

# Sistemas Multimídia

## Aula 8: Sinalização, SIP e WebRTC

Professor(a)

IFSC - São José  
Departamento de Telecomunicações

Semestre 2026.1

# Objetivos da Aula

## O que vamos aprender hoje?

- Compreender o papel da sinalização
- Entender o protocolo SIP
- Explorar DNS, HTTP e sinalização
- Compreender arquitetura cliente-servidor
- Introduzir WebRTC para jogos
- Entender fluxo de jogo multiplayer WebRTC

## Aviso Importante

Esta aula é teórica! Nas próximas, partiremos para a prática.

# O que é Sinalização?

## Definição

- **Sinalização:** Protocolo para estabelecer, controlar e finalizar sessões
- **Exemplos:** SIP, H.323, WebRTC signaling
- **Papel:** Localizar usuários, negociar codecs, estabelecer sessões

## Analogia

Como fazer uma chamada telefônica:

- 1 Discar o número (localizar)
- 2 Aguardar atendimento (negociar)
- 3 Conversar (sessão estabelecida)
- 4 Desligar (finalizar)

# O que é Sinalização?

**Sinalização**  
Protocolo de  
Controle  
Multimídia

## Três itens básicos:

- Localizar pares
- Negociar codecs
- Estabelecer sessão

# Desafios da Comunicação Direta

## Problemas na comunicação P2P

- Endereços IP dinâmicos
- NAT (Network Address Translation)
- Firewalls
- Políticas de operadoras
- Endereços IPv4 esgotados

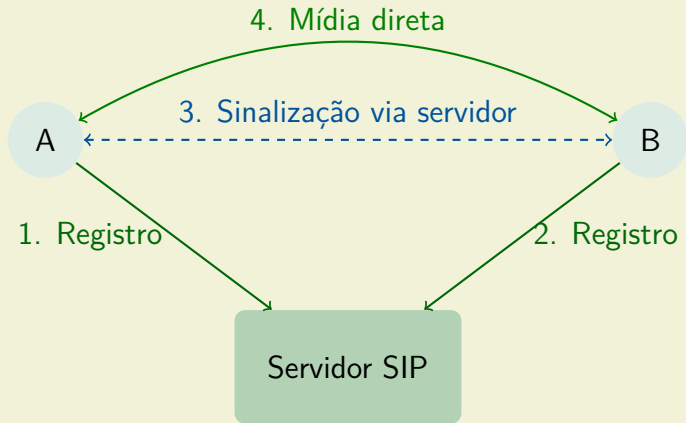
## Realidade Atual

- A internet é otimizada para **cliente-servidor**
- Comunicação direta é **difícil de implementar**
- Muitas aplicações usam intermediários

## Exemplo Prático

- BitTorrent: usa trackers (servidores)
- Jogos online: servidores de matchmaking
- VoIP: servidores SIP

# Solução: Servidor de Sinalização



Endereço fixo conhecido

# Solução: Servidor de Sinalização

## Papel do Servidor

- **Ponto de ancoragem:** Endereço conhecido e fixo
- **Registro:** Usuários se registram
- **Localização:** Permite encontrar outros usuários
- **Roteamento inicial:** Encaminha solicitações

# Protocolo SIP - Session Initiation Protocol

## Características do SIP

- Protocolo de sinalização para sessões multimídia
- Similar ao HTTP (textual, baseado em requisições/respostas)
- Usa URLs no formato: `sip:usuario@dominio`
- Exemplo: `sip:joao@ifsc.edu.br`

## Componentes SIP

- **User Agents:** Clientes (softphones, aplicativos)
- **Servidores:**
  - Registrar: Gerencia registros
  - Proxy: Roteia mensagens
  - Redirect: Redireciona



# Protocolo SIP - Session Initiation Protocol

**URI SIP:** sip:aluno@ifsc.edu.br

- Protocolo: sip
- Usuário: aluno
- Domínio: ifsc.edu.br

**Protocolo SIP**  
Sinalização  
Multimídia

# DNS - O Diretório da Internet

## Papel do DNS na Sinalização

- Tradicional: Traduz nomes para IPs (registros A, AAAA)
- Para sinalização: **Registros SRV**
- **SRV**: Localiza serviços específicos
- Exemplo: SIP sobre TCP no domínio ifsc.edu.br

## Importância

- Permite descoberta de serviços
- Não precisa saber IP do servidor
- Basta conhecer o domínio

# DNS - O Diretório da Internet

## Consulta DNS SRV

```
$ dig _sip._tcp.ifsc.edu.br SRV
```

Resposta:

```
_sip._tcp.ifsc.edu.br. 300 IN SRV
```

```
10 60 5060 sip01.ifsc.edu.br.
```

# HTTP/HTTPS e WebSockets

## HTTP/HTTPS - Base da Web Moderna

- **Cliente-servidor** por natureza
- **HTTPS**: Segurança (obrigatório para WebRTC)
- **Certificados**: Necessários para nomes válidos
- **Restrição**: Não pode usar IP direto

## Problema: Comunicação Bidirecional

HTTP é unidirecional (cliente → servidor)

