Requisitos - Micro serviço de security

Este documento tem como objetivo especificar os requisitos necessários para o desenvolvimento do micro serviço de security e o de gerenciamento de credenciais e acessos.

Micro Serviço de Gerenciamento de Credenciais e Acessos

O serviço de IAM deve implementar os seguintes requisitos funcionais:

- **Autenticação de dois fatores:**
  - O micro serviço deve permitir a autenticação de dois fatores (2FA) para aumentar a segurança do login. Ativado por padrão para colaboradores.
  - Os usuários podem escolher entre receber um código por SMS ou usar um **aplicativo de autenticação**.
- **Login:** permitir que os usuários façam login no sistema informando seu e-mail e senha. O sistema deve verificar se as credenciais informadas são válidas e, em caso afirmativo, gerar e retornar um token de acesso, um token de atualização e um remember token para o usuário. **Qualquer um que não seja funcionário poderá fazer login via google, facebook, instagram ou microsoft, do contrário poderá fazer login via google ou o padrão**.
- **Gestão de sessões:** “Gerador de sessão” - **Responsável: José Robson Siqueira Junior**
- **Atualização de Token:** permitir que os usuários atualizem seu token de acesso utilizando o token de atualização. O sistema deve verificar se o token de atualização é válido e gerar um novo token de acesso para o usuário.
- **Revogação de Token:** permitir que os usuários revoguem seu token de acesso, invalidando-o imediatamente. O sistema deve verificar se o token de acesso é válido antes de revogá-lo.
- **Verificação de Token:** permitir que os serviços da empresa verifiquem se um token de acesso é válido e se o usuário associado tem permissão para acessar o serviço solicitado.
- **Administração de Usuários:** permitir que os administradores do sistema gerenciem os usuários do sistema, incluindo adicionar, excluir e atualizar usuários, bem como definir seus papéis e permissões. O sistema deve garantir que apenas os administradores tenham permissão para executar essas operações.
- **Gerenciamento de Papéis de Usuário (roles) ou Grupos de Acesso:** permitir que os administradores do sistema definam os papéis de usuário, garantindo que cada usuário tenha o acesso adequado às tarefas e projetos relevantes.
- **Gerenciamento de Permissões de Acesso:** permitir que os administradores do sistema definem as permissões de acesso aos recursos do sistema, garantindo que apenas usuários autorizados tenham acesso a informações sensíveis ou ações

críticas do sistema. Criar permissões por meio de interface gráfica, podendo vincular a serviços ou a ações específicas dentro de determinados serviços.

- **Gerenciamento de funções:**
    - O micro serviço deve permitir a criação, leitura, atualização e exclusão de funções (papéis de usuário).
    - As funções devem ter permissões predefinidas, que serão aplicadas a todos os usuários que possuírem essa função.
    - As funções predefinidas incluem: admin, user

      Atribuição de permissões:

    - O micro serviço deve permitir a atribuição de permissões individuais a usuários específicos, independentemente de sua função.
    - As permissões incluem: criação, leitura, atualização e exclusão de usuários; busca por data de criação; busca por status do usuário; alteração de status do usuário; alteração do papel do usuário; remoção de permissão; adição de permissão; listagem de papéis do sistema; gerenciamento de grupos de usuários; suporte a vários idiomas; autenticação de dois fatores; personalização de perfil; integração com serviços de terceiros; recuperação de senha; exportação de dados do usuário. Segue link com mais permissões e papéis: https://docs.google.com/document/d/100zDwTJstPeqXhUxh9Ml43h-hJMmH5AOoe3TVmGyRbM/edit?usp=sharing
    - As permissões devem ser atribuídas individualmente e podem ser revogadas a qualquer momento.
    - O micro serviço deve permitir o gerenciamento de permissões de forma granular, de modo que os administradores possam definir permissões específicas para cada usuário ou grupo de usuários.
    - Os usuários com permissões específicas devem ter acesso somente às funcionalidades para as quais foram autorizados.
- **Gerenciamento de grupos de acesso:**
    - O micro serviço deve permitir o gerenciamento de grupos de usuários.
    - Os usuários podem ser organizados em grupos para facilitar o gerenciamento de permissões.
    - Os grupos podem ser criados e editados pelos usuários com função de SuperAdmin ou Admin.
    - Os usuários podem ser atribuídos a um ou mais grupos.
    - As permissões podem ser atribuídas a grupos inteiros, de modo que todos os usuários pertencentes a esse grupo recebam automaticamente essas permissões.
    - Atribuir uma permissão a um grupo não afeta os usuários que já têm essa permissão atribuída individualmente. Além disso, a remoção de uma permissão de um grupo não afeta os usuários que possuem essa permissão atribuída individualmente. Isso permite que os administradores concedam ou removam permissões em massa, com base nos grupos de usuários, sem interferir nas permissões individuais que possam ter sido definidas anteriormente.

