

# Resumão de API, REST, WebServices e Conceitos Relacionados

---

## ◆ API (Application Programming Interface)

---

- É um **conjunto de regras e definições** que permite que diferentes sistemas se comuniquem.
  - Exemplo: Um app de clima acessa dados de uma API meteorológica.
- 

## ◆ REST / RESTful

---

- **REST (Representational State Transfer):**
    - Estilo de arquitetura para criar APIs.
    - Baseado em recursos (ex: `/usuarios`, `/produtos`).
    - Usa **HTTP** para comunicação.
  - **RESTful:**
    - API que **segue fielmente os princípios REST**.
    - Características:
      - Sem estado (stateless).
      - Usa URLs para identificar recursos.
      - Métodos HTTP padronizados.
      - Respostas em formatos leves (JSON, XML).
- 

## ◆ WebServices

---

- Serviço disponível pela web para troca de informações entre sistemas.
  - Tipos:
    - **SOAP** → Mais antigo, usa XML e padrões rígidos.
    - **RESTful** → Mais moderno, leve, rápido, geralmente com JSON.
- 

## ◆ HTTP (HyperText Transfer Protocol)

---

- Protocolo de comunicação entre cliente e servidor.
  - Funciona por **requisições e respostas**.
  - Exemplo: navegador (cliente) pede uma página ao servidor.
- 

## ◆ Métodos HTTP (utilizados em APIs RESTful)

---

- **GET** → Buscar dados.

- **POST** → Criar novo recurso.
  - **PUT** → Atualizar recurso existente (inteiro).
  - **PATCH** → Atualizar parcialmente.
  - **DELETE** → Remover recurso.
  - **OPTIONS** → Informações sobre os métodos suportados.
  - **HEAD** → Igual ao GET, mas retorna só cabeçalhos.
- 

## ◆ HTTP x HTTPS

---

- **HTTP**: Comunicação sem criptografia.
  - **HTTPS**: Comunicação **segura** (usa TLS/SSL).
    - Garante **confidencialidade, integridade e autenticação**.
- 

## ◆ Listeners

---

- **Padrão de projeto** usado para "escutar" eventos e reagir.
  - Muito usado em interfaces gráficas e Android.
  - Exemplo: `onClickListener` reage ao clique de um botão.
- 

## ◆ Content Providers (Nativos no Android)

---

- Componente Android para compartilhar dados entre apps.
  - Exemplo:
    - Contatos → `ContactsContract`
    - Galeria → `MediaStore`
- 

## ◆ Services Providers Personalizados

---

- Criados pelo desenvolvedor para fornecer dados ou serviços.
  - Exemplo:
    - Um app que disponibiliza seus dados para outros apps via Content Provider próprio.
- 

## ◆ Carregamento de Dados em Segundo Plano

---

- Evita travar a interface do usuário.
- Técnicas/recursos:
  - **Threads** → Execução paralela.
  - **AsyncTask (obsoleto, mas clássico)**.
  - **WorkManager** → Tarefas agendadas.

- **Services** → Executam operações de longa duração.
  - **Coroutines (Kotlin)** → Maneira moderna de lidar com tarefas assíncronas.
- 

## Resumindo

---

- **API:** Interface para comunicação.
- **REST/RESTful:** Arquitetura leve baseada em HTTP.
- **WebServices:** Serviços acessados pela web.
- **HTTP/HTTPS:** Protocolo de comunicação (seguro ou não).
- **Métodos HTTP:** CRUD na web.
- **Listeners:** Escutam eventos.
- **Content Providers:** Compartilham dados no Android.
- **Services Providers Personalizados:** Extensão criada pelo dev.
- **Carregamento em segundo plano:** Mantém app fluido.