Tutorial de Instalação e Configuração

Instalações Necessárias

Antes de começar, verifique se você possui as seguintes ferramentas instaladas e as configurações realizadas em seu sistema:

- **Ferramenta de Desenvolvimento Recomendada:** É altamente recomendado utilizar o VSCode como sua ferramenta de desenvolvimento. Ela oferece um ambiente integrado e recursos que otimizam o fluxo de trabalho.
- **Ambiente Linux Sugerido:** Para uma experiência mais suave, sugerimos usar um ambiente Linux para o desenvolvimento. Isso se deve à integração e facilidade de instalação das bibliotecas utilizadas pela nossa arquitetura nesse sistema.
- **Configuração da VPN UFSC:** Certifique-se de configurar a VPN da UFSC antes de prosseguir. O acesso ao banco de dados de desenvolvimento é restrito e requer conexão à VPN.
- **Ferramentas de Gerenciamento de Banco de Dados:** Instale o PgAdmin 3 para facilitar o acesso ao PostgreSQL de desenvolvimento. Você também pode optar pelo dbVisualizer como alternativa.
- **Configuração do Servidor PostgreSQL:** No PgAdmin 3, configure uma nova conexão apontando para o servidor `150.162.67.43:5432`. Não é necessário fornecer senha para acessar o banco de dados de desenvolvimento. Certifique-se de selecionar o banco de dados `res_dev`.
- **Respeitando a Estrutura do BD:** Lembre-se de que não é permitido realizar alterações na estrutura do banco de dados por conta própria. Todas as solicitações de alterações devem ser encaminhadas ao DBA para avaliação e implementação.
- **NVM (Node Version Manager):** Utilize o NVM para gerenciar as versões do Node.js. É importante ter a versão 14.20 do Node.js para a maioria dos projetos. No entanto, observe que os projetos "stt-estomatologia" e "stt-espirometria" requerem a versão 18.17 do Node.js.
- **YARN:** O Yarn é um gerenciador de pacotes JavaScript criado pelo Facebook em colaboração com outras empresas de tecnologia. Ele foi projetado para ser uma alternativa ao npm (Node Package Manager), que é o gerenciador de pacotes padrão para projetos JavaScript e Node.js.
    1. Use o seguinte comando para instalar o Yarn globalmente:

       ```
       npm install -g yarn
       ```

- **Instalação das Ferramentas de Criptografia:** Instale o git-secret versão 0.5.0 e GPG versão 2.2.19 para lidar com informações sensíveis e criptografia.
- **Configuração de Chaves GPG:** Siga os passos abaixo para configurar as chaves GPG:
    1. Instale o GPG versão 2.2.19 usando o gerenciador de pacotes de sua distribuição Linux.
    2. Abra o terminal e execute os seguintes comandos para gerar um novo par de chaves GPG:

       ```
       gpg --gen-key
       ```

       Durante a configuração, você será solicitado a escolher o tipo de chave e o tamanho da chave. Recomendamos usar as opções padrão para a maioria dos casos.
    3. Após gerar a chave, liste suas chaves para encontrar o ID da chave recém-gerada:

```
gpg --list-keys
```

4. Exporte a chave pública usando o ID da chave (substitua `CHAVE_ID` pelo ID real da sua chave):

```
gpg --export -a CHAVE_ID > chave-publica.asc
```

5. Envie o arquivo `chave-publica.asc` para os endereços de e-mail de contato conforme necessário.

Certifique-se de adaptar os comandos de acordo com o seu sistema e suas preferências. Lembre-se de manter sua chave privada segura e compartilhar apenas a chave pública conforme necessário.

Certifique-se de adaptar as instruções conforme a configuração e as ferramentas específicas do seu ambiente.

Configuração STT-2

O módulo STT-2 é essencial para a funcionalidade deste tutorial. Siga os passos abaixo para configurá-lo:

- Clone o repositório do projeto STT-2 em sua máquina local:

```
git clone https://codigos.ufsc.br/incod/stt-2.git
```

- Acesse o diretório do projeto e faça o checkout na branch "dev" deste projeto:

```
cd stt-2
git checkout dev
```

- Instalar as depências básicas do STT-2 (lerna)

```
yarn install
```

Configuração do Módulo a Ser Trabalhado

Agora vamos configurar o módulo específico com o qual você irá trabalhar:

- A partir da pasta raiz do projeto STT-2, siga os passos a seguir:
    1. Clone o repositório do projeto stt-utilitarios como submódulo utilizando o comando:

```
git submodule update --init stt-utilitarios
```

    2. Acesse o diretório do projeto stt-utilitarios e faça o checkout na branch "dev" deste projeto

```
cd stt-utilitarios
git checkout dev
```

    3. Clone o repositório do projeto stt-componentes como submódulo utilizando o comando:

```
git submodule update --init stt-componentes
```

    4. Acesse o diretório do projeto stt-componentes e faça o checkout na branch "dev" deste projeto

```
cd stt-componentes
git checkout dev
```

- Para cada projeto que você pretende trabalhar, repita os 2 passos à seguir como submódulo, utilizando os comandos:
    1. Clone o repositório do projeto que você irá trabalhar como submódulo utilizando o comando:

       ```
       git submodule update --init stt-<MODULO>
       ```

    2. Acesse o diretório do projeto que você irá e faça o checkout na branch "dev" deste projeto

       ```
       cd stt-<MODULO>
       git checkout dev
       ```

- Caso você clone um projeto localmente e encontre arquivos com a extensão "*.secret", isso indica que você precisará usar o git-secret para descriptografar o conteúdo desses arquivos. A seguir, estão os passos necessários:
    1. Acesse o diretório do projeto no qual você irá descriptografar os arquivos:

       ```
       cd stt-<MODULO>
       ```

    2. Execute o comando a seguir para revelar o conteúdo dos arquivos criptografados:

       ```
       git-secret reveal -f
       ```

Obs.: Lembre-se de substituir `<MODULO>` pelo nome do módulo específico. Certifique-se de que as instruções reflitam corretamente a ação de descriptografar arquivos usando o git-secret de acordo com o seu ambiente.

Obs.: Lembre-se de verificar se você possui permissão de acesso a esses projetos no GitLab da UFSC antes de prosseguir com a configuração.

Instalação das Bibliotecas Localmente

Para garantir o correto funcionamento dos projetos, siga as instruções abaixo para instalar as bibliotecas necessárias localmente:

- **Ckeditor:** Se o projeto utiliza o editor de texto "Ckeditor", execute os seguintes comandos:
    1. Acesse o diretório do componente:

       ```
       cd stt-componentes/packages/stt-componentes-ckeditor/ckeditor5
       ```

    2. Instale as dependências deste projeto:

       ```
       npm install
       ```

    3. Realize o processo de build:

       ```
       npm run build
       ```

- **Bootstrap de Módulo:** Para o bootstrap do módulo desejado, utilize o seguinte comando:

  ```
  yarn bootstrap:<MODULO>
  ```

- **Execução do Modo de Desenvolvimento:** Para iniciar o modo de desenvolvimento do módulo especificado, execute o comando:

  ```
  yarn dev:<MODULO>
  ```

Obs.: Certifique-se de substituir `<MODULO>` pelo nome do módulo específico. Lembre-se de adaptar as instruções conforme necessário para o seu ambiente.

Obs.: Todos os comandos para bootstrap e execução em modo de desenvolvimento se encontram no arquivo package.json.

Uso Correto das Branches

Garantir um fluxo de trabalho organizado e colaborativo envolve o uso correto das branches do Git. Siga estas orientações para manter um controle adequado das branches:

- Dentro da pasta raiz do seu projeto, acesse a branch "default". A nomenclatura das branches pode variar de acordo com o projeto. Alguns projetos utilizam a branch "dev", enquanto outros usam "master" ou "main".
    - Exemplo: Para acessar a branch "dev", execute o seguinte comando:

      ```
      git checkout dev
      ```

- Certifique-se de que o seu repositório esteja atualizado. Atualize a branch "default" do projeto com o comando:
    - Exemplo: Para atualizar a branch "dev", execute o seguinte comando:

      ```
      git pull origin dev
      ```

- Crie uma nova tarefa a partir da branch "default", seguindo o padrão de nomenclatura "redmine-stt-#", onde "#" é o número da tarefa no Redmine. Por exemplo:

  ```
  git checkout -b redmine-stt-4272
  ```

Essas práticas ajudam a manter a organização das branches e facilitam a colaboração em equipe. Lembre-se de substituir `default` pela nomenclatura da branch específica do seu projeto e de adaptar as instruções conforme necessário.

Bom trabalho! ?