    - O micro serviço deve permitir que os administradores atribuam permissões a grupos inteiros de usuários, de forma a agilizar o processo de gerenciamento de permissões. Os grupos podem ser criados e editados pelos usuários com função de SuperAdmin ou Admin, e podem ser atribuídos a um ou mais usuários.
    - As permissões atribuídas a um grupo serão automaticamente aplicadas a todos os usuários que pertencem a esse grupo. No entanto, se um usuário tiver permissões atribuídas individualmente, essas permissões não serão afetadas pelas permissões atribuídas ao grupo. Isso permite aos administradores personalizar as permissões de usuários individuais, ao mesmo tempo em que gerenciam permissões em massa por meio de grupos.
    - Os usuários também devem ter a capacidade de visualizar as permissões que foram atribuídas a eles, tanto individualmente quanto por meio de grupos. Isso ajudará a garantir que os usuários tenham a visibilidade necessária sobre suas permissões no sistema e possam relatar quaisquer problemas aos administradores, se necessário.

O serviço de IAM também deve atender aos seguintes requisitos de interface:

- O micro serviço deve fornecer uma interface fácil de usar para atribuir e gerenciar permissões de grupo. Os administradores devem ser capazes de selecionar o grupo para o qual deseja atribuir permissões e, em seguida, selecionar as permissões que desejam atribuir a esse grupo. As permissões atribuídas a um grupo devem ser claramente exibidas na interface do usuário, para que os administradores possam ver facilmente quais permissões estão sendo aplicadas a cada grupo.

O serviço de IAM também deve atender aos seguintes requisitos não funcionais:

- Segurança: o sistema deve armazenar senhas criptografadas e utilizar algoritmos seguros de hash. O serviço deve utilizar um método seguro de comunicação (por exemplo, HTTPS) para transmitir informações confidenciais entre os serviços.
- Escalabilidade: o serviço deve ser escalável horizontalmente para lidar com um grande número de usuários e solicitações de serviços simultâneas.
- Disponibilidade: o serviço deve ter alta disponibilidade e tolerância a falhas para garantir que os usuários possam fazer login e acessar os serviços da empresa a qualquer momento.

Requisitos funcionais

1. Autenticação de usuários através de login e senha.
2. Autenticação de usuários através de token de autenticação.
3. Autenticação de usuários através de login social: Google, Facebook, Instagram e Microsoft.
4. A autenticação deve ser segura e utilizar padrões abertos, como OAuth ou OpenID Connect.
5. Gerenciamento de permissões de acesso aos recursos do sistema.
6. Gerenciamento de tokens de acesso

