



# Autenticação e Autorização com FastAPI

Nesta apresentação, exploraremos os conceitos de autenticação e autorização usando a biblioteca FastAPI, com foco em mecanismos de geração de tokens, armazenamento em cookies e segurança.



# Autenticação e Autorização: Uma Introdução

## Autenticação

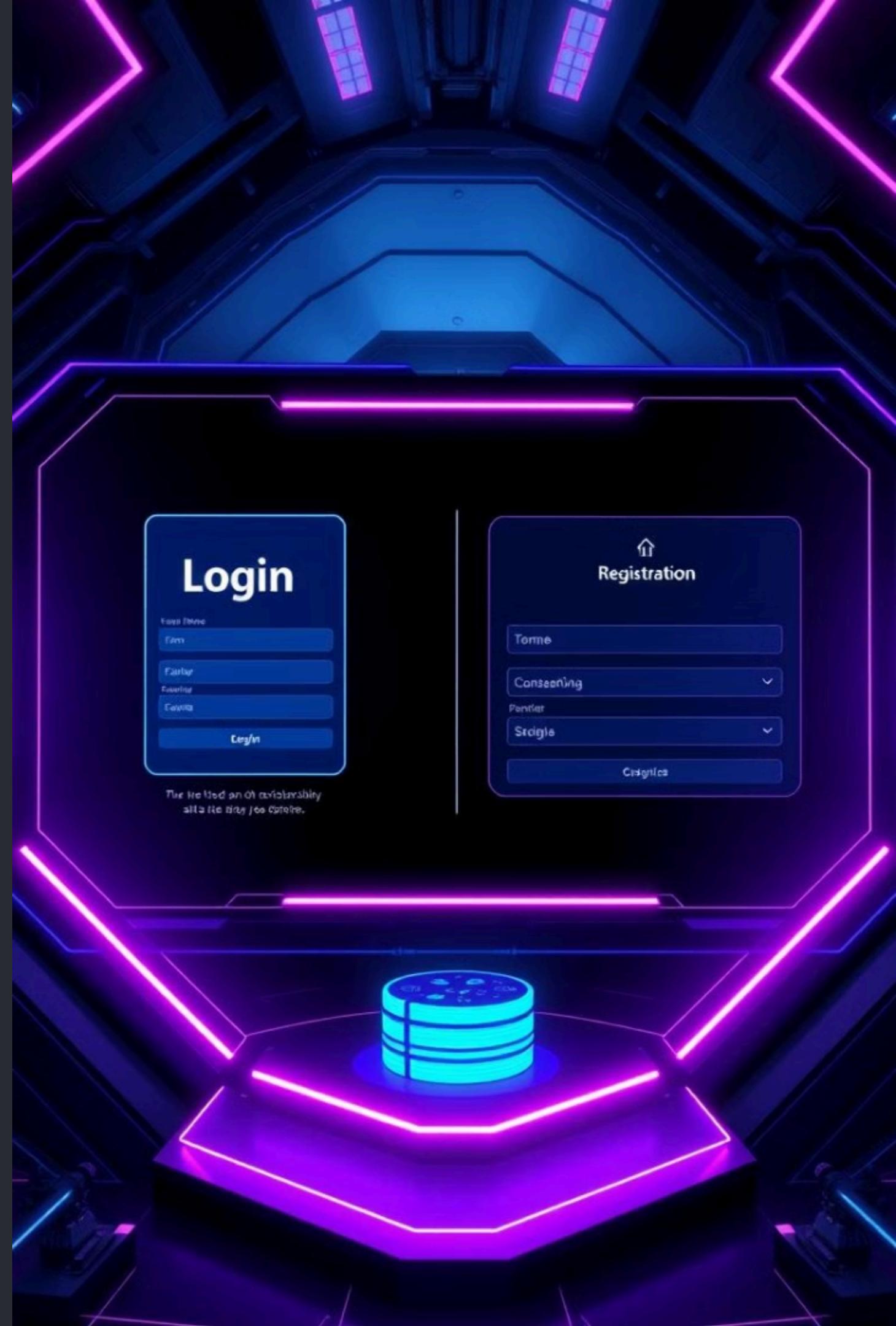
É o processo de verificar se um usuário é quem ele afirma ser. Isso envolve a validação de credenciais como nome de usuário e senha.

