcc)/TCC (tcc)
$ scout
usage: ScoutCamp [-h] [-r] [-p PATH] [-c CREATE] [-t TEST] [-s] [-d] [-v]

Scout Camp - Static HTML Group Manager

optional arguments:
  -h, --help            show this help message and exit
  -r, --render           compile project using default config file
  -p PATH, --path PATH  compile using alternative path
  -c CREATE, --create CREATE
                        create new ScoutCamp project
  -t TEST, --test TEST  compile using another config file
  -s, --server           start the Scout Camp server
  -d, --data            generate SQLite database
  -v, --version         show version

tumeo@NWSEA-PC MINGW64 ~/Documents/GitHub/scout-camp (tcc)/TCC (tcc)
$ |

```

Fonte: Próprio autor (2016)

O ciclo de vida do programa é único, não existe uma interação dentro do programa, sendo ele só um processador de dados, como um compilador.

Para criar um projeto, basta executar o comando específico seguido do nome do projeto e então um template padrão será criado.

Então é preciso editar os dados manualmente pelos arquivos locais, adicionando membros, criando as medalhas e vinculando aos membros.

Depois basta executar o comando específico para compilar o projeto e todos os dados serão transformados em uma página HTML pronta. Para testar o resultado é só criar um servidor local a partir de um programa da sua preferência ou simplesmente usar o comando específico que já cria o servidor para testes.

Todos os dados do projeto serão armazenados em arquivos YAML e os templates serão feitos com HTML e a linguagem Mustache.

Os dados podem ser exportados para arquivos JSON ou embutidos nos scripts por meio de template.

Um recurso de exportar os dados para banco de dados SQLite foi criado mas ainda não será utilizado no resultado final.

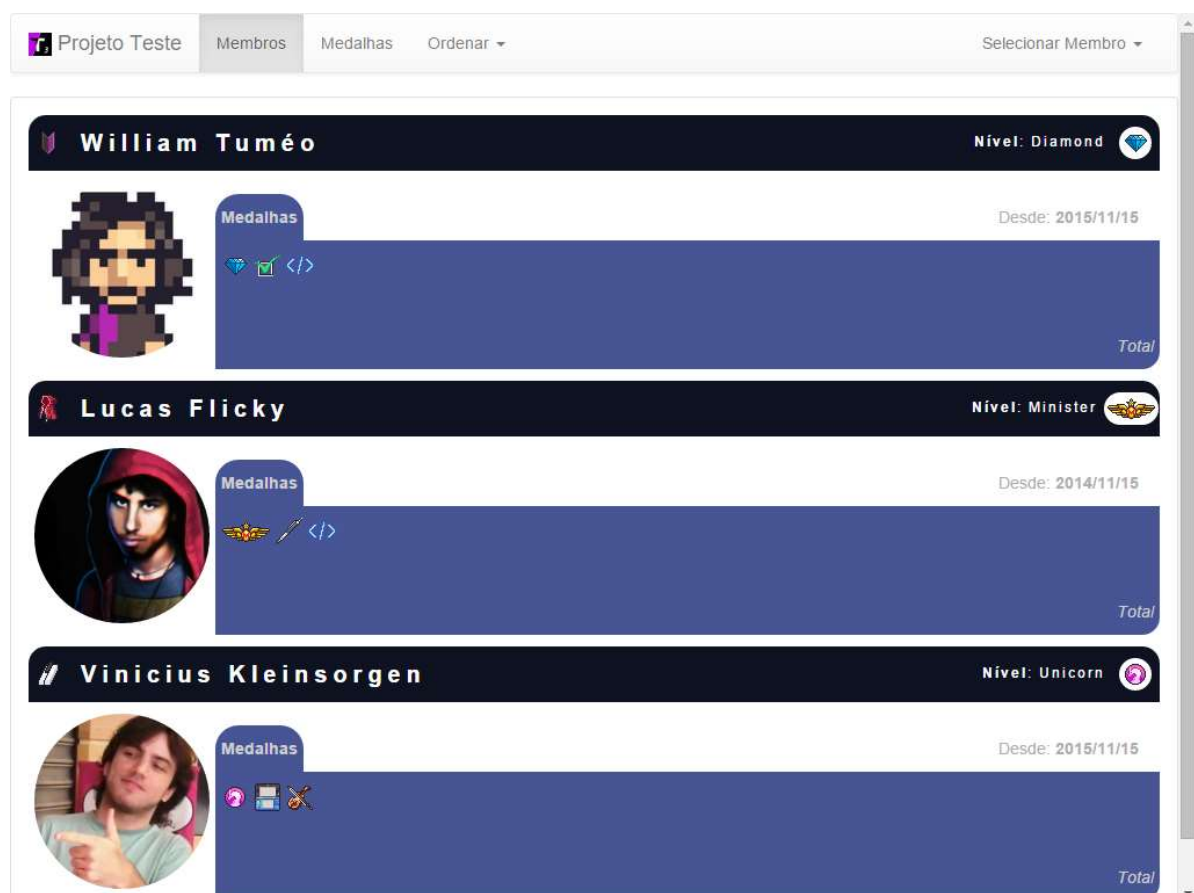
7. RESULTADOS E DISCUSSÕES

Este capítulo apresenta o que foi obtido como resultado do trabalho, como foi testado e sobre o funcionamento do projeto pronto.

7.1 IMPLEMENTAÇÃO DO SISTEMA

O sistema foi implementado utilizando alguns membros já existentes na página que já estava pronta e algumas modificações no estilo da página, utilizando alguns componentes do Bootstrap.

Figura 8 - Teste usando o programa pronto



Fonte: Próprio autor (2016)