    1. O serviço deve permitir que os usuários gerenciem seus tokens de acesso, incluindo revogação, renovação e expiração.
    2. O serviço deve enviar notificações para os usuários quando seus tokens estiverem prestes a expirar.
7. Gerenciamento de refresh token:
    1. O serviço deve gerar e enviar um refresh token para os usuários após a autenticação bem-sucedida.
    2. O refresh token deve ser criptografado e possuir um tempo de expiração configurável.
    3. O refresh token deve permitir que o usuário obtenha um novo access token sem precisar se autenticar novamente.
8. Gerenciamento de remember token:
    1. O serviço deve permitir que os usuários sejam autenticados automaticamente quando acessarem o serviço novamente, sem precisar digitar suas credenciais.
    2. O serviço deve armazenar o remember token de forma segura, utilizando um algoritmo de hash seguro.
    3. O remember token deve ter um tempo de expiração configurável e ser exclusivo para cada usuário.
9. Criptografia de dados sensíveis, como senhas e informações pessoais dos usuários.
10. Registro de log de acesso e ações realizadas pelos usuários.
11. Possibilidade de bloqueio de usuários e revogação de permissões de acesso.
12. Bloqueio temporário de usuário após várias tentativas falhas de login.
13. Criptografia de dados entre front-end e back-end
    1. O serviço deve criptografar todas as informações sensíveis que trafegam entre o front-end e o back-end.
    2. O serviço deve utilizar um algoritmo de criptografia forte e configurável.
14. Controle de acesso granular
    1. O serviço deve permitir que os administradores do sistema configurem regras de controle de acesso granular para os usuários.
    2. As regras de controle de acesso devem ser aplicadas em diferentes níveis, como nível de serviço, nível de recurso e nível de operação.
    3. O serviço deve suportar diferentes tipos de usuários, como usuários regulares, administradores, superadministradores, etc.
    4. O sistema deve permitir a configuração de permissões e papéis de usuário, garantindo que apenas as pessoas certas tenham acesso às tarefas e projetos relevantes.
15. Detecção de atividades suspeita
    1. O serviço deve monitorar as atividades dos usuários e detectar atividades suspeitas, como tentativas de acesso não autorizadas e login de usuários em locais incomuns.
    2. O serviço deve enviar notificações para os usuários e administradores do sistema quando atividades suspeitas forem detectadas.
    3. O serviço deve registrar as atividades suspeitas em um log de segurança para análise posterior.
    4. O serviço deve implementar medidas de segurança adicionais, como bloqueio de conta e solicitação de confirmação de identidade, em casos de atividades suspeitas.

16. Auditoria de segurança
    1. O serviço deve registrar todas as atividades sensíveis dos usuários, como tentativas de login, mudanças de senha e outras atividades relevantes.
    2. O serviço deve armazenar os logs de auditoria de segurança em um local seguro e criptografado.
    3. O serviço deve permitir que os administradores do sistema acessem os logs de auditoria para fins de investigação e análise.
17. Proteção contra ataques de força bruta
    1. O serviço deve implementar medidas de segurança para evitar ataques de força bruta, como bloqueio temporário ou permanente de contas após várias tentativas fracassadas de login.
    2. O serviço deve permitir que os usuários desbloqueiem suas contas ou redefinam suas senhas após um determinado período de tempo.
18. Proteção contra ataques de injeção de código (por exemplo, SQL injection, XSS, MITM)
    1. O serviço deve implementar medidas de segurança para evitar ataques de injeção de código, como validação de entrada de dados, sanitização de dados e filtragem de caracteres especiais.
    2. O serviço deve utilizar ferramentas de segurança, como firewalls e antivírus, para proteger o sistema contra ataques de injeção de código.
    3. O serviço deve ser atualizado regularmente com correções de segurança e patches de segurança para evitar vulnerabilidades conhecidas.