## Autorização

É o processo de definir quais recursos um usuário autenticado pode acessar. Isso envolve a atribuição de permissões específicas a cada usuário.

# Fluxo de Autenticação: Do Cadastro ao Login

- 1 O usuário cria uma conta fornecendo informações pessoais e escolhendo uma senha segura.
- 2 O sistema verifica se o nome de usuário já existe e salva as credenciais criptografadas em um banco de dados.
- 3 O usuário insere suas credenciais (nome de usuário e senha) e o sistema valida se elas correspondem aos registros do banco de dados.





# Geração de Tokens de Autenticação

## Token JWT (JSON Web Token)

É um padrão de token que permite que o sistema emita um token seguro para autenticar o usuário.

## Criptografia

Os tokens JWT são criptografados para proteger as informações do usuário e evitar a manipulação não autorizada.

## Validade

Os tokens têm um tempo de vida definido para garantir que os usuários não tenham acesso indefinido ao sistema.

# Armazenamento Seguro de Tokens nos Cookies



## Cookies HTTP

Os cookies são pequenos pedaços de dados armazenados no navegador do usuário para guardar informações sobre a sessão.



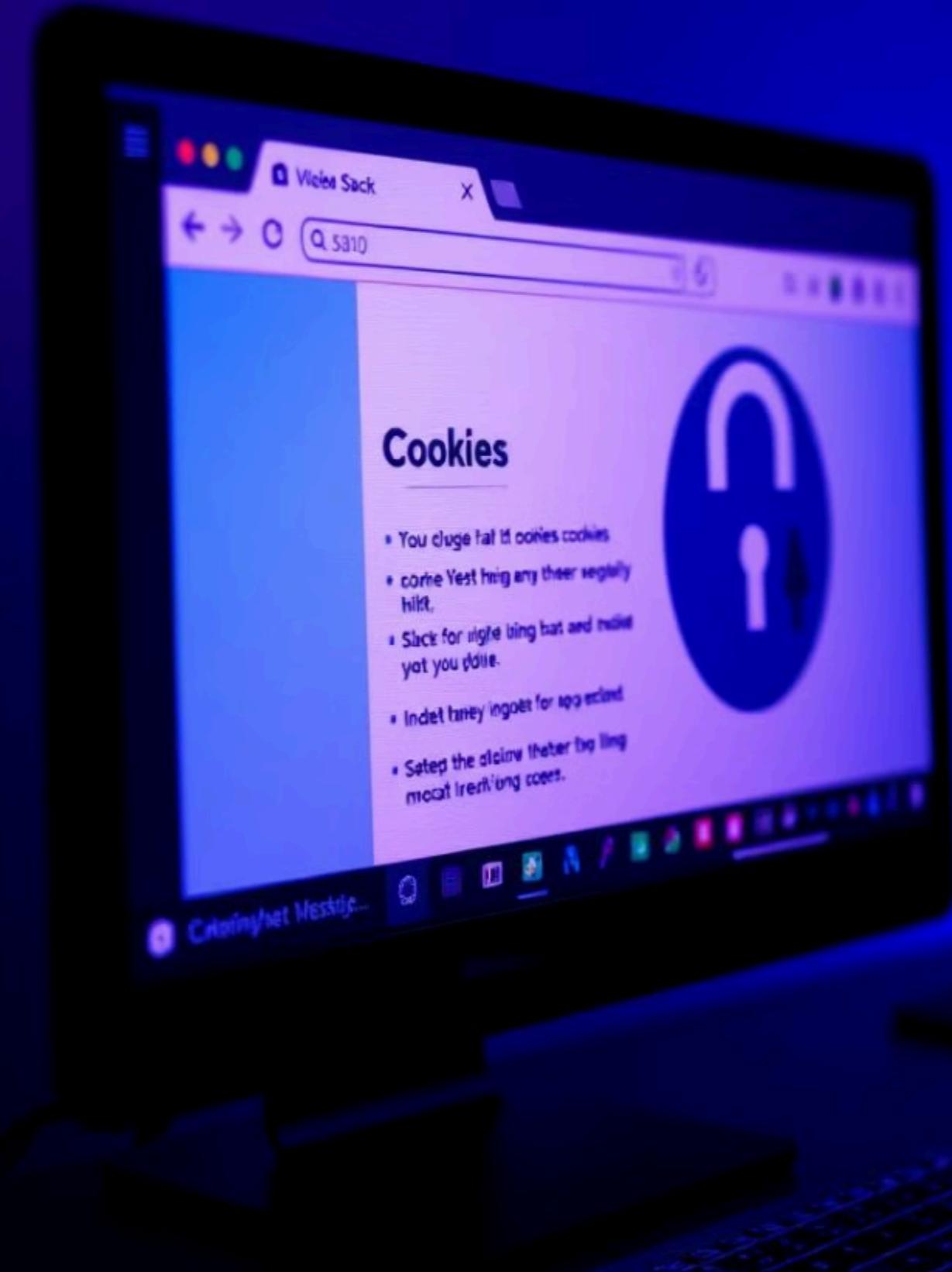
## Segurança

O armazenamento do token no cookie é feito de forma segura, com medidas para evitar ataques de Cross-Site Scripting (XSS).



## Transmissão Segura

A transmissão do token entre o servidor e o navegador é feita através de conexões HTTPS para garantir a confidencialidade dos dados.



# Verificação de Identidade do Usuário

1

## Verificação do Token

2

### Validação de Assinatura

O servidor verifica se a assinatura do token é válida, garantindo que ele não foi adulterado.

3

### Verificação de Expiração

O servidor verifica se o token ainda é válido, comparando a data atual com a data de expiração.

