



# REDES DE COMPUTADORES



# REDES DE COMPUTADORES

AULAS 06 e 07



AGENDA

**Camada de Aplicação**  
**Serviços/Aplicações**  
**APIs Webs**  
**Protocolos – HTTP**  
**TED Atividade/Seminário**



# VIMOS QUE...

## OSI

APLICAÇÃO

APRESENTAÇÃO

SESSÃO

TRANSPORTE

REDE

ENLACE

FÍSICA

## TCP/IP

APLICAÇÃO

TRANSPORTE

REDE

ENLACE

FÍSICA

# VIMOS QUE...

## CAMADA DE APLICAÇÃO

- DISPONIBILIZAÇÃO DE SERVIÇOS/  
APLICAÇÕES

É a camada que te permite usar comunicação em uma rede. Vai fazer a interface com os usuários, disponibilizar serviços, como e-mail, acesso a transferência de arquivos remotos, gerenciamento de bancos de dados compartilhados... e outros tipos de serviços de informações distribuídas , ou seja , **é o que você vai fazer de fato!**

**PRIMEIRAMENTE, O QUE SERIAM OS  
SERVIÇOS?**

The background of the slide is a dark blue-grey color. It is covered with numerous 3D question marks of varying sizes and orientations. Some question marks are light blue and stand out more than others, which are darker and blend into the background. The overall effect is a textured, three-dimensional field of question marks.

# **INTEGRAÇÃO DE SERVIÇOS EXTERNOS EM APLICATIVOS?**

# INTEGRAÇÃO DE SERVIÇOS EXTERNOS

A integração de serviços externos em aplicativos se refere ao processo de incorporar funcionalidades ou dados de serviços de terceiros em um aplicativo que está sendo desenvolvido.



**“Isso é fundamental para desenvolvedores de aplicativos e sistemas para Internet, uma vez que muitos aplicativos modernos dependem de serviços e recursos externos para ampliar suas funcionalidades e oferecer uma experiência mais rica aos usuários”.**



# INTEGRAÇÃO DE SERVIÇOS EXTERNOS



**Imagine que você está construindo um aplicativo ou um site e deseja adicionar recursos que não fazem parte do seu aplicativo principal, como um sistema de pagamento, mapas, autenticação de usuário ou integração com redes sociais...**



**Em vez de reinventar a roda e criar esses recursos do zero, você pode se beneficiar da integração de serviços externos!**



# INTEGRAÇÃO DE SERVIÇOS EXTERNOS

**No geral, a integração de serviços externos é uma parte essencial do desenvolvimento de aplicativos e sistemas para Internet, pois permite que os desenvolvedores aproveitem as capacidades de terceiros para enriquecer seus aplicativos e fornecer uma experiência mais completa aos usuários.**



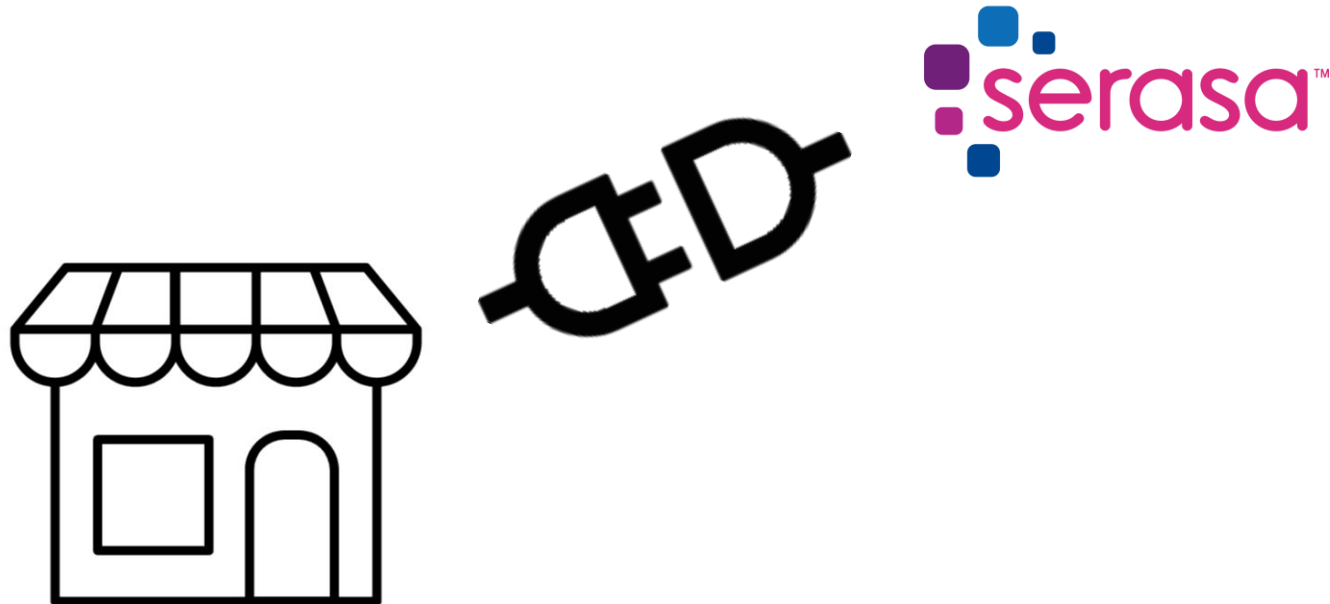
**A LOJA POSSUI UM SISTEMA DE VENDAS,  
MAS ANTES DE VENDER PRECISA  
CONSULTAR EM OUTRO SISTEMA (EX:  
SPC/SERASA) SE O CLIENTE ESTÁ  
NEGATIVADO. E AGORA?  
JÁ QUE SÃO DOIS SISTEMAS, DOIS BANCOS  
DE DADOS DIFERENTES?**