Requisitos não-funcionais

1. O micro serviço deve ser desenvolvido em linguagem Java.
2. O micro serviço deve seguir os padrões de segurança recomendados pela OWASP.
3. O micro serviço deve ser escalável e suportar um grande volume de acessos simultâneos.
4. O micro serviço deve lidar com picos de tráfego de forma eficiente.
5. O tempo de resposta do micro serviço deve ser inferior a 500ms.
6. O micro serviço deve ser rápido e responsivo, com tempos de resposta baixos e capacidade de lidar com grandes volumes de dados.
7. O micro serviço deve ser integrado com um sistema de gerenciamento de identidade e acesso (IAM) já existente na empresa.
8. Deve ser possível monitorar o desempenho e a disponibilidade do micro serviço.
9. O micro serviço deve estar disponível 24 horas por dia, 7 dias por semana, com tempo de inatividade mínimo.
10. O micro serviço deve ser seguro e proteger as informações confidenciais dos usuários contra ameaças externas e internas.
11. Conformidade: o serviço deve estar em conformidade com os regulamentos de segurança e privacidade aplicáveis, como a LGPD, GDPR e HIPAA.
12. Testabilidade: o serviço deve ser fácil de testar e validar, com testes automatizados e ferramentas de simulação de carga.
13. Manutenibilidade: o serviço deve ser fácil de manter e atualizar, com uma arquitetura modular e documentação abrangente.

14. Interoperabilidade: o serviço deve ser capaz de interoperar com outros sistemas e serviços, utilizando padrões abertos e APIs bem documentadas.
15. Monitoramento: o serviço deve ser monitorado continuamente para garantir a disponibilidade, desempenho e segurança adequados. O serviço deve gerar alertas e notificações em caso de problemas.
16. Capacidade de registro: o serviço deve ser capaz de registrar eventos e atividades relevantes em um log de segurança seguro e criptografado.
17. Backup e recuperação: o serviço deve ter um sistema de backup e recuperação robusto, capaz de recuperar dados e informações em caso de falha do sistema.
18. Tolerância a falhas: o serviço deve ser tolerante a falhas, com mecanismos de redundância e recuperação automática em caso de falhas.
19. Adequação: o serviço deve ser adequado às necessidades e requisitos dos usuários e da empresa, com funcionalidades e recursos adequados e suficientes para atender às demandas.

Requisitos de interface

1. O micro serviço deve disponibilizar uma API REST para acesso aos recursos do sistema.
2. A API REST deve seguir o padrão JSON para envio e recebimento de dados.
3. A documentação da API REST deve ser disponibilizada em formato Swagger.

Relacionamentos entre entidades

Como teremos várias entidades, estas que possuem interface com usuários e suas credenciais, segue abaixo uma sugestão de árvore de relacionamentos para este fim:

1. **Sistema de Gerenciamento de Identidade e Acesso (IAM);** O serviço de IAM seria responsável por gerenciar a autenticação e autorização de usuários em todos os outros serviços. Ele poderia ser implementado utilizando uma arquitetura baseada em tokens, como JWT ou OAuth2, para permitir a autenticação e autorização seguras dos usuários. As entidades e atributos necessários para esse serviço incluem:
    - **Entidade Usuário:** ID, nome, sobrenome, e-mail, senha, papel (por exemplo, administrador, usuário comum), permissões (excluir usuário, adicionar tarefa a usuário, etc), provider_id, provider_name, avatar.
    - **Entidade Token:** token de acesso, token de atualização, remember token, data de expiração do token de acesso, data de expiração do token de atualização, data de expiração do remember token.
    - **Entidade Papel:** ID, nome, descrição.
    - **Entidade Permissão:** ID, nome, descrição.
    - **Atributos de Configuração:** chave secreta para assinatura de token, tempo de expiração de token.
2. **Sistema de Gerenciamento de Usuários;** O serviço de gerenciamento de usuários seria responsável por gerenciar os dados dos usuários do sistema, incluindo informações de perfil e configurações. Ele poderia ser implementado utilizando uma

arquitetura de microserviços com um banco de dados para armazenamento dos dados. As entidades e atributos necessários para esse serviço incluem:

- **Entidade Usuário:** ID, nome, sobrenome, e-mail, senha, papel (por exemplo, administrador, usuário comum), preferências de usuário (por exemplo, idioma, tema).
- **Atributos de Configuração:** conexão com o banco de dados, credenciais de acesso ao banco de dados.

