Autorização e controle de acesso

SEG786203 – CST em Análise e Desenvolvimento de Sistemas

Prof. Emerson Ribeiro de Mello

mello@ifsc.edu.br

INSTITUTO FEDERAL
Santa Catarina
Campus São José

Licenciamento

Sumário

Modelos de controle de acesso

Controle de acesso e autorização

Política de controle de acesso

Usuário possui **autorização** para passar pela catraca

Mecanismo de controle de acesso

Modelo de controle de acesso

- Consiste em uma abstração para políticas e mecanismos de controle de acesso
- Permite que usuários, administradores e desenvolvedores possam entender e implementar mecanismos de controle de acesso

Termo	Descrição
Sujeito	Entidade que acessa um objeto
Objeto	Recurso que é acessado por um sujeito por meio de uma operação
Operação	Ação realizada por um sujeito em um objeto
Permissão	Autorização para realizar uma operação em um objeto

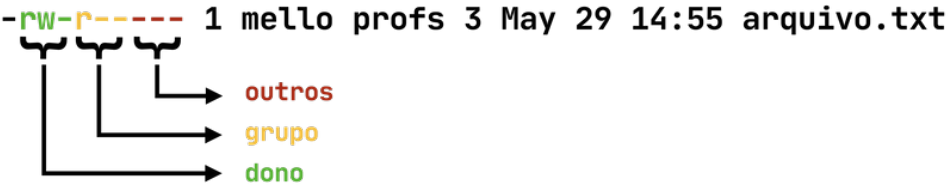
Modelos de controle de acesso

Controle de acesso discricionário (DAC)

Controle de acesso discricionário

***Discretionary Access Control* (DAC)**

- Todo **objeto possui** um **dono** o qual **possui discricionariedade** (poder) sobre o objeto, podendo conceder ou revogar permissões de acesso a outros sujeitos ou grupos
- Exemplo: Controle de acesso a arquivos nos sistemas POSIX
  - Cada arquivo possui um dono e um grupo, sendo que o dono pode conceder permissões de acesso a outros usuários



Lista de controle de acesso

***Access Control List* (ACL)**

- Lista de regras que especifica quais sujeitos podem acessar quais objetos e quais operações podem ser realizadas sobre os objetos

	disciplinas.txt	notas.txt	comum.txt
Alice	{leitura, escrita}	{leitura, escrita}	{leitura, escrita}
Bob	{leitura}		{leitura, escrita}
Charles			{leitura, escrita}

Quais permissões cada usuário possui sobre os arquivos

- Tipos de ACLs
  - Sistemas de arquivos: POSIX[1], NTFS, NFSv4
  - Redes e sistemas: roteadores, *firewalls*, serviço de diretórios, etc

[1] https://wiki.debian.org/Permissions#Access_Control_Lists_in_Linux

Lista de controle de acesso

***Access Control List* (ACL)**

- Exemplo de ACL em um *firewall* utilizando o *Uncomplicated Firewall* (UFW)

```
sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 3306/tcp                   ALLOW IN    192.168.0.2
[ 2] 3306/tcp                   DENY IN     Anywhere
[ 3] 192.168.0.1 OpenSSH        DENY IN     Anywhere
[ 4] 80/tcp                     ALLOW IN    Anywhere
[ 5] 443/tcp                    ALLOW IN    Anywhere
[ 6] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 7] 443/tcp (v6)               ALLOW IN    Anywhere (v6)
```

Lista de controle de acesso

***Access Control List* (ACL)**

- Exemplo de ACL em um serviço de diretórios OpenLDAP
  - Administrador pode escrever em todas entradas abaixo do ramo `ou=people`
  - Segunda regra é mais específica, dando o direito de escrita apenas ao criador da entrada

```
access to dn.exact="ou=people,dc=raiz" attrs=children
        by group.exact="cn=admins,ou=group,dc=raiz" write
        by * none
access to dn.children="ou=people,dc=raiz" attrs=entry,@extensibleObject
        by set="this/creatorsName & user" write
        by group.exact="cn=admins,ou=group,dc=raiz" read
        by group.exact="cn=leitores,ou=group,dc=raiz" read
        by set="this/-* & user" read
        by * none
```

Exercício – 10 minutos

Em dupla

- Para um sistema acadêmico, pense em uma política de controle de acesso que siga o modelo DAC
  - É necessário definir sujeitos, objetos e as permissões (direitos)
- Descreva como seria implementado o mecanismo de controle de acesso
  - Escolha um caso de uso do sistema acadêmico
  - Detalhe onde ficariam armazenadas as políticas, como e quando seriam consultadas e atualizadas

Modelos de controle de acesso

Controle de acesso obrigatório (MAC)

Controle de acesso obrigatório

***Mandatory Access Control* (MAC)**

- Definida de forma centralizada por um administrador, impõe regras de acesso obrigatórias (restrições) que não podem ser alteradas pelos sujeitos
- Em sistemas operacionais os sujeitos podem ser processos e objetos podem ser arquivos, diretórios, sockets, etc.
- Em sistemas de banco de dados os sujeitos podem ser usuários e objetos podem ser tabelas, visões, procedimentos armazenados, etc.

```
GRANT SELECT ON academico.Disciplina TO 'alice'@'localhost';
REVOKE ALL ON academico.Disciplina FROM 'bob'@'%';
```

Código: Exemplo de controle de acesso obrigatório em um banco de dados MySQL

Modelos de controle de acesso

Controle de acesso baseado em papéis (RBAC)

Controle de acesso baseado em papéis

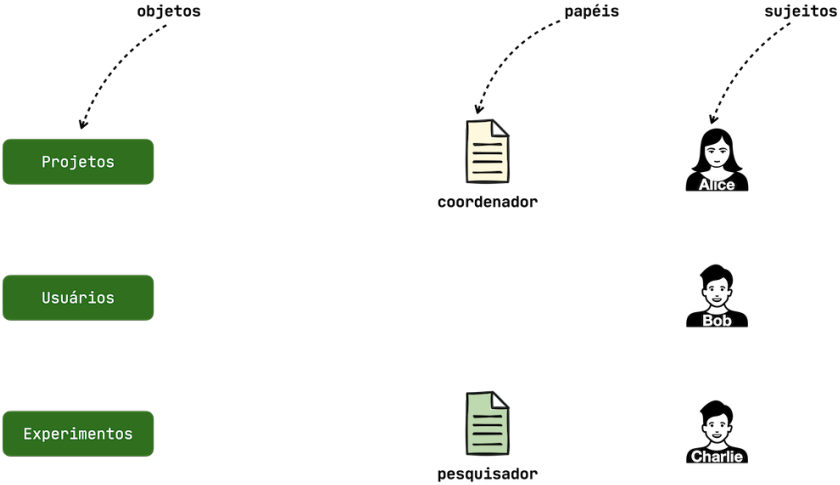
***Role-Based Access Control* (RBAC)**

Permite a implementação de políticas de controle de acesso de forma mais flexível e escalável, quando comparado com DAC e MAC

- **Papel**, função desempenhada por um **sujeito**
  - Ex: professor, aluno, administrador, etc.
- **Permissões** sobre **objetos** são associadas a **papéis**
  - Ex: coordenador pode criar disciplinas
- A um **sujeito** é associado a um ou mais **papéis**
  - Ex: Alice é professora

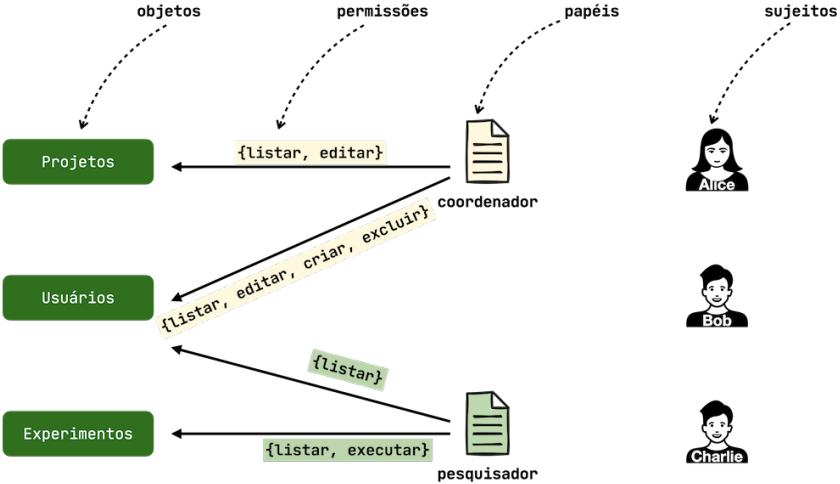
Controle de acesso baseado em papéis

***Role-Based Access Control* (RBAC)**



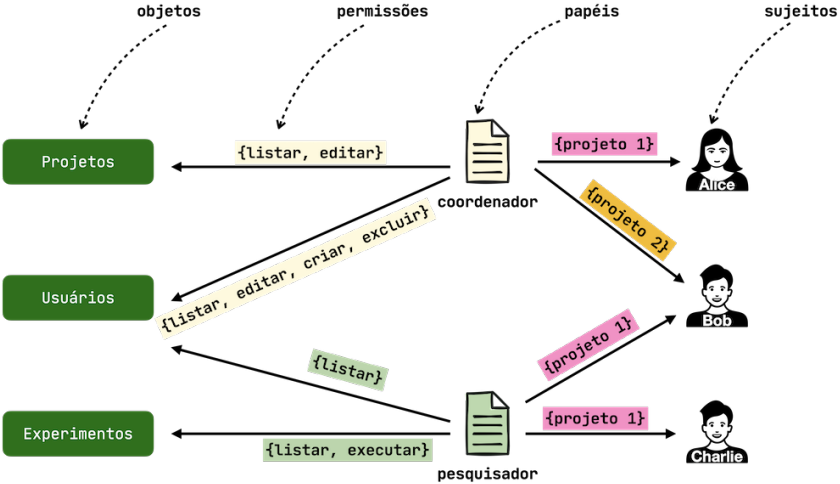
Controle de acesso baseado em papéis

***Role-Based Access Control* (RBAC)**



Controle de acesso baseado em papéis

***Role-Based Access Control* (RBAC)**



Controle de acesso baseado em papéis

***Role-Based Access Control* (RBAC)**

- Um **sujeito pode assumir um ou mais papéis**, podendo essa atribuição ser dinâmica ou estática
- **RBAC estático**, um sujeito assume um papel permanentemente
  - Todos os papéis são atribuídos a um sujeito no momento de sua criação
  - Todos os papéis estão ativos ao mesmo tempo
- **RBAC dinâmico**, um sujeito assume um papel temporariamente
  - limitação de tempo
  - limitação de contexto
  - limitação de sessão

