TechSolutions OS Summit 2026

Cenário

Vocês são desenvolvedores da TechSolutions, uma consultoria de tecnologia que acaba de fechar um contrato para modernizar a infraestrutura de uma fintech. Para garantir que o software seja eficiente, seguro e esteja em conformidade com as leis globais, a diretoria convocou um simpósio técnico. Cada equipe deverá pesquisar e apresentar um pilar crítico dos Sistemas Operacionais (SO) modernos, conectando a teoria com os desafios práticos de 2026, como Inteligência Artificial e Resiliência Digital.

Desafio (Entrega)

As equipes devem realizar uma pesquisa profunda e preparar:

- Apresentação (20 a 25 min): Uso de multimídia e demonstrações práticas (se possível).
- Documentação Formal:
    - Relatório técnico: pesquisa detalhada em formatação técnica (Introdução, Desenvolvimento, Conclusão, Referências)
    - Material multimídia: Slides, vídeos curtos ou mapas mentais.
    - Atividade Prática Rápida: Um desafio de 5 minutos para a turma (ex: um comando CLI, uma pergunta polêmica ou um pequeno quiz).

Os temas são:

1. