**Arquitetura**

**Desenho da Arquitetura**

Diagrama

Descrição gerada automaticamente

**Definições**

1. Geral

* Por se tratar de um serviço de demonstração o banco de dados utilizado será InMemory, isso é os registros ficam guardados em memória enquanto o serviço estiver em execução, caso o serviço seja reiniciado os dados armazenados como Refresh Token, Registros de Credito e Registros de Debito serão perdidos.
* A referência ao Database deve poder ser alterada a qualquer momento de maneira simples para que se ajuste a qualquer banco de dados relacional que tenha o drive do Entity Framework como SQL Server, Oracle, MySql, etc... e assim persistir os dados mesmo em caso de reinicialização dos serviços.
* Os Serviços deve estar preparado para ser disponibilizado pelo IIS, Docker (precisa ter o DockerFile) ou Self Hosted (nesse caso há a necessidade de criação de um serviço de health check, que irá monitorar se o serviço está no ar e, caso não esteja, colocá-lo no ar)
* As Datas devem ser armazenadas na globalização UTC. Sendo convertidas para LocalDateTime quando forem enviadas para o usuário.

1. Autenticação

O Serviço de Autenticação deve disponibilizar um endpoint Token que é responsável por validar o usuário e senha e caso estejam corretos devolve um Token JWT com validade de 30 Minutos e um Token de Refresh com validade de 1440 minutos (24 horas)

O Serviço de Autenticação deve disponibilizar um endpoint de refresh que é responsável por validar se o token de refresh é válido e emitir um novo Token JWT e um novo Token de Refresh sem a necessidade de se enviar novamente o Usuário e Senha.

Diretrizes:

* As chamadas aos endpoints de autenticação devem passar por uma camada de SSL TLS 1.3
* As senhas devem ser armazenadas em Hash com Salt, isso evita que duas senhas iguais tenham o mesmo Hash.
* As senhas devem ser comparadas Hash com Hash, uma vez que não há descriptografia para Hash.
* O Token JWT deve conter o UserName do usuário na Claim Sub, e a Role do Usuário na Claim Role
* O Serviço deve ser disponibilizado em WebApi Rest
* O Serviço deve estar disponível na Internet, com acesso anônimo.
* Por se tratar de um serviço de demonstração, não serão criados os endpoints relativos ao cadastro de usuários, sendo criados automaticamente por seed os seguintes usuários:
  + UserName: SystemAdmin / Password: Papa32%45TK / Role: AdmSistema
  + UserName: SCredito / Password: oLary%K2k\* / Role: Credito
  + UserName: SDebito / Password: posKj\*66&dj / Role: Debito
  + UserName: ManagerUser / Password: Papa32%45TK / Role: Gerencial
* Por se tratar de um serviço de demonstração não teremos policys como quantidade de tentativas de acesso, formato de senha, etc...

1. Gateway

O Gateway é responsável por centralizar todos os acessos externos as APIs, garantindo que o usuário esteja autenticado e que sua role permita acesso a API/Endpoint chamado.

Uma vez validado o token do usuário e a permissão de acesso por Role do serviço chamado o Gateway redireciona a chamada para o serviço solicitado.

O Gateway tem seus endpoints criados em runtime a partir de uma configuração no banco de dados, o que facilita a escalabilidade podendo-se a qualquer momento adicionar mais serviços a plataforma sem precisar alterar seu código fonte.

Diretrizes

* As chamadas aos endpoints do Gateway devem passar por uma camada de SSL TLS 1.3
* O Serviço deve ser disponibilizado em WebApi Rest
* O Serviço deve estar disponível na Internet, com acesso anônimo.
* Os endpoints devem validar se o usuário está autenticado por um Token JWT válido
* Os endpoints devem validar se o usuário tem a Role necessária para efetuar o acesso
* Os endpoints devem recuperar do Token o UserName (claim sub)
* Os endpoints devem redirecionar a chamada para o serviço correto, passando o UserName no Header da chamada
* Por se tratar de um serviço de demonstração, não serão criados os endpoints relativos ao cadastro de configuração do Gateway, sendo criados automaticamente por seed as configurações relativas ao serviço de fluxo de caixa. Caso seja necessário alterar alguma configuração como por exemplo a URL de Redirecionamento, o mesmo deve ser feito na Classe Context, que se encontra dentro da pasta Repository
* Inserir, Atualizar, Excluir, PesquisarPorData, PesquisarPorDescricao, Obter, ObterTodos de Crédito somente podem ser acessados por usuários com as Roles de AdmSistema e Credito.
* Inserir, Atualizar, Excluir, PesquisarPorData, PesquisarPorDescricao, Obter, ObterTodos de Débito somente podem ser acessados por usuários com as Roles de AdmSistema e Debito.
* Os relatórios RelatorioCategoria e RelatorioFormaPagamento só podem ser acessados por usuários com roles de AdmSistema e Gerencial

1. Fluxo de Caixa

O Serviço de Fluxo de Caixa é o serviço responsável por manter os registros de Débito e Crédito da empresa.

O Serviço deve prover as funcionalidades de Inserir, Atualizar, Excluir, PesquisarPorData, PesquisarPorDescricao, Obter, ObterTodos para crédito e para débito.

O Serviço deve prover os dados para os Relatórios Consolidados diários por Forma de Pagamento e por Categoria.

* O Serviço deve ser acessado apenas pelo Gateway, isso pode ser feito através do Cors ou bloqueio de acesso do servidor por IP/Porta
* Se possível o serviço não deve ser exposto na Internet, isso pode ser feito colocando em um servidor local da rede e/ou um servidor com as portas bloqueadas.
* O Serviço não precisa validar o usuário/role pois o Gateway já fez esse trabalho antes de chamá-lo.
* O Serviço não precisa SSL e TLS.