Controle de acesso baseado em papéis

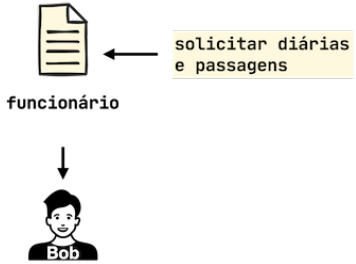
***Role-Based Access Control* (RBAC)**

- **Separação de responsabilidades** (SoD, *Separation of Duties*) é um conceito que visa evitar conflitos de interesse, fraudes e erros
- **Sujeito não pode ter permissões conflitantes**, sendo que a execução de uma operação **requer a participação de mais de um sujeito**

Controle de acesso baseado em papéis

***Role-Based Access Control* (RBAC)**

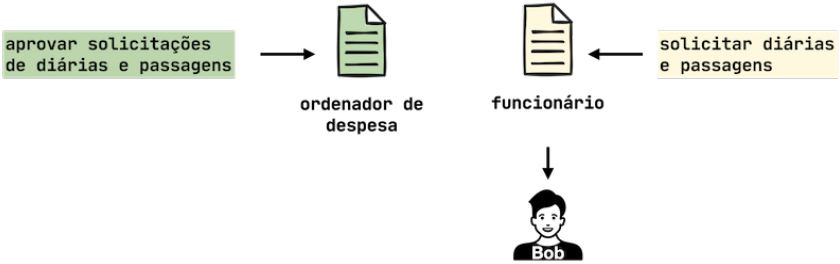
- **Separação de responsabilidades** (SoD, *Separation of Duties*) é um conceito que visa evitar conflitos de interesse, fraudes e erros
- **Sujeito não pode ter permissões conflitantes**, sendo que a execução de uma operação **requer a participação de mais de um sujeito**



Controle de acesso baseado em papéis

***Role-Based Access Control* (RBAC)**

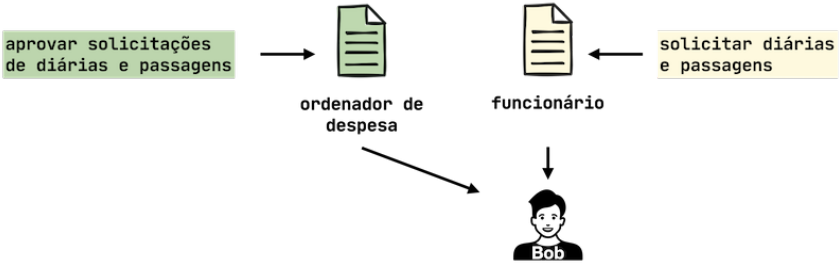
- **Separação de responsabilidades** (SoD, *Separation of Duties*) é um conceito que visa evitar conflitos de interesse, fraudes e erros
- **Sujeito não pode ter permissões conflitantes**, sendo que a execução de uma operação **requer a participação de mais de um sujeito**



Controle de acesso baseado em papéis

***Role-Based Access Control* (RBAC)**

- **Separação de responsabilidades** (SoD, *Separation of Duties*) é um conceito que visa evitar conflitos de interesse, fraudes e erros
- **Sujeito não pode ter permissões conflitantes**, sendo que a execução de uma operação **requer a participação de mais de um sujeito**

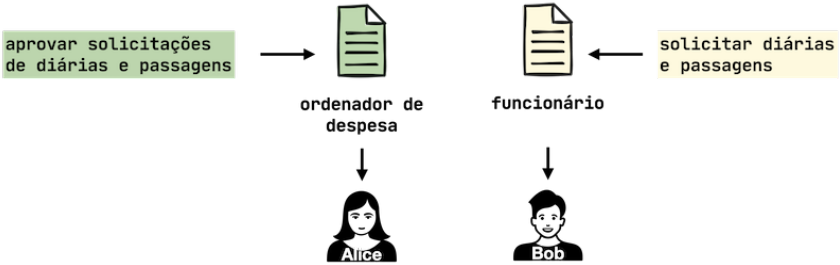


Bob poderá aprovar suas próprias despesas, o que não é desejado

Controle de acesso baseado em papéis

***Role-Based Access Control* (RBAC)**

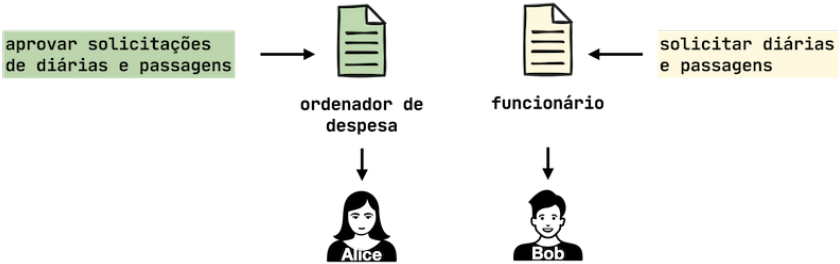
- **Separação de responsabilidades** (SoD, *Separation of Duties*) é um conceito que visa evitar conflitos de interesse, fraudes e erros
- **Sujeito não pode ter permissões conflitantes**, sendo que a execução de uma operação **requer a participação de mais de um sujeito**



Controle de acesso baseado em papéis

***Role-Based Access Control* (RBAC)**

- **Separação de responsabilidades** (SoD, *Separation of Duties*) é um conceito que visa evitar conflitos de interesse, fraudes e erros
- **Sujeito não pode ter permissões conflitantes**, sendo que a execução de uma operação **requer a participação de mais de um sujeito**



Como fazer se Alice também é uma funcionária?

Controle de acesso baseado em papéis

***Role-Based Access Control* (RBAC)**

- **Separação de responsabilidades estática**
  - Política de controle de acesso define quais papéis não poderão ser assumidos simultaneamente ou restrições com relação ao objeto de acesso
- **Separação de responsabilidades dinâmica**
  - Quais papéis um sujeito pode assumir simultaneamente em um contexto ou sessão específica
  - Seria possível um sujeito ser “funcionário” e “ordenador de despesas”, porém não poder aprovar suas próprias despesas

Controle de acesso baseado em papéis

***Role-Based Access Control* (RBAC)**

- **RBAC hierárquico**, um papel herda permissões de outro papel enquanto adiciona novas permissões
  - Captura a estrutura organizacional de uma empresa
- Exemplo
  - O **pesquisador** pode listar projetos
  - O **coordenador** pode listar e criar projetos
  - O **administrador** pode listar, criar e excluir projetos

Controle de acesso baseado em papéis

***Role-Based Access Control* (RBAC)**

- **RBAC simplifica a definição de políticas de controle de acesso** com base nas funções que um sujeito desempenha em uma organização
  - A administração dos mecanismos de controle de acesso se resume a adicionar ou remover usuários de papéis
- A facilidade da administração vem com um **custo de complexidade na definição dos papéis e permissões**
  - "engenharia de papéis" consiste na definição de todos os papéis e permissões
  - Maior granularidade de papéis e permissões resulta em maior complexidade

ACL vs RBAC I

Manter ACL é mais trabalhoso do que manter os papéis e permissões (RBAC)

Usuário	Permissões		
	Criar	Listar	Excluir
Alice	✓	✓	✓
Bob	✓	✓	
Charles	✓		
Daiana	✓		
Eve	✓	✓	

Tabela: Exemplo de ACL

ACL vs RBAC II

Manter ACL é mais trabalhoso do que manter os papéis e permissões (RBAC)

Papel	Permissões		
	Criar	Listar	Excluir
Pesquisador	✓		
Coordenador	✓	✓	
Administrador	✓	✓	✓

Tabela: Exemplo de RBAC, permissões associadas a papéis

Usuário	Papéis
Alice	Administrador
Bob	Coordenador
Charles	Pesquisador
Daiana	Pesquisador
Eve	Coordenador

Tabela: Exemplo de RBAC, atribuição de papéis

Exercício – 10 minutos

Em dupla

- Para um sistema de biblioteca, pense em uma política de controle de acesso que siga o modelo RBAC
  - É necessário definir todos os papéis, direitos e a associação com os sujeitos
- Descreva como seria implementado o mecanismo de controle de acesso
  - Escolha um caso de uso do sistema de biblioteca
  - Detalhe onde ficariam armazenadas as políticas, como e quando seriam consultadas e atualizadas

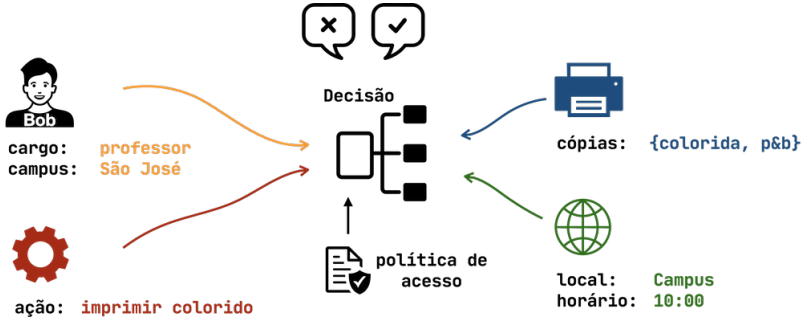
Modelos de controle de acesso

Controle de acesso baseado em atributos (ABAC)

Controle de acesso baseado em atributos

***Attribute-Based Access Control* (ABAC)**

- No ABAC as decisões de acesso são feitas sobre os **valores dos atributos** associados a sujeitos, objetos e ambiente



- Ex: Permitir que professor de São José possa imprimir colorido em uma impressora específica, quando estiver no campus e em horário comercial