Utilizando a organização pré-definida no sistema de arquivos, foi feita a implementação sem nenhuma dificuldade, organizando separadamente os templates, tabela do membro, membros, medalhas, scripts e outros recursos diversos.

7.2 DADOS OBTIDOS E DISCUSSÕES

A princípio, o sistema ficou relativamente fechado, podendo detalhar somente membros e medalhas, mas a ideia inicial seria algo mais versátil, para não ter essa limitação específica e detalhar separadamente vários outros tipos de coisas.

A organização do projeto ainda permite essa alteração posteriormente sem muita dificuldade.

Trabalhar com a linguagem Python foi muito prazeroso, de forma que foi bem fácil e prático conselhar com a linguagem YAML e os templates em Mustache, o único problema em utilizar a versão 2.7, como já citado, é a incompatibilidade com caracteres unicode, que foi um grande empecilho durante o desenvolvimento do projeto.

Fora esses problemas, o projeto foi concluído como o esperado, tendo as funções operando normalmente, com exceção a ferramenta de exportar para um banco de dados SQLite, que foi deixada de lado até então.

Os templates em Mustache funcionaram como o esperado, podendo listar todos os membros e suas medalhas na própria execução do programa, sem precisar preencher as tabelas utilizando as variáveis que foram exportadas em JSON.

8. CONCLUSÃO

Levando em conta todos os aspectos do projeto citados, o projeto foi muito bem-sucedido em facilitar algo que já estava estabelecido. Utilizar templates proporcionou uma facilidade muito grande no desenvolvimento da página dos membros. E como o projeto foi bem estruturado, será possível adicionar muitos outros recursos sem ter que alterar muito do que já está pronto.

9. REFERÊNCIAS

Servindo sites estáticos com Jekyll. Citado em 8 de Novembro de 2015. Disponível em: <http://tableless.com.br/jekyll-servindo-sites-estaticos/>

A História do Python. Citado em 23 de Novembro de 2015. Disponível em: <http://mindbending.org/pt/a-historia-do-python>

“Usar engine de template ou trabalhar com PHP puro?”. Citado em 6 de Dezembro de 2015. Disponível em: <http://pt.stackoverflow.com/questions/16462/usar-engine-de-template-ou-trabalhar-com-php-puro>

Introdução ao Padrão MVC. Citado em 16 de Dezembro de 2015. Disponível em: <http://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>