- Como o servidor notifica o cliente?
- Solução: **WebSockets**

# HTTP/HTTPS e WebSockets

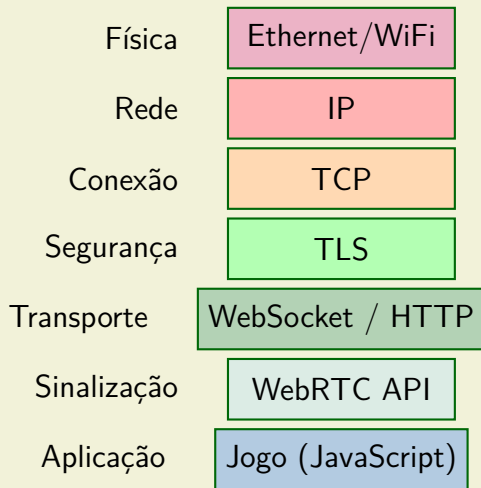
## WebSocket

- Protocolo sobre HTTP
- Cria canal bidirecional
- Mantém conexão aberta
- Permite comunicação em tempo real

### Fluxo:

- 1 Cliente faz upgrade HTTP
- 2 Conexão persiste
- 3 Comunicação bidirecional

# Pilha de Protocolos para Jogo WebRTC



# Pilha de Protocolos para Jogo WebRTC

## Papel de cada camada

- **WebSocket**: Sinalização (presença, convites)
- **WebRTC**: Mídia (áudio, vídeo, dados)
- **TLS**: Segurança (obrigatório!)
- **TCP/HTTP**: Transporte da sinalização

# Fluxo do Jogo WebRTC - Passo a Passo

## Passo 1: Download do Jogo

- 1 Cliente acessa URL
- 2 Baixa HTML/CSS/JS
- 3 Jogo carrega no navegador

## Passo 2: WebSocket

- 4 Cliente pede upgrade para WS
- 5 Servidor responde 101 Switching
- 6 Canal bidirecional estabelecido

## Passo 3: Registro

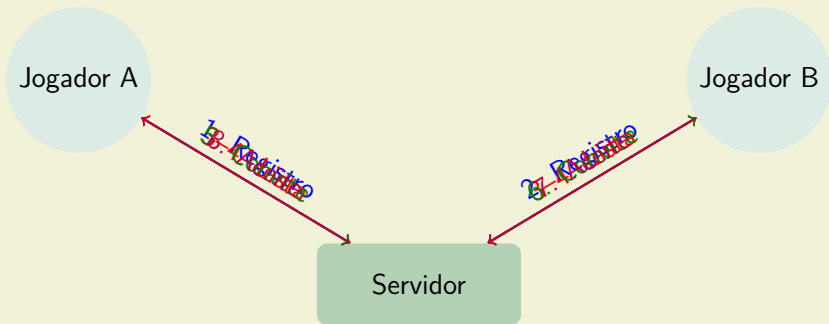
- 7 Cliente envia registro via WS
- 8 Servidor armazena usuário
- 9 Atualiza lista de presença

## Keep-Alive

- Mensagens periódicas
- Mantém conexão aberta
- Atualiza estado (online/offline)



# Fluxo do Jogo WebRTC – Conectando Jogadores



# Fluxo do Jogo WebRTC - Conectando Jogadores

## Publicação/Inscrição (Pub/Sub)

- Jogador publica sua presença
- Jogadores se inscrevem para lista atualizada
- Servidor notifica todos sobre mudanças

# Negociação de Mídia com WebRTC

## O que WebRTC faz por nós

- **NAT Traversal:** STUN, TURN, ICE
- **Negociação de Codecs:** VP8, H.264, Opus
- **Estabelecimento de Sessão:** SDP
- **Transmissão de Mídia:** SRTP
- **Canal de Dados:** DataChannel

## Sinalização $\neq$ WebRTC

- WebRTC **não especifica** sinalização
- Precisamos implementar nosso protocolo
- Usamos WebSocket para isso

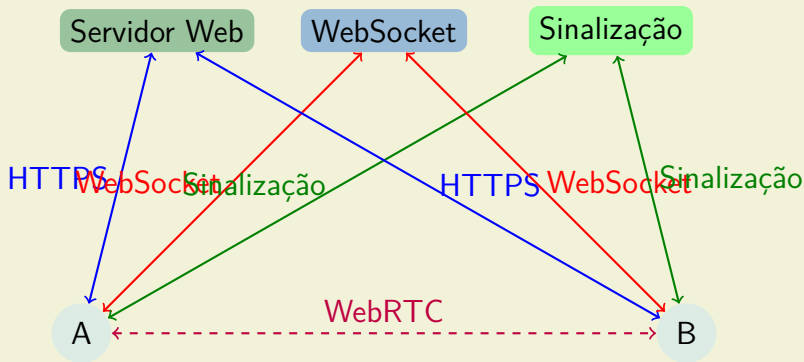
# Negociação de Mídia com WebRTC

## Fluxo Oferta/Resposta

- 1 **Oferta:** SDP com capacidades
- 2 **Resposta:** SDP com aceitação
- 3 **Candidatos ICE:** Endereços
- 4 **Conectado:** Mídia flui

**Vantagem:** Navegador implementa complexidade de mídia

# Arquitetura do Sistema Completo



## Pode ser uma única aplicação

- Servidor único com múltiplas funcionalidades
- Mais simples para implementar inicialmente

# Opções de Implementação - Decisões de Projeto

## Autenticação e Identificação

- **Nomes fixos:** usuario@ifsc.edu.br
  - Vantagem: Reconhecível
  - Desafio: Autenticação necessária
- **Nomes dinâmicos:** Jogador-ABC123
  - Vantagem: Simples
  - Desafio: Não reconhecível

## Importante!

- HTTPS é obrigatório
- Certificado válido necessário
- Não pode usar IP local
- Nome de domínio necessário

# Perguntas?

Obrigado pela atenção!