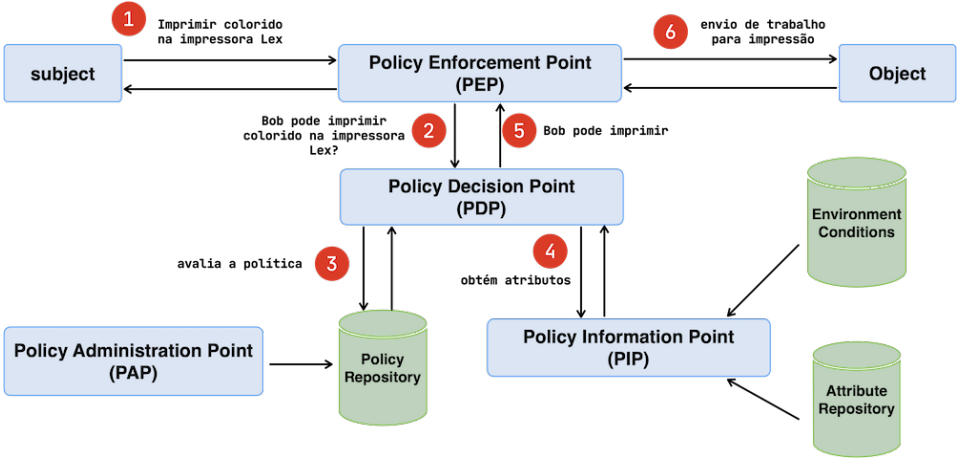
Controle de acesso baseado em atributos

***Attribute-Based Access Control* (ABAC)**

- Os **atributos** podem ser **atômicos** (ex: data de nascimento) ou **multivalorado** (ex: cargos que um usuário desempenha)
- **ABAC** também é conhecido como **controle de acesso baseado em políticas** (PBAC, *Policy-Based Access Control*)
  - Usa políticas ao invés de permissões estáticas para controlar o acesso
- **XACML** é um exemplo de implementação de **ABAC**
  - padrão para especificação de políticas de controle de acesso em XML e um protocolo para avaliação de políticas

XACML – *eXtensible Access Control Markup Language*

Componentes da arquitetura



Fonte: Adaptado de Mello *et al.* (2022)

Modelos de controle de acesso

Controle de acesso baseado em relacionamentos (ReBAC)

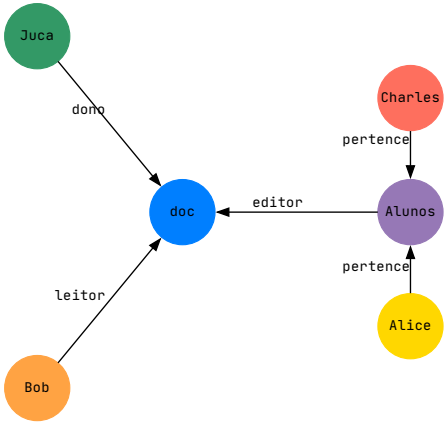
Controle de acesso baseado em relacionamentos I

***Relationship-Based Access Control* (ReBAC)**

- Políticas de controle de acesso definidas com base em **relacionamentos entre sujeitos e recursos**
  - Sujeito – pode ser um usuário, grupo, aplicativo, etc.
  - Objeto – pode ser um arquivo, diretório, banco de dados, etc.
  - Relacionamento – autoria, colaboração, revisão, etc.
- **Adequado para ambientes dinâmicos**, pois se adapta as alterações nos relacionamentos entre sujeitos e objetos
  - Ex: Em uma rede social, o acesso a uma publicação depende do relacionamento entre os usuários e não apenas seus papéis ou atributos que possuem

Controle de acesso baseado em relacionamentos II

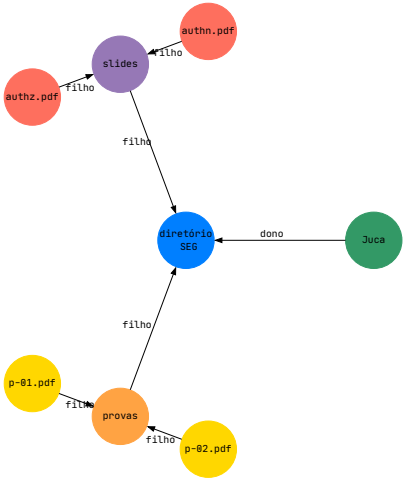
***Relationship-Based Access Control* (ReBAC)**



- Uma política no ReBAC
  - *Um usuário pode visualizar um documento se ele for o dono, se recebeu o direito de leitor ou se fizer parte de um grupo que está diretamente envolvido com o documento*
- ReBAC pode ser visualizado como um **grafo direcionado**
  - **Nós** representam sujeitos e recursos
  - **Arestas** representam relacionamentos

ReBAC – tipos de relacionamentos

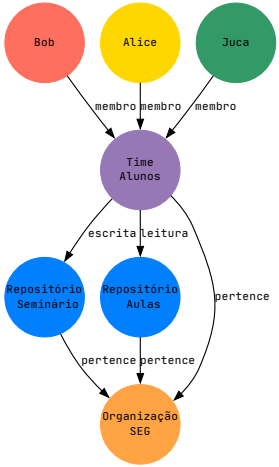
Hierárquico



- Descreve o aninhamento de recursos dentro de outros recursos
- Juca é dono do **diretório** `SEG` e por consequência será dono de todos os arquivos que estiveram abaixo desse diretório

ReBAC – tipos de relacionamentos

Organização ou grupos



- Baseado em grupos de usuários
- Usuários pertencentes ao time `Alunos` podem escrever no repositório `Seminário` e ler o repositório `Aulas`

Implementação de ReBAC na prática

- Zanzibar (Pang *et al.*, 2019) é um exemplo de sistema que implementa **ReBAC** no Google
  - Sistema global de autorização que gerencia políticas de controle de acesso para milhares de serviços e bilhões de solicitações por segundo
  - Utilizado em sistemas como Google Cloud, Google Photos, Google Drive, etc.
- Ponteiros interessantes
  - https://research.google/pubs/pub48190
  - https://zanzibar.academy