The background is a dark, textured surface covered with numerous 3D letters and symbols in various colors (shades of blue, grey, and white). The letters are of different sizes and are scattered across the frame, creating a sense of depth and complexity. A single, prominent white lowercase 'u' is positioned in the upper left quadrant, slightly above the main text.

**ESSES SERVIÇOS PRECISAM ESTAR  
INTEGRADOS.**

# INTEGRAÇÃO DE SERVIÇOS EXTERNOS

Exemplo:  
Loja com Sistema de Vendas

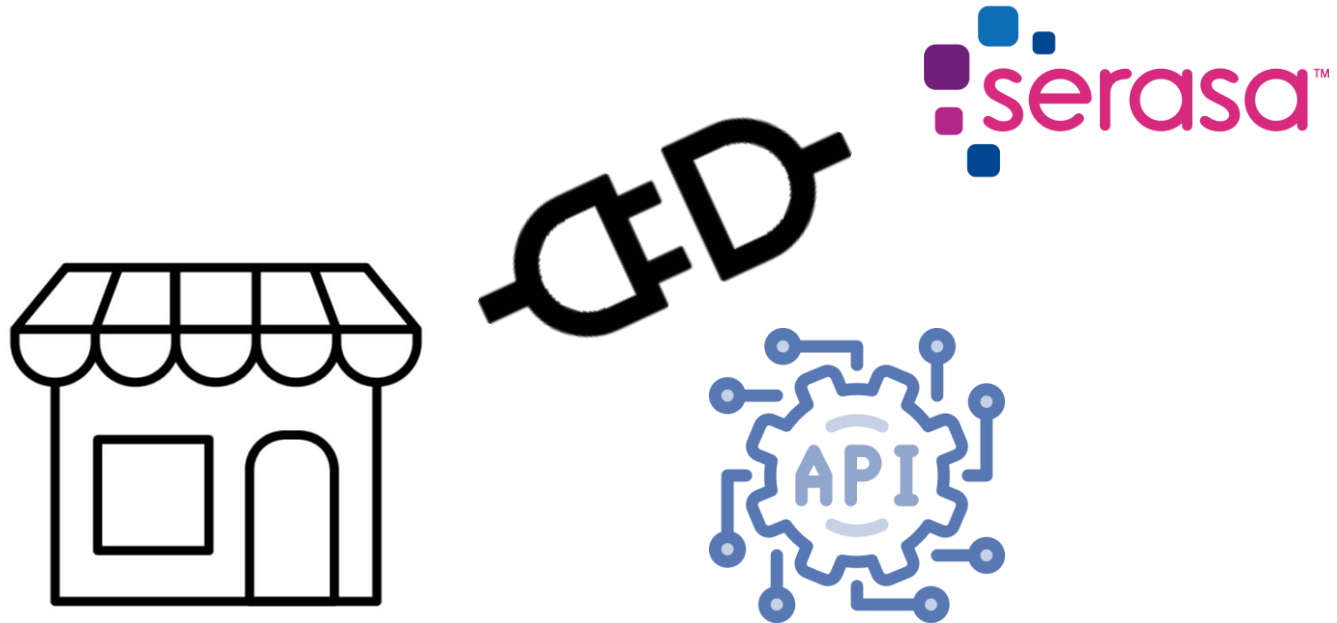


Como essa integração funcionaria?



