PETRVS
(Cooperação ANTAQ - PRF)

Objetivo

A Plataforma Eletrônica de Trabalho e Visão Sistêmica (PETRVS) é uma solução que visa permitir o gerenciamento de equipes e de processos no setor público de maneira inovadora, contando ainda com ferramenta de apoio a gestão de projetos e possibilidade de integrações via API com diversos serviços e sistemas.

Arquitetura

Por se tratar de uma aplicação web o sistema está dividido em ***front-end*** (*client-side*, lado cliente, Interface com o usuário) e ***back-end*** (*server-side*, lado servidor, API servidora). A comunicação entre o *front-end* e o *back-end* é realizada através de uma API REST fornecida pelo back-end (que possibilita futuramente a sua utilização por Aplicativos Mobile ou integrações B2B). Toda comunicação é feita utilizando o encapsulamento JSON, e é projetado para trabalhar sobre canal seguro SSL/TLS (requisições https, com a definição de políticas CORS, *Cross-origin resource sharing*).

Front-end

O lado cliente é fornecido de duas maneiras distintas, como **extensão do browser** e como **SPA** (*Single Page Application*):

- Extensão Chrome ou Mozilla Firefox: Neste modelo a aplicação é *embedded* (incorporada, injetada) dentro de outra aplicação hospedeira, que inicialmente foi configurada para o sistema Sei (Mas permite que seja incorporado a outros sistemas, caso seja necessário). Serão fornecidos todos os recursos da aplicação, mas em uma versão compactada em um contêiner dentro da tela do hospedeiro. Fornecerá ainda funcionalidades que estendem o próprio hospedeiro (no caso o sistema Sei), incorporando facilidades, como por editores mais elaborados (vantagem essa da utilização da versão como extensão ao invés da versão SPA);
- SPA: versão fornecida diretamente através de uma DNS (endereço eletrônico, exemplo: petrvs.antaq.gov.br). Aplicação independente, que permite melhor usabilidade do usuário, pois não estará mais limitado a um container do hospedeiro. O modelo SPA permite que a aplicação que contém a UI/UX (interface com o usuário) seja desenvolvida e carregada pelo browser como uma aplicação única, de modo que não é necessário ficar recarregando a página para carregar uma nova tela (melhora de performance).

Framework front-end

O front-end é desenvolvido utilizando o Angular, framework em TypeScript/JavaScript da **Google**, que permite criar uma aplicação SPA modularizada, de modo que a aplicação é carregada no browser do usuário por demanda, melhorando a performance, e evitando a necessidade de recarregar páginas e diminuindo o tráfego com o servidor. Todo o desenvolvimento com o Angular é modularizado e feito utilizando o *Design Pattern* MVCS (Model, View, Controller and Services), permitindo que funcionalidades possam ser criadas sem influenciarem as já existentes, de modo que o desenvolvimento possa ser desenvolvido **multiagência** sem maiores problemas. Além do Angular é utilizado também a biblioteca de Bootstrap, que permite a criação de aplicações **responsivas** (que funcionam tanto em computador quanto em dispositivos móveis como tablets ou celulares), além de fornecer um padrão de interface com o usuário e diversos componentes de iteração.

A utilização do Angular também permite que as informações renderizadas em tela sejam realizadas por demanda, evitando delays e processamentos desnecessários. Somente quando demandado pelo usuário, mas de forma automática, os dados são renderizados e disponibilizados ao usuário, gerando um ganho de desempenho considerável.

Back-end

O lado servidor é desenvolvido utilizando a linguagem PHP e o banco de dados MySQL (mas está estruturado para trabalhar com outros banco de dados, caso necessário, como o MS SQL Server, PostgreSQL e Oracle). O servidor tanto fornecerá acesso a API quanto será o host da aplicação SPA (todo o deploy da aplicação é feito em um único local). O designer pattern adotado também é o MVCS.

Framework back-end

Foi adotado o Laravel para desenvolvimento do back-end devido ao fato de ser um framework maduro e considerado o mais utilizado atualmente, além de fornecer diversos recursos e auxiliar no desenvolvimento da aplicação, além de estar alinhado com os Design Patterns e Project Patterns recomendados para desenvolvimento de sistemas.