Tecnologias

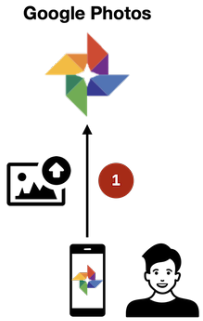
Interação entre sistemas

Um sistema acessar recursos em nome de um usuário em outro sistema

- Juca tira fotos com seu celular

Interação entre sistemas

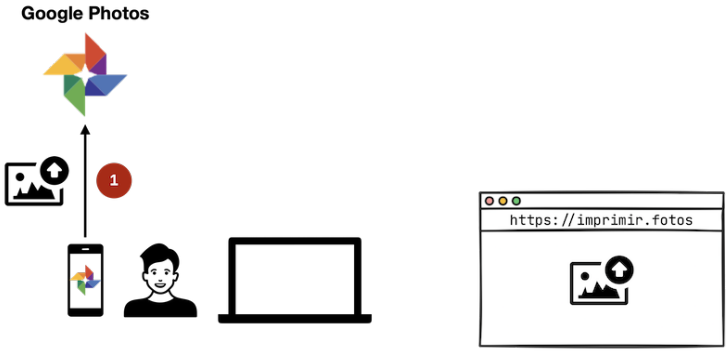
Um sistema acessar recursos em nome de um usuário em outro sistema



- As fotos são sincronizadas com o Google Fotos

Interação entre sistemas

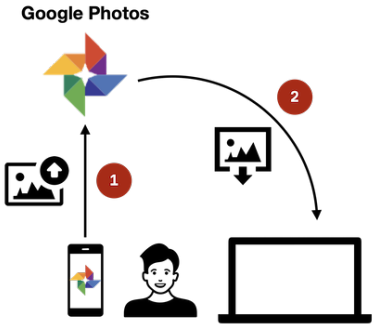
Um sistema acessar recursos em nome de um usuário em outro sistema

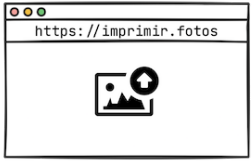


- Juca usa seu computador para acessar *site* para imprimir fotos

Interação entre sistemas

Um sistema acessar recursos em nome de um usuário em outro sistema

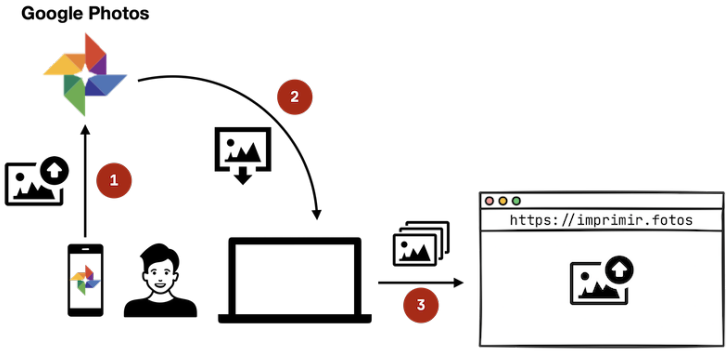




- Juca precisa baixar as fotos do Google Fotos para o seu computador

Interação entre sistemas

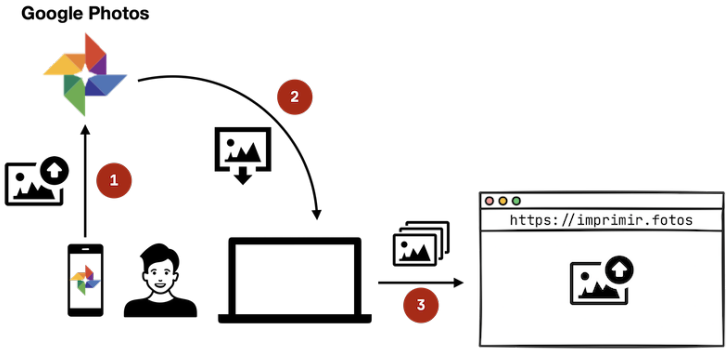
Um sistema acessar recursos em nome de um usuário em outro sistema



- E depois subir as fotos para o *site* de impressão

Interação entre sistemas

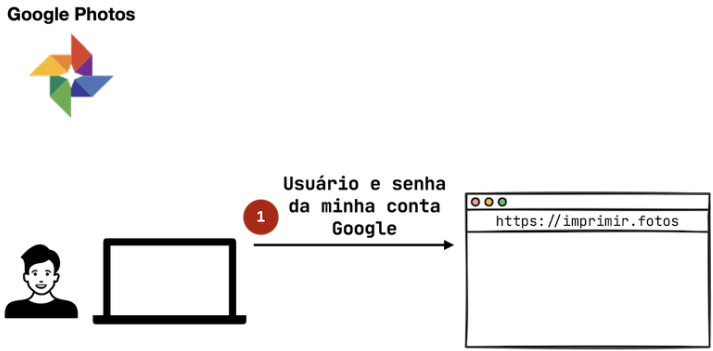
Um sistema acessar recursos em nome de um usuário em outro sistema



- Juca não gosta de baixar as fotos e depois subir para o *site* de impressão

Interação entre sistemas

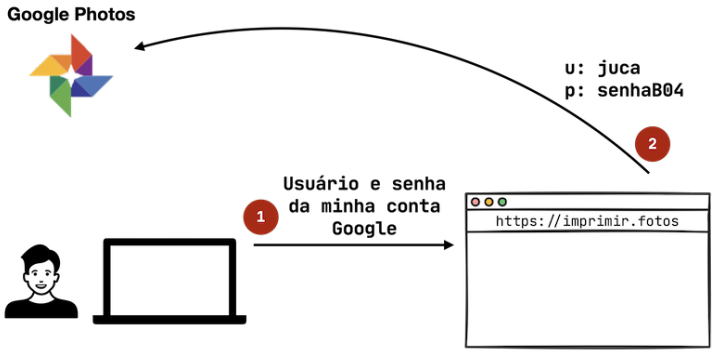
Um sistema acessar recursos em nome de um usuário em outro sistema