# INTEGRAÇÃO DE SERVIÇOS EXTERNOS

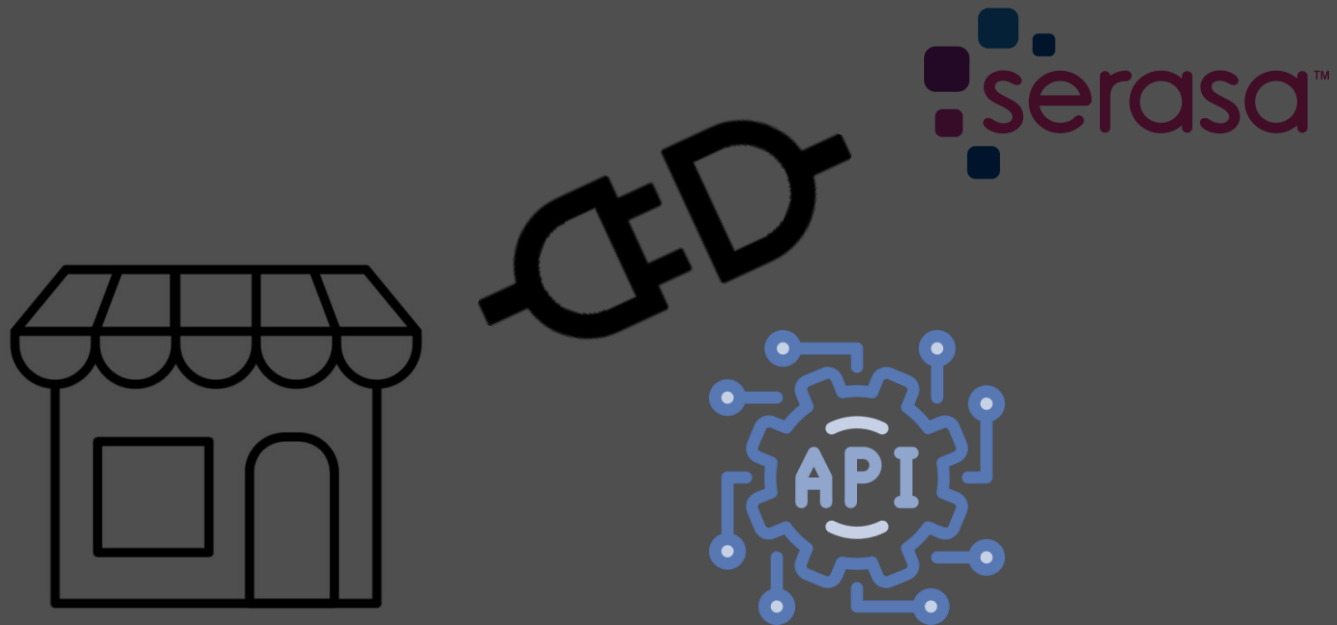
Exemplo:  
Loja com Sistema de Vendas



Como essa integração funcionaria?

# API ?

Exemplo:  
Loja com Sistema de Vendas



Como essa integração funcionaria?

**O QUE É API?**

# API

## Application Programming Interface

- Interface de Programação de Aplicação

Conjunto de normas que possibilita a comunicação entre plataformas

- Através de protocolos e padrões

**Situação: “Meu back precisa se comunicar com o meu front!” - API nele!**





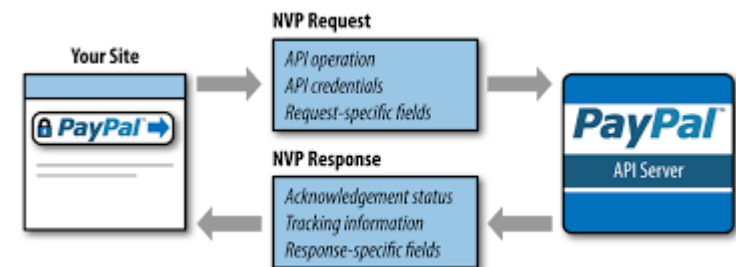
# API

As APIs são projetadas para permitir que um software acesse e utilize funcionalidades ou dados de outro software, serviço ou sistema, tornando a integração e a interação entre aplicativos mais eficientes e eficazes.

**Exemplos de APIs  
e como elas são usadas:**

## APIs de Pagamento:

Exemplo: O **PayPal** oferece uma API que permite que os comerciantes aceitem pagamentos on-line em seus sites ou aplicativos. Os desenvolvedores podem integrar essa API para processar transações financeiras.



# API

**As APIs são projetadas para permitir que um software acesse e utilize funcionalidades ou dados de outro software, serviço ou sistema, tornando a integração e a interação entre aplicativos mais eficientes e eficazes.**

## Exemplos de APIs e como elas são usadas:

## APIs de Redes Sociais:

Exemplo: A **API do Facebook Graph** permite que aplicativos acessem dados do Facebook e interajam com a plataforma, como postar em um perfil ou recuperar informações de um usuário.