Os principais recursos e funcionalidades do back-end são:

- Laravel Migrations: Toda a criação e modificação do banco de dados é feito de forma versionalizada, permitindo que o sistema seja executado em diversos servidores sem problema de compatibilidades. Além de permitir a migração de um banco de dados para outro, quando necessário;
- Laravel Sanctum: Mecanismos de autenticação e validação de usuários que permitiu a implementação de login com **G-Suíte** (contas gmail ou institucionais), login autônomo, login interno da instituição e futuramente a autenticação **Microsoft** e o SSO **Gov.br**;
- Tratamento de políticas de CORS e fornecendo acesso a API somente para usuários autenticados, com controle de perfis de usuários;

- Fornecendo ao mesmo tempo a API e o acesso a aplicação SPA, de modo que tanto os usuários da extensão quanto os que acessam diretamente a aplicação tenham acesso a mesma aplicação (resolvendo problemas com novas versões, que serão imediatamente acessadas pelos usuários, sem a necessidade de atualizar a extensão ou qualquer outro processo);
- **Garantia de integridade** das informações, sendo utilizado *transactions* para gravar os dados no servidor, de modo que todas as informações sejam gravadas, ou caso um falhe, todas retornem ao estado inicial, o usuário seja informado e a integridade seja sempre mantida.
- Criação de índices no banco de dados para garantir a **performance**;
- Campos chaves (local ou estrangeira) das tabelas em formato de UUID, para permitir a distribuição de informações em vários bancos de dados ou futura unificação.
- Utilização do ORM Eloquent para manipulação do banco de dados como objetos, evitando invasões do tipo SQL-Injection;

Modelo Entidade Relacionamento

O modelo MER apresentado é baseado no modelo adotado pela extensão em sua versão anterior. De modo que modificações são realizadas à medida que as funcionalidades são implementadas, e podem não refletir exatamente o resultado final.

O MER está presente no Anexo 01.

Cronograma

O cronograma do projeto está dividido em Sprints, segundo o modelo Scrum de desenvolvimento.

Sprint 1 (01/10/2021 - 31/12/2021)

- Criação do projeto Angular;
- Criação do projeto Laravel;
- Criação de classes base DAO, Model, Controller e View para Angular;
- Criação de classes base DAO, Model, Controller para Laravel;
- Implementação da autenticação com G-Suíte, Microsoft, Gov.br;
- Criação da migration (banco de dados);
- Configurações de segurança e acesso a rotas;
- Criação da tela de Login;
- Criação da tela Home;
- Módulos de cadastro:
    - Plano de trabalho;
    - Programas de Gestão;
    - Tarefas
    - Tipos de Documentos;
    - Tipos de Justificativas;
    - Tipos de Modalidades;

    - Tipos de Requisições;
- Módulo Atividades;
- Afastamentos;
- Relatórios.