4

### Extração de Informações

O servidor extrai informações do token, como o ID do usuário e o escopo de suas permissões.

# Controle de Acesso com Base nas Permissões

## Definir Permissões

1

O sistema define permissões específicas para cada recurso, como acesso a determinadas áreas do site ou funções.

## Atribuir Permissões

2

Cada usuário recebe um conjunto de permissões de acordo com seu papel ou perfil no sistema.

## Validar Permissões

3

O sistema verifica se o usuário autenticado tem permissão para acessar o recurso solicitado.



# Implementação da Autenticação no Código

1

**Instalação do FastAPI**

2

**Configurar a Rota de Login**

Definir uma rota para o login do usuário e a geração do token JWT.

3

**Criar Decoradores**

Criar decoradores para proteger as rotas, garantindo que apenas usuários autenticados podem acessar.

4

**Implementar a Validação de Tokens**

Implementar a lógica para verificar o token e extrair as informações do usuário.



# Boas Práticas de Segurança

1

## Criptografia Forte

Utilizar algoritmos de criptografia robustos para proteger as credenciais do usuário.

2

## Gerenciamento de Senhas

Implementar mecanismos de gerenciamento de senhas seguras, como hashing e salting.

3

## Controle de Acesso

Implementar medidas de controle de acesso para restringir o acesso a recursos sensíveis.

4

## Atualizações de Segurança

Manter o FastAPI e as bibliotecas de segurança atualizadas para corrigir vulnerabilidades.

# Conclusão e Próximos Passos



A autenticação e autorização com FastAPI garantem a segurança e o controle de acesso em aplicações web. Para aprofundar seus conhecimentos, explore documentação e bibliotecas específicas.