# Graph API



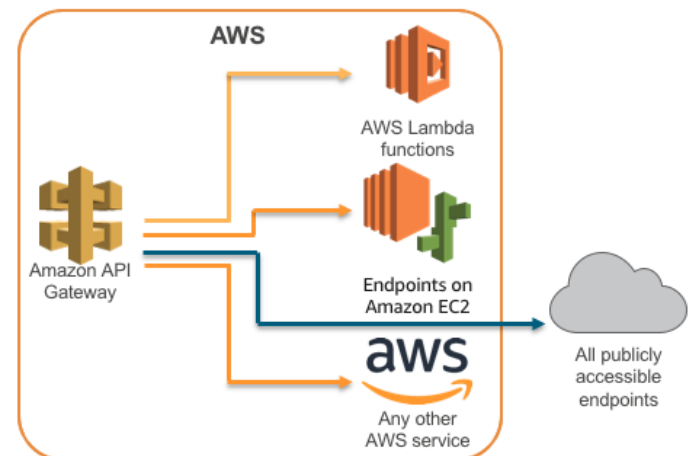
# API

As APIs são projetadas para permitir que um software acesse e utilize funcionalidades ou dados de outro software, serviço ou sistema, tornando a integração e a interação entre aplicativos mais eficientes e eficazes.

Exemplos de APIs  
e como elas são usadas:

## APIs de Serviços em Nuvem:

Exemplo: A **Amazon Web Services (AWS)** oferece uma série de APIs para serviços em nuvem, como armazenamento de dados, computação em nuvem e aprendizado de máquina. Os desenvolvedores podem usar essas APIs para criar aplicativos escaláveis e seguros.



# API

As APIs são projetadas para permitir que um software acesse e utilize funcionalidades ou dados de outro software, serviço ou sistema, tornando a integração e a interação entre aplicativos mais eficientes e eficazes.

**Exemplos de APIs  
e como elas são usadas:**

## **APIs de Bancos de Dados:**

Exemplo: O **JDBC** (Java Database Connectivity) é uma API que permite que aplicativos Java se conectem a bancos de dados, executem consultas e atualizem registros.





# API

As APIs são projetadas para permitir que um software acesse e utilize funcionalidades ou dados de outro software, serviço ou sistema, tornando a integração e a interação entre aplicativos mais eficientes e eficazes.

**Exemplos de APIs  
e como elas são usadas:**

## **APIs de Inteligência Artificial (IA):**

Exemplo: A API do **Google Cloud Vision** permite que aplicativos analisem e interpretem imagens, identificando objetos, rostos e texto em fotos.



# API

As APIs são projetadas para permitir que um software acesse e utilize funcionalidades ou dados de outro software, serviço ou sistema, tornando a integração e a interação entre aplicativos mais eficientes e eficazes.

**Exemplos de APIs  
e como elas são usadas:**

## APIs de Dispositivos e Hardware:

Exemplo: O **Android** fornece APIs para interagir com sensores de dispositivos, como GPS, câmera e acelerômetro, permitindo o desenvolvimento de aplicativos que tiram proveito desses recursos.



# API

As APIs são projetadas para permitir que um software acesse e utilize funcionalidades ou dados de outro software, serviço ou sistema, tornando a integração e a interação entre aplicativos mais eficientes e eficazes.

**Exemplos de APIs  
e como elas são usadas:**

## **APIs de Mapas e Geolocalização:**

Exemplo: A API do **Google Maps** permite que aplicativos incorporem mapas, direções e informações de localização em seus serviços. Empresas de entrega de alimentos usam essa API para exibir mapas e rastrear entregas em tempo real.



**HTTP ?**

# HTTP

O HTTP opera na camada de aplicação do modelo OSI, que é a camada mais alta do modelo. No modelo TCP/IP, o HTTP também é considerado um protocolo de camada de aplicação e funciona em conjunto com os protocolos das camadas inferiores, como TCP (Transmission Control Protocol) para controle de transmissão e IP (Internet Protocol) para roteamento e endereçamento de pacotes.