- Juca fornece usuário e senha da sua conta Google para o *site* de impressão

Interação entre sistemas

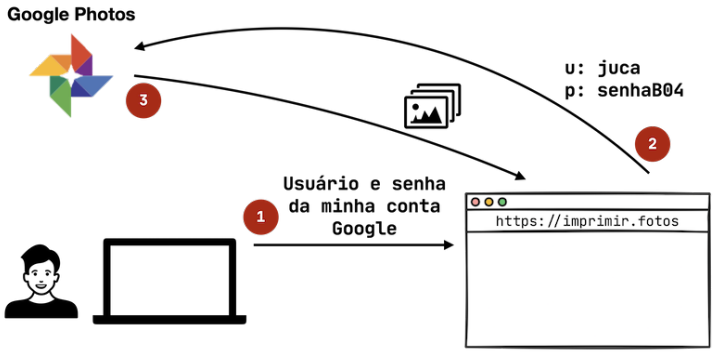
Um sistema acessar recursos em nome de um usuário em outro sistema



- *Site* de impressão acessa o Google Fotos em nome de Juca

Interação entre sistemas

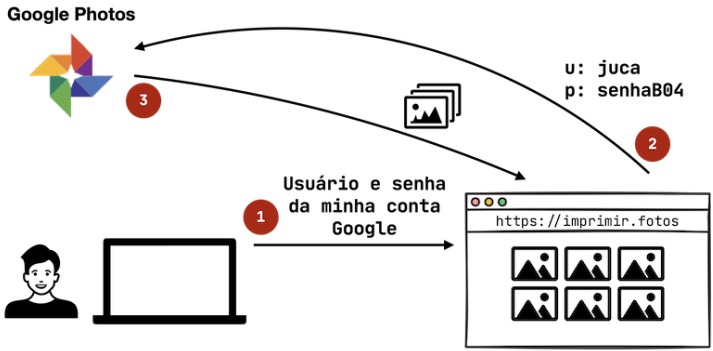
Um sistema acessar recursos em nome de um usuário em outro sistema



- Google Fotos envia as fotos para o *site* de impressão

Interação entre sistemas

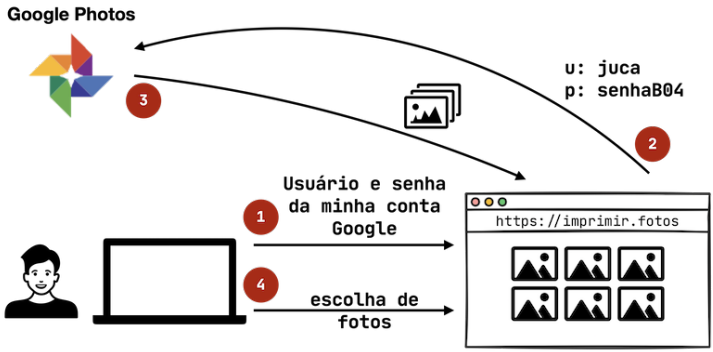
Um sistema acessar recursos em nome de um usuário em outro sistema



- *Site* de impressão lista as fotos para Juca escolher quais imprimir

Interação entre sistemas

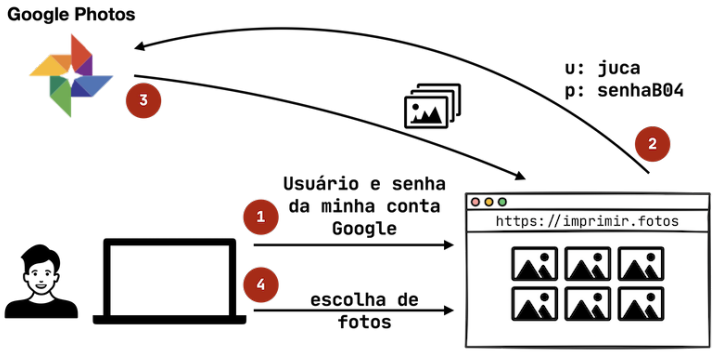
Um sistema acessar recursos em nome de um usuário em outro sistema



- Juca escolhe as fotos para imprimir

Interação entre sistemas

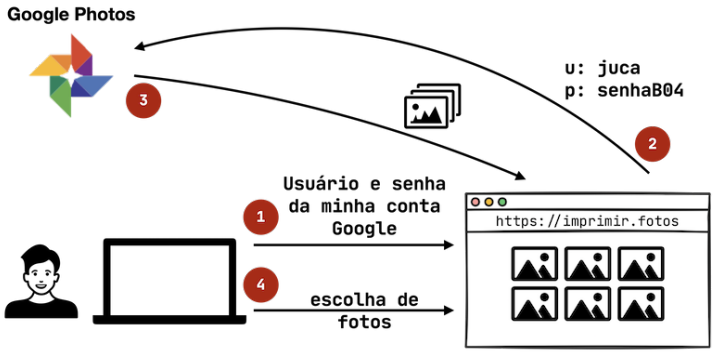
Um sistema acessar recursos em nome de um usuário em outro sistema