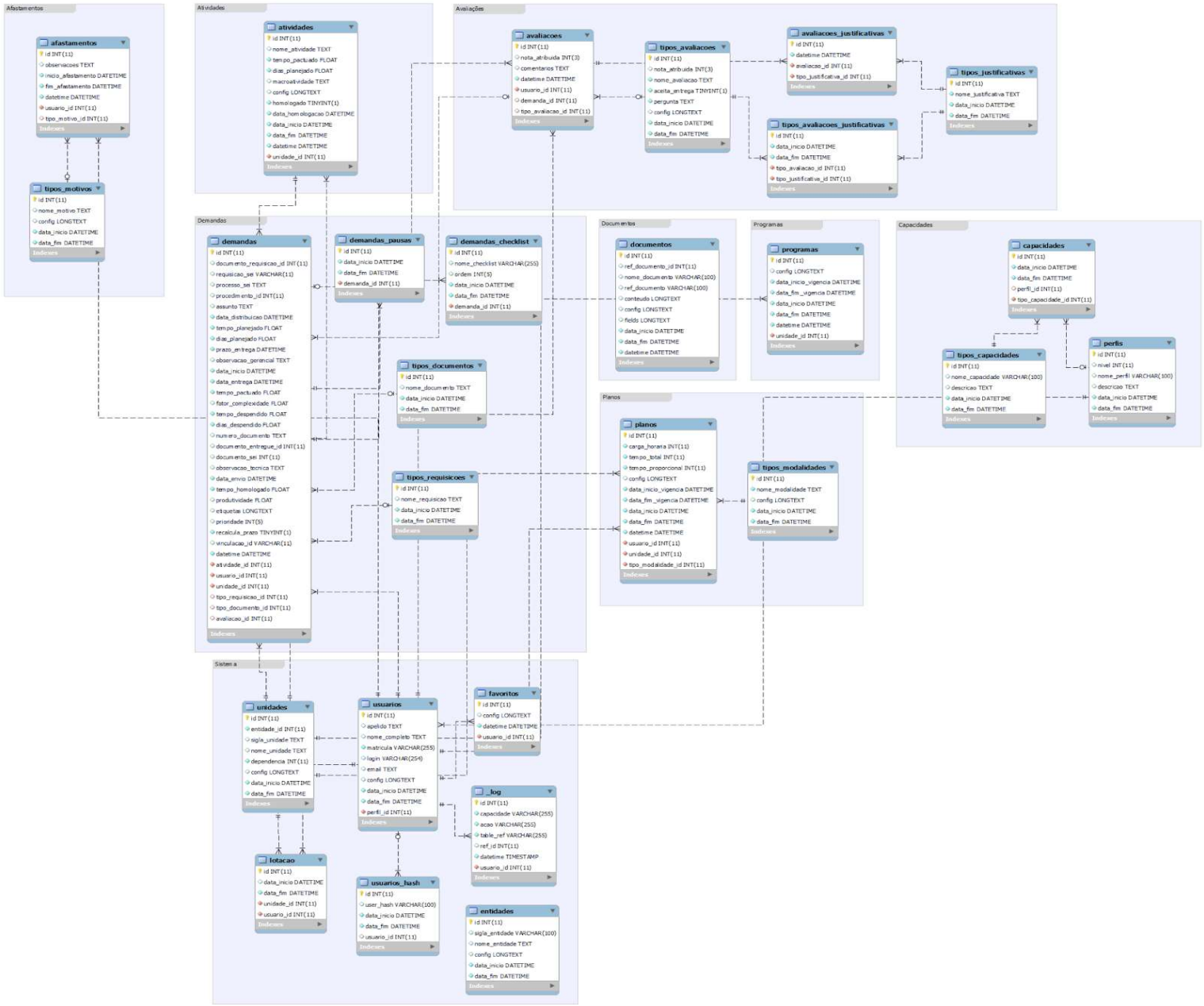
Repositório GIT

Todo o sistema está sendo mantido em um repositório GIT único que pode ser acessado por todas as agências envolvidas no projeto. E será ponto único, mesmo que sejam desenvolvidas funcionalidades específicas para cada órgão, ficarão disponíveis para as demais.

Considerações

O novo sistema está sendo inteiramente desenvolvido para resolver problemas de performance da versão anterior, que atualmente inviabiliza o uso com grande volume de informações, e de forma estruturada para permitir que multiagências possam trabalhar no mesmo projeto, justamente por ser desenvolvido utilizando frameworks conhecidos, onde inclusive empresas terceirizadas possam dar continuidade facilmente.

