

Projeto de software para o gerenciamento do Planejamento Participativo da UTFPR-PB

Resumo

O projeto visa gerenciar e exibir informações sobre o Planejamento Participativo da UTFPR, campus Pato Branco. O gerenciamento das informações é realizado utilizando os softwares OpenProject e Google Planilhas. Os dados são armazenados em um banco de dados relacional PostgreSQL, hospedado localmente na instituição. O projeto também inclui um website próprio para a geração e visualização de relatórios e dashboards.

Arquitetura

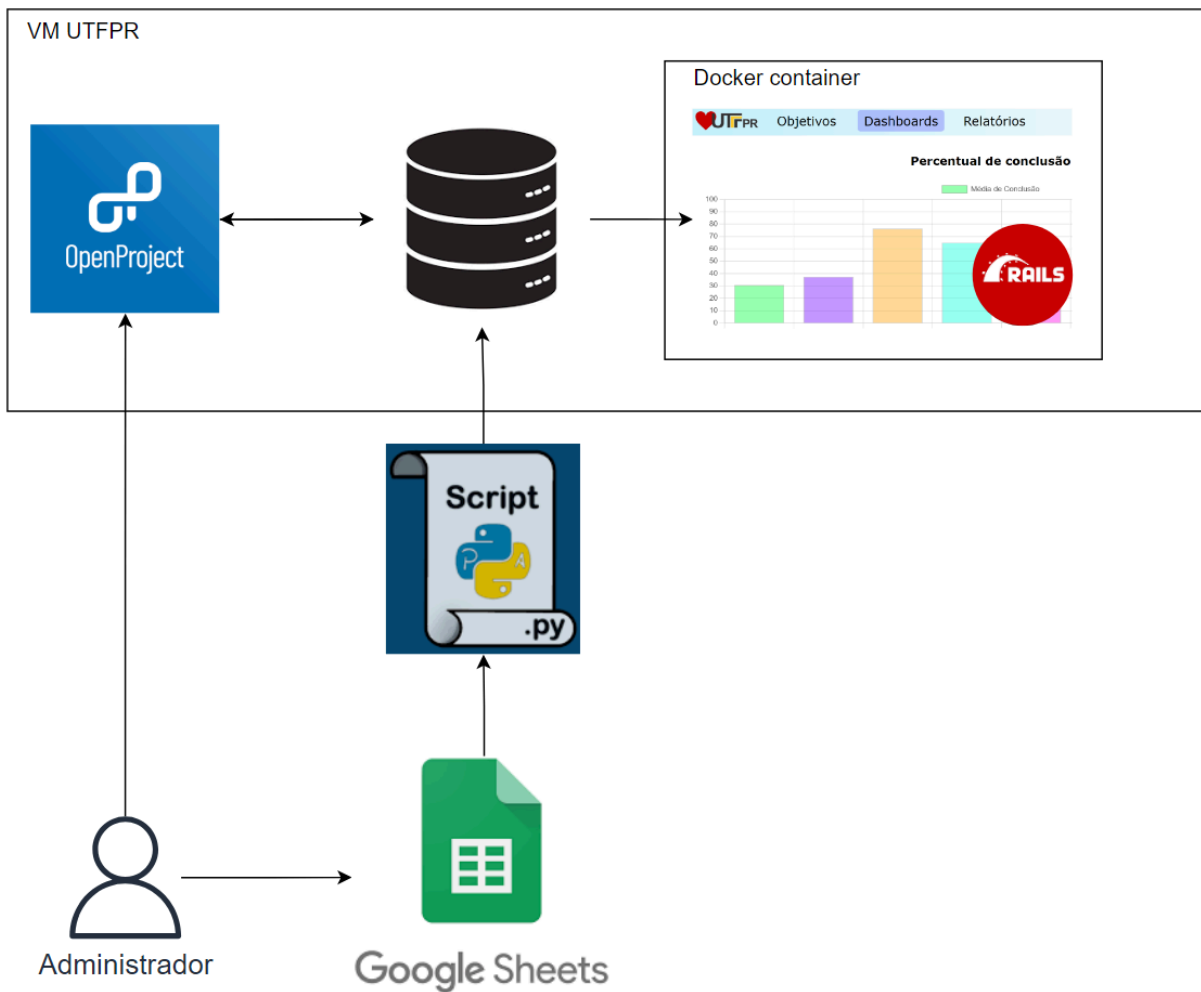
A arquitetura do projeto consiste em adicionar as informações do OpenProject e/ou Google Planilhas no banco de dados e, posteriormente, exibi-las em formato de relatórios e dashboards.

O fluxo de funcionamento segue a seguinte sequência:

1. O administrador do sistema insere as informações em uma planilha do Google Planilhas;
2. O script em Python insere as informações da planilha no banco de dados PostgreSQL;
3. O OpenProject sincroniza as informações com o banco de dados;
4. A aplicação Ruby on Rails coleta e filtra as informações do banco de dados e exibe seus relatórios e dashboards;
5. O container Docker hospeda a aplicação Rails no endereço <http://172.29.150.197:3000/>