- *Site* de impressão tem acesso total e irrestrito a conta Google de Juca

Interação entre sistemas

Um sistema acessar recursos em nome de um usuário em outro sistema



- Juca não parece ser muito cuidadoso com suas senhas

OAuth2: *framework* de autorização

https://www.rfc-editor.org/rfc/rfc6749.txt

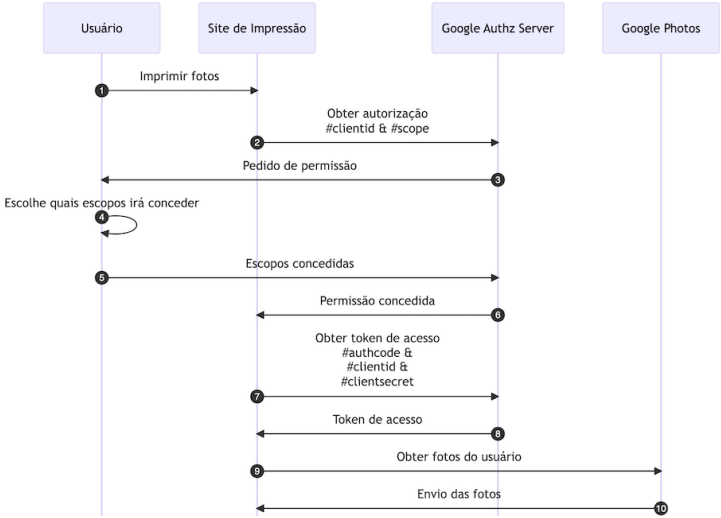
- Protocolo de autorização que permite a um site ou aplicativo acessar recursos hospedados por outros sites em nome de um usuário
- Fornece acesso limitado (escopos) a recursos protegidos sem a necessidade de compartilhar senhas
  - Ex: Juca autoriza o *site* de impressão a acessar suas fotos no Google Fotos
- O JWT é geralmente usado para representar o **token de acesso**
  - Tempo de vida limitado e pode ser revogado a qualquer momento
  - Pode carregar informações sobre o usuário e o aplicativo
- Amplamente utilizado na indústria (em APIs, aplicações web, etc.)
  - Ex: Google, Facebook, GitHub, Conta Gov.BR, etc.

OAuth2

Papéis

- **Proprietário dos recursos** (usuário)
  - Autoriza o acesso a seus recursos
- **Cliente** (site de impressão)
  - Aplicativo que deseja acessar os recursos do usuário
- **Servidor de autorização** (Google Authz Server)
  - Recebe solicitações do cliente para tokens de acesso e os emite mediante autenticação e consentimento do usuário
- **Servidor de recursos** (Google Fotos)
  - Servidor que hospeda os recursos protegidos

OAuth2: fluxo de autorização



OAuth2

Tipos de concessão

- **Código de autorização** (*Authorization Code Flow*)
  - Utilizado por aplicações web, onde o cliente é um servidor web
  - Utilizado por aplicações móveis com o PKCE (*Proof Key for Code Exchange*)
- **Senha do proprietário** (*Resource Owner Password Flow*)
  - Utilizado por aplicações confiáveis, como aplicações nativas no dispositivo do usuário
- **Credenciais do cliente** (*Client Credentials Flow*)
  - Utilizado por aplicações que acessam recursos em seu próprio nome
  - Aplicações não interativas, como processos automatizados, microsserviços
- **Fluxo implícito** (*Implicit Flow with Form Post*)
  - Aplicações JavaScript executadas no navegador do usuário (ex: *Single Page Applications*)

OAuth2

Alguns ponteiros úteis

- *OAuth 2.0 Security Best Current Practice*
  - https://datatracker.ietf.org/doc/html/draft-ietf-oauth-security-topics
- Modelos de ataque e considerações de segurança
  - https://datatracker.ietf.org/doc/html/rfc6819
- Como usar o OAuth 2.0 para acessar as APIs do Google
  - https://developers.google.com/identity/protocols/oauth2?hl=pt-br
- Autorização OAuth2.0 com o Microsoft Entra ID
  - https://learn.microsoft.com/pt-br/entra/architecture/auth-oauth2
- Aplicações OAuth2.0 com o GitHub
  - https://docs.github.com/en/apps/oauth-apps/building-oauth-apps/authorizing-oauth-apps

Curiosidades

Venda de milhas aéreas

Maxmilhas[3]

- **Maxmilhas**[2] é uma empresa que compra milhas aéreas de usuários e revende passagens aéreas para terceiros
- Você deve **fornecer seu usuário e senha** do programa de milhagem
  - A Maxmilhas acessa o programa de milhagem com sua senha
- A Maxmilhas tem acesso total e irrestrito a sua conta do programa de milhagem
  - Pode emitir passagens aéreas, transferir milhas, etc.

Juca confia!

[2] Cuidado, em recuperação judicial em 21/09/23

[3] https://ajuda.maxmilhas.com.br/hc/pt-br/articles/29181987777043-Verifica%C3%A7%C3%A3o-de-duas-etapas-Smiles

Referências I

MELLO, Emerson Ribeiro de *et al.* Autenticação e Autorização: antigas demandas, novos desafios e tecnologias emergentes. *In:* MINICURSOS do XXI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg). Porto Alegre, RS, set. 2022. DOI: https://doi.org/10.5753/sbc.10710.3.1.

PANG, Ruoming *et al.* Zanzibar: Google's Consistent, Global Authorization System. *In:* 2019 USENIX Annual Technical Conference (USENIX ATC '19). Renton, WA, 2019.

Direitos autorais das imagens

- Alguns ícones presentes nas ilustrações foram obtidos de https://uxwing.com ou de https://www.flaticon.com