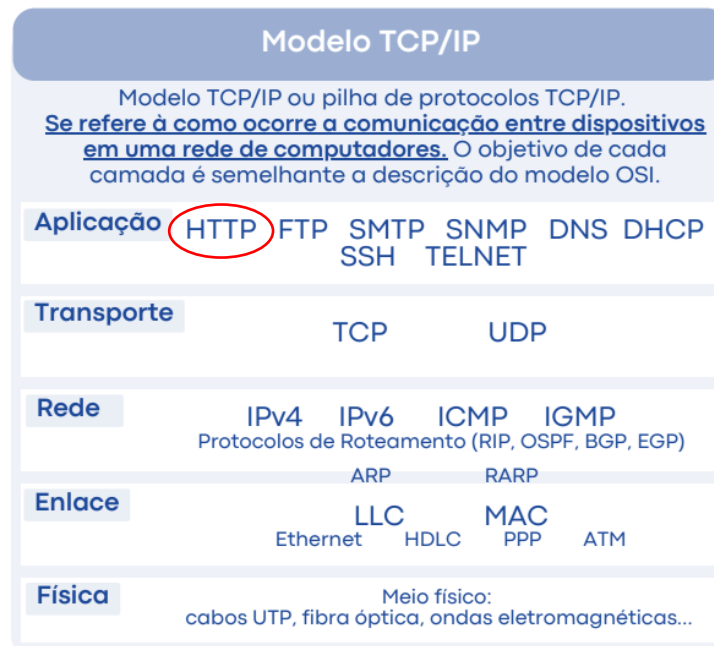


# HTTP

O protocolo **HTTP (Hypertext Transfer Protocol)** é um protocolo de comunicação que opera no topo da pilha de protocolos do modelo TCP/IP.



O modelo OSI (Open Systems Interconnection) é uma estrutura teórica que descreve como as redes de computadores devem funcionar, enquanto o modelo TCP/IP é uma implementação prática amplamente usada para redes e comunicações na Internet.



# HTTP

é amplamente utilizado para a transferência de hipertexto e comunicação na **World Wide Web**.



# HTTP

é amplamente utilizado para a transferência de hipertexto e comunicação na **World Wide Web**.

WWW é a sigla que em português significa "**rede de alcance mundial**".

É o sistema global de arquivos e documentos interligados que se tornou a interface mais comum da internet, acessível por meio de navegadores.

- ☐ Criação: Foi inventada em 1989 pelo cientista da computação britânico Tim Berners-Lee.
- ☐ Acesso: Você acessa a WWW digitando endereços web (URLs) no seu navegador, que então busca os servidores para encontrar e exibir as informações solicitadas.

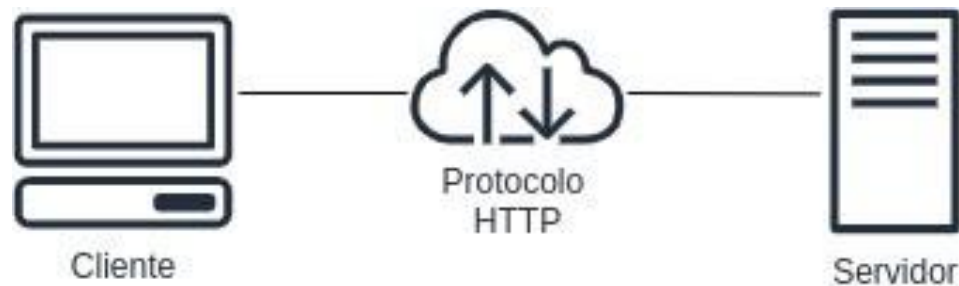
# HTTP

## Hypertext Transfer Protocol

- ❑ Camada de Aplicação modelo OSI e TCP/IP.
- ❑ REGRAS da comunicação entre o cliente (ex: Navegador) e um servidor na internet.

Requisição (request): Todo pedido que é enviado ao servidor.

Resposta (response): Resposta do servidor, seguindo o fluxo do request.



# HTTP

Na maioria dos casos, a integração de serviços é realizada por meio do protocolo HTTP, especialmente quando se trata da arquitetura de APIs. Isso se deve ao fato de o HTTP ser amplamente adotado e bem-suportado na web, tornando-o uma escolha comum para a comunicação entre aplicativos e serviços externos.



o HTTP é a base comum para a maioria das integrações de serviços externos por meio de APIs. No entanto, é importante observar que existem outros protocolos de comunicação, como o WebSocket e o gRPC, que também são usados em contextos específicos de integração, mas o HTTP ainda é a escolha predominante na integração de serviços na web.



# HTTP

**O HTTP é o alicerce da comunicação na World Wide Web. Ele é o protocolo que permite que os aplicativos se comuniquem na internet, incluindo a integração de serviços externos por meio de APIs**



# HTTP

WWW ou simplesmente Web, é um sistema global de documentos interconectados e recursos multimídia, acessíveis via Internet. Ela é uma das principais formas de acesso à informação na internet e é amplamente utilizada para compartilhar documentos, mídia e serviços online.



**O HTTP é o alicerce da comunicação na World Wide Web. Ele é o protocolo que permite que os aplicativos se comuniquem na internet, incluindo a integração de serviços externos por meio de APIs**



# HTTP

- ❑ A compreensão do HTTP é fundamental para a integração de serviços externos por meio de APIs, porque a maioria das APIs usa o HTTP como o meio de comunicação.
- ❑ Quando você faz uma solicitação a uma API, essa solicitação é enviada usando o protocolo HTTP. A resposta da API também é enviada de volta ao seu aplicativo usando o HTTP.
- ❑ Portanto, o HTTP é a linguagem comum que permite que aplicativos se conectem e troquem informações com serviços externos, tornando possível a integração e a comunicação na web.

**Cientes de que na maior parte dos casos os serviços são integrados através do protocolo HTTP segundo a arquitetura de apis...**

# INTEGRAÇÃO DE SERVIÇOS EXTERNOS



# INTEGRAÇÃO DE SERVIÇOS EXTERNOS

**Exemplo:**

## **Aplicativo de Previsão do Tempo**

Imagine que você está desenvolvendo um aplicativo de previsão do tempo para dispositivos móveis. Em vez de construir um sistema meteorológico complexo do zero, você pode integrar serviços externos para obter informações precisas e atualizadas sobre as condições climáticas.

**Como essa integração funcionaria?**





# INTEGRAÇÃO DE SERVIÇOS EXTERNOS

## Exemplo:

### Aplicativo de Previsão do Tempo

Imagine que você está desenvolvendo um aplicativo de previsão do tempo para dispositivos móveis. Em vez de construir um sistema meteorológico complexo do zero, você pode integrar serviços externos para obter informações precisas e atualizadas sobre as condições climáticas.

Como essa integração funcionaria?



# INTEGRAÇÃO DE SERVIÇOS EXTERNOS

## Exemplo:

### Aplicativo de Previsão do Tempo

**1. Fonte de Dados Meteorológicos Externa:** Você integra um serviço externo que fornece dados meteorológicos em tempo real. Este serviço pode ser uma API oferecida por uma empresa especializada em meteorologia, como o Weather.com ou o OpenWeatherMap.

**2. Solicitação de Dados:** No seu aplicativo, quando os usuários desejam verificar a previsão do tempo para uma localização específica, o aplicativo envia uma solicitação para o serviço externo. A solicitação pode incluir a localização (por exemplo, coordenadas geográficas ou nome da cidade) e as informações desejadas, como temperatura, condições atuais e previsões futuras.



# INTEGRAÇÃO DE SERVIÇOS EXTERNOS

## Exemplo:

### Aplicativo de Previsão do Tempo

**3. Processamento dos Dados Recebidos:** O serviço externo retorna os dados meteorológicos em um formato estruturado, como JSON ou XML. O seu aplicativo analisa esses dados e extrai as informações relevantes.

**4. Apresentação dos Dados:** O aplicativo exibe as informações da previsão do tempo para o usuário de maneira amigável, como gráficos, ícones, texto e tabelas.



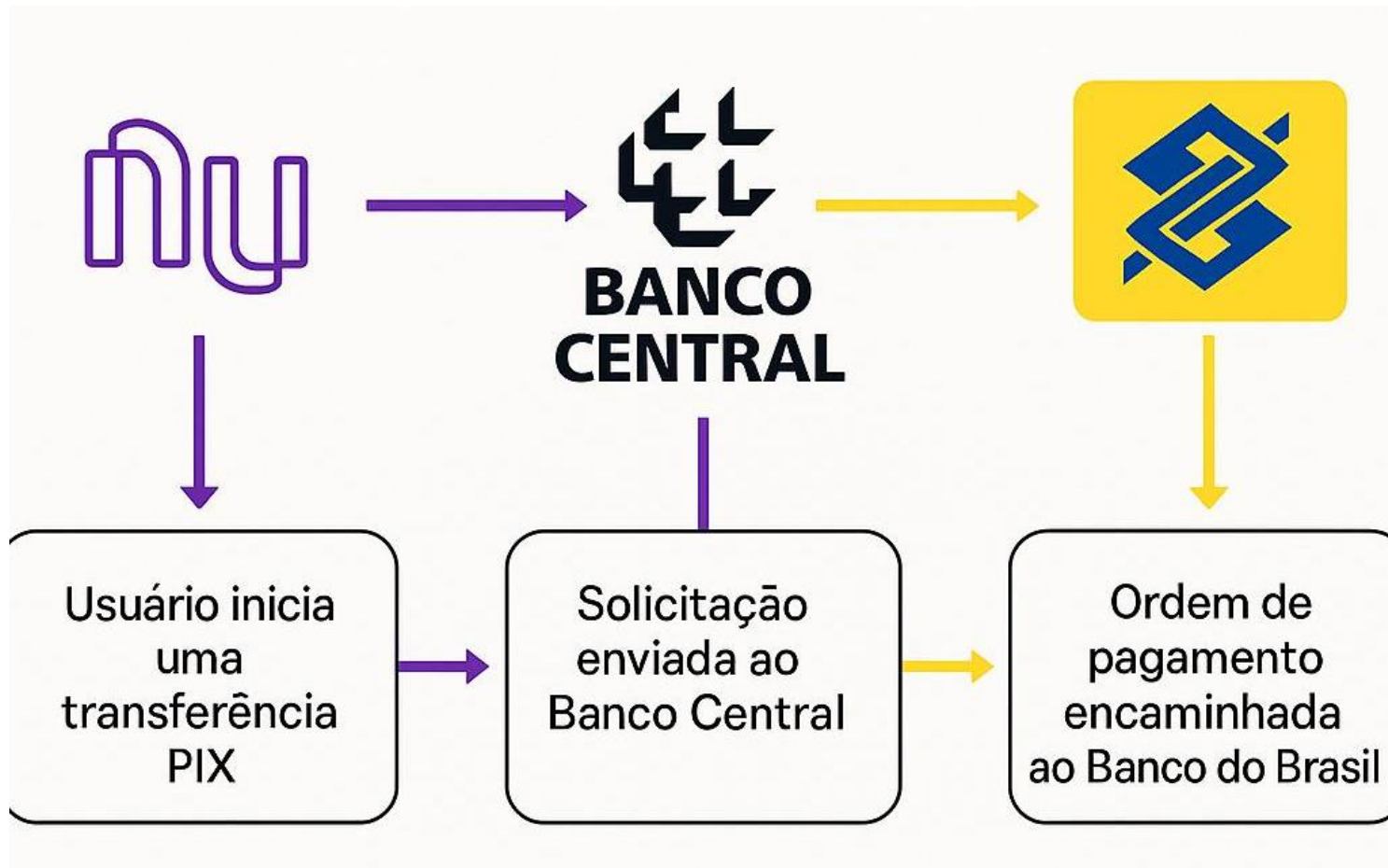
Este exemplo ilustra como a integração de serviços externos pode ser benéfica para desenvolvedores, permitindo que eles aproveitem os recursos e dados especializados de terceiros para enriquecer seus aplicativos e fornecer funcionalidades adicionais aos usuários.