Anexo 01 - DER


Afastamentos
afastamentos
id INT(11)
observacoes TEXT
inicio_afastamento DATETIME
fim_afastamento DATETIME
datetime DATETIME
usuario_id INT(11)
tipo_motivo_id INT(11)
Indexes
tipos_motivos
id INT(11)
nome_motivo TEXT
config LONGTEXT
data_inicio DATETIME
data_fim DATETIME
Indexes
Atividades
atividades
id INT(11)
nome_atividade TEXT
tempo_pactuado FLOAT
dias_planejado FLOAT
macroatividade TEXT
config LONGTEXT
homologado TINYINT(1)
data_homologacao DATETIME
data_inicio DATETIME
data_fim DATETIME
datetime DATETIME
unidade_id INT(11)
Indexes
Avaliações
avaliacoes
id INT(11)
nota_atribuida INT(3)
comentarios TEXT
datetime DATETIME
usuario_id INT(11)
demanda_id INT(11)
tipo_avaliacao_id INT(11)
Indexes
tipos_avaliacoes
id INT(11)
nota_atribuida INT(3)
nome_avaliacao TEXT
aceita_entrega TINYINT(1)
pergunta TEXT
config LONGTEXT
data_inicio DATETIME
data_fim DATETIME
Indexes
avaliacoes_justificativas
id INT(11)
datetime DATETIME
avaliacao_id INT(11)
tipo_justificativa_id INT(11)
Indexes
tipos_justificativas
id INT(11)
nome_justificativa TEXT
data_inicio DATETIME
data_fim DATETIME
Indexes
tipos_avaliacoes_justificativas
id INT(11)
data_inicio DATETIME
data_fim DATETIME
tipo_avaliacao_id INT(11)
tipo_justificativa_id INT(11)
Indexes
Demandas
demandas
id INT(11)
documento_requisicao_id INT(11)
requisicao_se VARCHAR(11)
processo_se TEXT
procedimento_id INT(11)
assunto TEXT
data_distribuicao DATETIME
tempo_planejado FLOAT
dias_planejado FLOAT
prazo_entrega DATETIME
observacao_gerencial TEXT
data_inicio DATETIME
data_entrega DATETIME
tempo_pactuado FLOAT
fator_complexidade FLOAT
tempo_despendido FLOAT
dias_despendido FLOAT
numero_documento TEXT
documento_entregue_id INT(11)
documento_sei INT(11)
observacao_tecnica TEXT
data_envio DATETIME
tempo_homologado FLOAT
produtividade FLOAT
etiquetas LONGTEXT
prioridade INT(5)
recalcula_prazo TINYINT(1)
vinculacao_id VARCHAR(11)
datetime DATETIME
atividade_id INT(11)
usuario_id INT(11)
unidade_id INT(11)
tipo_requisicao_id INT(11)
tipo_documento_id INT(11)
avaliacao_id INT(11)
Indexes
demandas_pausas
id INT(11)
data_inicio DATETIME
data_fim DATETIME
demanda_id INT(11)
Indexes
demandas_checklist
id INT(11)
nome_checklist VARCHAR(255)
ordem INT(5)
data_inicio DATETIME
data_fim DATETIME
demanda_id INT(11)
Indexes
tipos_documentos
id INT(11)
nome_documento TEXT
data_inicio DATETIME
data_fim DATETIME
Indexes
tipos_requisicoes
id INT(11)
nome_requisicao TEXT
data_inicio DATETIME
data_fim DATETIME
Indexes
Documentos
documentos
id INT(11)
ref_documento_id INT(11)
nome_documento VARCHAR(100)
ref_documento VARCHAR(100)
conteudo LONGTEXT
config LONGTEXT
fields LONGTEXT
data_inicio DATETIME
data_fim DATETIME
datetime DATETIME
Indexes
Programas
programas
id INT(11)
config LONGTEXT
data_inicio_vigencia DATETIME
data_fim_vigencia DATETIME
data_inicio DATETIME
data_fim DATETIME
datetime DATETIME
unidade_id INT(11)
Indexes
Capacidades
capacidades
id INT(11)
data_inicio DATETIME
data_fim DATETIME
perfil_id INT(11)
tipo_capacidade_id INT(11)
Indexes
tipos_capacidades
id INT(11)
nome_capacidade VARCHAR(100)
descricao TEXT
data_inicio DATETIME
data_fim DATETIME
Indexes
perfis
id INT(11)
nivel INT(11)
nome_perfil VARCHAR(100)
descricao TEXT
data_inicio DATETIME
data_fim DATETIME
Indexes
Planos
planos
id INT(11)
carga_horaria INT(11)
tempo_total INT(11)
tempo_proporcional INT(11)
config LONGTEXT
data_inicio_vigencia DATETIME
data_fim_vigencia DATETIME
data_inicio DATETIME
data_fim DATETIME
datetime DATETIME
usuario_id INT(11)
unidade_id INT(11)
tipo_modalidade_id INT(11)
Indexes
tipos_modalidades
id INT(11)
nome_modalidade TEXT
config LONGTEXT
data_inicio DATETIME
data_fim DATETIME
Indexes
Sistema
unidades
id INT(11)
entidade_id INT(11)
sigla_unidade TEXT
nome_unidade TEXT
dependencia INT(11)
config LONGTEXT
data_inicio DATETIME
data_fim DATETIME
Indexes
usuarios
id INT(11)
apelido TEXT
nome_completo TEXT
matricula VARCHAR(255)
login VARCHAR(254)
email TEXT
config LONGTEXT
data_inicio DATETIME
data_fim DATETIME
perfil_id INT(11)
Indexes
favoritos
id INT(11)
config LONGTEXT
datetime DATETIME
usuario_id INT(11)
Indexes
_log
id INT(11)
capacidade VARCHAR(255)
acao VARCHAR(255)
table_ref VARCHAR(255)
ref_id INT(11)
datetime TIMESTAMP
usuario_id INT(11)
Indexes
lotacao
id INT(11)
data_inicio DATETIME
data_fim DATETIME
unidade_id INT(11)
usuario_id INT(11)
Indexes
usuarios_hash
id INT(11)
user_hash VARCHAR(100)
data_inicio DATETIME
data_fim DATETIME
usuario_id INT(11)
Indexes
entidades
id INT(11)
sigla_entidade VARCHAR(100)
nome_entidade TEXT
config LONGTEXT
data_inicio DATETIME
data_fim DATETIME
Indexes