Observação: Nesta etapa, apenas trabalharemos com os micro serviços de Gerenciamento de Identidade e Acesso (IAM) junto ao Security.

O micro serviço de security é responsável por fornecer as funcionalidades de segurança em um sistema, garantindo a proteção dos dados e informações dos usuários. A autenticação e autorização de usuários são fundamentais para garantir a segurança do sistema, e por isso, o micro serviço deve oferecer diferentes formas de autenticação, como login e senha ou token de autenticação.

Além disso, o gerenciamento de permissões de acesso aos recursos do sistema é essencial para garantir que somente usuários autorizados tenham acesso a informações sensíveis ou ações críticas do sistema. Por isso, o micro serviço deve oferecer recursos para gerenciar as permissões de acesso dos usuários.

A criptografia de dados sensíveis, como senhas e informações pessoais dos usuários, é outra funcionalidade importante do micro serviço de security. Essa criptografia deve ser forte o suficiente para tornar os dados ilegíveis em caso de acesso não autorizado.

O registro de log de acesso e ações realizadas pelos usuários é fundamental para garantir a transparência e a responsabilização no uso do sistema. Dessa forma, o micro serviço deve disponibilizar recursos para registro de logs.

A possibilidade de bloqueio de usuários e revogação de permissões de acesso é importante em casos de suspeita de acesso não autorizado ou comportamento inadequado dos usuários. Por isso, o micro serviço deve oferecer recursos para bloqueio de usuários e revogação de permissões de acesso.

Os requisitos não-funcionais do micro serviço de security incluem o desenvolvimento em linguagem Java, seguindo os padrões de segurança recomendados pela OWASP. O micro serviço deve ser escalável e suportar um grande volume de acessos simultâneos, com tempo de resposta inferior a 500ms. A integração com um sistema de gerenciamento de identidade e acesso já existente na empresa é necessária para garantir a interoperabilidade com outros sistemas.

Por fim, o micro serviço deve disponibilizar uma API REST para acesso aos recursos do sistema, seguindo o padrão JSON para envio e recebimento de dados. A documentação da API REST deve ser disponibilizada em formato Swagger para facilitar o desenvolvimento de clientes que utilizam a API.

Requisitos - Micro Serviço de Gerenciamento de Usuários.

O micro serviço de gerenciamento de usuários tem como objetivo fornecer funcionalidades relacionadas ao gerenciamento de usuários do sistema, incluindo a criação, atualização e exclusão de usuários, além de recursos para gerenciamento de permissões e papéis de usuário.

Requisitos Funcionais

1. **Cadastro de Usuário:** permitir que os usuários se registrem no sistema informando seu nome, sobrenome, e-mail e senha. O sistema deve criptografar a senha utilizando um algoritmo seguro de hash antes de armazená-la no banco de dados.
2. Busca por data de criação
3. Busca por status do usuário
4. **Alteração de status do usuário:** permite aos administradores ou moderadores alterar o status de um usuário, como ativar ou desativar uma conta de usuário, bloquear um usuário por comportamento inadequado ou suspender um usuário por violações de termos de uso.
5. **Suporte a vários idiomas:**
    1. O micro serviço deve oferecer suporte a múltiplos idiomas, permitindo que os usuários selecionem seu idioma preferido para a interface do usuário.
    2. Os idiomas suportados devem ser definidos no sistema.
6. Personalização de perfil: O micro serviço deve permitir que os usuários personalizem seu perfil, adicionando informações pessoais e escolhendo suas preferências de interface.
7. Integração com serviços de terceiros
8. **Exportação de dados do usuário:**
    1. O micro serviço deve permitir que os usuários exportem seus próprios dados pessoais em um formato estruturado. Antes de exportar, o usuário terá que consentir e tal ação será registrada para fins de auditoria, no banco de dados (LGPD). - Nesta tabela nenhuma ação além de findAll, findOne ou insertOne será permitida.
    2. Os usuários devem ter controle sobre quais dados deseja exportar.
    3. A exportação de dados deve ser protegida por padrões de segurança, para garantir que as informações exportadas sejam acessadas somente pelo usuário autorizado.
9. Recuperação de Senha: permitir que os usuários redefinam suas senhas em caso de esquecimento ou perda.
10. Bloqueio de Usuários: permitir que os administradores do sistema bloqueiem usuários em casos de comportamento inadequado ou suspeita de acesso não autorizado.
11. Registro de Log de Acesso e Ações de Usuário: permitir que o sistema registre o acesso e as ações realizadas pelos usuários, garantindo a transparência e a responsabilização no uso do sistema.