# INTEGRAÇÃO DE SERVIÇOS EXTERNOS

**Exemplo: PIX**

# INTEGRAÇÃO DE SERVIÇOS EXTERNOS

## Exemplo: PIX





# INTEGRAÇÃO DE SERVIÇOS EXTERNOS

## Como funciona o PIX entre bancos diferentes?



**1. O cliente faz um PIX no app do Nubank**

Abre o app, escolhe a chave, digita o valor e... *enviar!*  
(App do Nubank)



**2. O Nubank manda o pedido pro Banco Central**

Quem cuida disso é o SPI (Sistema de Pagamentos Instantâneos).

(Enviando pro „coração’ do PIX)



**3. O Banco Central confere tudo**

Verifica chave, saldo e se está tudo certo.

(Tipo um juiz; „tá liberado!”)



**4. Ele encaminha o valor pro Banco do Brasil**

Agora o Banco do Brasil recebe o valor.

(Chegando no destino!)



**5. O dinheiro cai na conta do destinatário**

# INTEGRAÇÃO DE SERVIÇOS EXTERNOS

## IMPORTÂNCIA E VANTAGENS

**Ampliação de Funcionalidades:** Integrar serviços externos permite que você adicione funcionalidades que, de outra forma, exigiriam muito tempo e esforço para desenvolver internamente. Por exemplo, ao integrar um serviço de pagamento, seu aplicativo pode aceitar pagamentos com cartão de crédito sem que você precise criar seu próprio sistema de processamento de pagamentos.

**Reutilização de Recursos:** A integração de serviços externos permite que você aproveite os recursos já existentes. Em vez de criar e manter todos os recursos internamente, você pode usar serviços de terceiros que são especializados em determinadas áreas. Por exemplo, um aplicativo de mapeamento pode aproveitar os serviços de mapas do Google para exibir mapas e obter direções.

**Economia de Tempo e Recursos:** Desenvolver e manter todos os recursos internamente pode ser demorado e caro. A integração de serviços externos permite economizar tempo e recursos, acelerando o desenvolvimento do aplicativo.

# INTEGRAÇÃO DE SERVIÇOS EXTERNOS

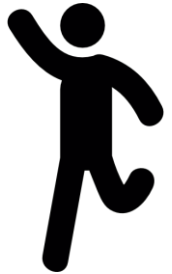
## IMPORTÂNCIA E VANTAGENS

**Foco no Core Business:** Ao delegar certas funcionalidades a serviços externos, você pode se concentrar no núcleo do seu negócio, ou seja, naquilo que torna seu aplicativo único e valioso para os usuários.

**Escalabilidade e Atualizações:** Os serviços externos são frequentemente escaláveis e atualizados regularmente pelos provedores, o que significa que seu aplicativo pode se beneficiar de melhorias contínuas e acomodar um número crescente de usuários.