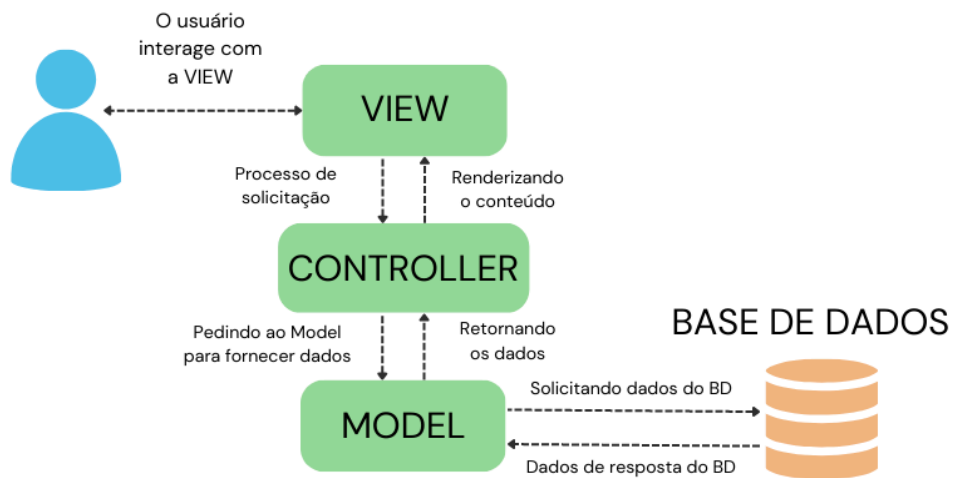
O administrador também pode gerenciar as informações diretamente pelo OpenProject através de <http://172.29.150.197/openproject/>

Diagrama da arquitetura do projeto:



Aplicação Ruby on Rails (RoR)

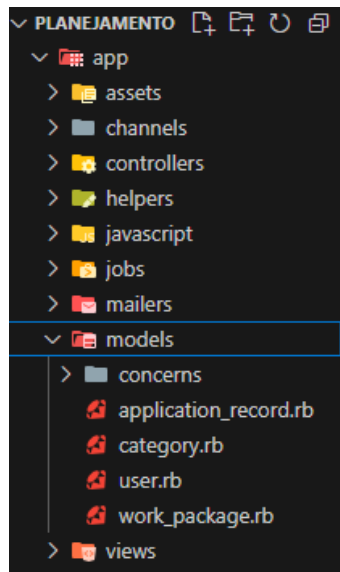
O Ruby on Rails segue uma arquitetura MVC (Model, View e Controller)



(1)

Toda vez que a planilha entra em uma nova versão, é rodado o script em Python para inserção dos dados no BD, assim, a coluna *project_id* da tabela *work_packages* é atualizada.

A aplicação Rails irá filtrar apenas os objetivos e ações do *project_id* desejado. Isso é feito pelo Model *work_package.rb* do Rails:



Nas linhas 13 e 14 definir o “project_id: ” atual:

```
work_package.rb app\models\work_package.rb\...
1 class WorkPackage < ApplicationRecord
2   self.table_name = 'work_packages'
3   self.primary_key = 'id'
4
5   # Associação para responsável
6   belongs_to :responsible_user, class_name: 'User', foreign_key: 'responsible_id', optional: true
7   # Associação para as tuplas filhas
8   has_many :children, class_name: "WorkPackage", foreign_key: "parent_id"
9
10  # Associação para categorias:
11  belongs_to :category, class_name: 'Category', foreign_key: 'category_id', optional: true
12
13  # Definindo o escopo padrão para filtrar por project_id
14  default_scope { where(project_id: 8).where(type_id: [8, 9]) }
15 end
16
```

Todas as relações de usuários são extraídas dos perfis do OpenProject e já estão corretamente configuradas.

Guia instalação

Um guia de instalação e configuração de ambiente para trabalhar com o WebSite RoR

```
sudo apt-get update
sudo apt-get install git curl build-essential zlib1g-dev libyaml-dev libssl-dev
libpq-dev libreadline-dev

# Install rbenv locally for the dev user
git clone https://github.com/rbenv/rbenv.git ~/.rbenv
# Optional: Compile bash extensions
cd ~/.rbenv && src/configure && make -C src
```

```
# Add rbenv to the shell's $PATH.
echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc

# Run rbenv-init and follow the instructions to initialize rbenv on any shell
~/.rbenv/bin/rbenv init
# Issue the recommended command from the stdout of the last command
echo 'eval "$(rbenv init - bash)"' >> ~/.bashrc
# Source bashrc
source ~/.bashrc

git clone https://github.com/rbenv/ruby-build.git ~/.rbenv/plugins/ruby-build

# Install the required version as read from the Gemfile
rbenv install 3.3.3

rbenv global 3.3.3
rbenv rehash

sudo apt-get install postgresql postgresql-client
sudo su - postgres
[postgres@ubuntu]# createuser -d -P openproject

[postgres@ubuntu]# createdb -O openproject openproject_dev
[postgres@ubuntu]# createdb -O openproject openproject_test

# Exit the shell as postgres
[postgres@ubuntu]# exit
```

Verifique a instalação com:

```
ruby -version

bundler -version
```

Conexão Rails e PostgreSQL

É necessário estar conectado na rede local da UTFPR para desenvolver o projeto Rails, já que o Model se comunica diretamente com o BD hospedado na VM. Para desenvolvimento remoto, utilizar VPN ou realizar uma cópia local do banco de dados.

Docker

Para rodar a aplicação Rails na VM da UTFPR-PB, utiliza-se um container Docker que está no Docker Hub.

OBS: O Rails gera uma chave secreta de segurança para a aplicação, para visualizar a chave rode no terminal da aplicação o comando `rails secret`.

Para rodar o container na VM faz-se:

```
docker pull thassiamorim/planejamento  
docker run -p 3000:3000 -d -e SECRET_KEY_BASE=<chave gerada1> thassiamorim/planejamento
```

Referências

- (1) JOAO. O que é MVC? Entenda arquitetura de padrão MVC. Disponível em:
<https://www.usandopy.com/pt/artigo/o-que-e-mvc-entenda-arquitetura-de-padrao-mvc>

¹ Chave gerada pelo comando `rails secret` no terminal de desenvolvimento da aplicação Rails (Não na VM)