Requisitos Não-Funcionais

1. O micro serviço deve ser desenvolvido em linguagem Java, seguindo as melhores práticas de programação desse ambiente.
2. O micro serviço deve ser escalável, permitindo lidar com um grande volume de acessos simultâneos.
3. O tempo de resposta do micro serviço deve ser inferior a 500 ms, garantindo um desempenho ágil e eficiente.
4. O micro serviço deve utilizar métodos seguros de comunicação, como HTTPS, para transmitir informações confidenciais entre os serviços.
5. O micro serviço deve ser integrado com um sistema de gerenciamento de identidade e acesso (IAM) já existente na empresa, garantindo uma gestão unificada e organizada dos usuários.
6. A documentação da API REST deve ser disponibilizada em formato Swagger, permitindo que os desenvolvedores façam consultas e testes com maior facilidade.
7. O micro serviço deve atender aos requisitos de segurança e privacidade aplicáveis, como a LGPD, GDPR e HIPAA, garantindo o cumprimento das leis e regulamentações vigentes.
8. O micro serviço deve ser fácil de testar e validar, com testes automatizados e ferramentas de simulação de carga, garantindo a qualidade do serviço.
9. O micro serviço deve ser fácil de manter e atualizar, com uma arquitetura modular e documentação abrangente, facilitando a realização de manutenções e atualizações.

Requisitos de Interface

1. O micro serviço deve disponibilizar uma API REST para acesso aos recursos do sistema, seguindo as melhores práticas de desenvolvimento de APIs RESTful.
2. A API REST deve seguir o padrão JSON para envio e recebimento de dados, garantindo uma comunicação eficiente e padronizada.
3. A documentação da API REST deve ser disponibilizada em formato Swagger, facilitando o consumo e a consulta da documentação pelos desenvolvedores.

Relacionamentos entre Entidades

As principais entidades envolvidas no micro serviço de gerenciamento de usuários são:

1. Entidade Usuário: ID, nome, sobrenome, e-mail, senha, papel (por exemplo, administrador, usuário comum), permissões (excluir usuário, adicionar tarefa a usuário, etc.).
2. Entidade Registro de Log: ID, data e hora, usuário, ação realizada.

O micro serviço de gerenciamento de usuários deve permitir que os administradores do sistema gerenciem essas entidades, incluindo a criação, atualização e exclusão de usuários, papéis e permissões. Além disso, o serviço deve permitir que os administradores

gerenciem as permissões de acesso aos recursos do sistema, garantindo que apenas usuários autorizados tenham acesso a informações sensíveis ou ações críticas do sistema. O registro de log de acesso e ações de usuário é fundamental para garantir a transparência e a responsabilização no uso do sistema e deve ser implementado no micro serviço de gerenciamento de usuários.

Em resumo, o micro serviço de gerenciamento de usuários deve fornecer uma solução completa e eficiente para o gerenciamento de usuários do sistema, garantindo a segurança, a privacidade e a organização das informações dos usuários. Com uma arquitetura modular, uma documentação abrangente e uma interface padronizada, o serviço deve ser fácil de manter, atualizar e integrar com outros sistemas, tornando-se uma peça fundamental na construção de sistemas robustos e escaláveis.