# RESUMO

**Relevância da RC para devs.**  
**Integração de Serviços Externos em Aplicativos**  
**APIs Webs**  
**HTTP**  
**Importância e Vantagens da Integração de Serviços**



# ATIVIDADE

**Entregar no link a seguir, um cenário imaginário de uma comunicação HTTP integrando os serviços, diferente dos exemplos vistos em aula.**



# INTEGRAÇÃO DE SERVIÇOS EXTERNOS

## DESAFIOS DE SEGURANÇA E PRIVACIDADE

é importante considerar os desafios relacionados à segurança e privacidade ao integrar serviços externos.

Os desenvolvedores precisam garantir que os dados dos usuários sejam protegidos e que os serviços externos sejam seguros e conformes com as regulamentações de privacidade.



**HTTPS (Hypertext Transfer Protocol Secure)** está intimamente relacionado a questões de segurança e privacidade ao integrar serviços externos.

O HTTPS é uma extensão segura do HTTP, e seu uso é crucial para proteger a integridade e a confidencialidade das informações transferidas pela World Wide Web.

# seminário

## **SEGURANÇA EM REDE BÁSICA**

1. Malware,
2. Phishing,
3. Ataques de Força Bruta,
4. Ataques de Negação de Serviço (DoS),
5. Sniffing de Rede,
6. Engenharia Social,

## **SEGURANÇA EM REDE BÁSICA - temas**

- 1 Malware
- 2 Engenharia Social & Phishing
- 3 Ataques de Força Bruta
- 4 Ataques de Negação de Serviço (DoS)
- 5 Sniffing de Rede e Interceptação
- 6 Noções de Criptografia (tipos) e Hashing
- 7 Normas e Regulamentações



# **seminário**

## **SEGURANÇA EM REDE BÁSICA - grupos**

- 1. João Gabriel, Rafael, Dante**
- 2. Luca, Luan, Celso**
- 3. Ana Clara, Daniel, Guilherme**
- 4. Marcos, Gabriel, Layson**
- 5. Samuel, Otavio e Atari**
- 6. Danilo, Gabriel, Acassio**
- 7. Helena, Clovis, Eduarda**

# seminário

## SEGURANÇA EM REDE BÁSICA

### Instruções Gerais

**Formato:** Apresentação em grupo (slides, demonstrações, etc.) com duração de **15 a 20 minutos**.

**Entrega:** Devem ser entregues o **documento de slides** e um **pequeno relatório** (2-3 páginas) resumindo os pontos chave e as referências utilizadas.

**Avaliação:** Será baseada na **profundidade do conteúdo**, na **clareza da apresentação**, na **qualidade das recomendações de mitigação** e na **participação** de todos os membros do grupo.

# seminário

## SEGURANÇA EM REDE BÁSICA

### Instruções Gerais – (Grupos 1 a 5)

Cada grupo deve estruturar sua apresentação (15-20 minutos) e relatório cobrindo os seguintes pontos, com foco em uma rede básica:

- **Conceito e Variações:** Definição clara da ameaça e suas principais formas (ex: *Malware inclui Vírus, Worms, e Ransomware*).
- **Mecanismo de Ação (Como Funciona):** Detalhar o *modus operandi* da ameaça. Explicar tecnicamente como ela atinge um sistema ou serviço de rede.
- **Impacto:** Quais são os riscos e danos potenciais (perda de dados, indisponibilidade, prejuízo financeiro, etc.).
- **Mitigação:** Apresentar **pelo menos três medidas práticas e essenciais** que podem ser implementadas em uma rede básica para prevenir ou responder ao ataque (Ex: uso de ferramentas, políticas, ou configurações específicas).

# seminário

## SEGURANÇA EM REDE BÁSICA

### Instruções Gerais – (Grupo 6)

O grupo deve estruturar sua apresentação (15-20 minutos) e relatório cobrindo os seguintes pontos, com foco em uma rede básica:

#### **Noções de Criptografia (tipos) e Hashing**

O papel da criptografia (confidencialidade) e do hashing (integridade) na segurança de rede. Explicar a diferença entre criptografia simétrica e assimétrica, etc.

Este grupo não precisa focar em um ataque, mas sim na **defesa**. Deve explicar *como* essas técnicas são usadas para proteger as comunicações (Ex: SSL/TLS) e o armazenamento de senhas.

# seminário

## SEGURANÇA EM REDE BÁSICA

### Instruções Gerais – (Grupo 7)

O grupo deve estruturar sua apresentação (15-20 minutos) e relatório cobrindo os seguintes pontos, com foco em uma rede básica:

#### **Normas e Regulamentações**

As **leis e normas** que obrigam as empresas a protegerem suas redes. Abordar brevemente a **LGPD** (Lei Geral de Proteção de Dados) e normas como a **ISO 27001** e demais no contexto de redes básicas. (ISO/IEC 27002, recomendações NIST (National Institute of Standards and Technology, GDPR (General Data Protection Regulation), PCI DSS (Payment Card Industry Data Security Standard), etc.)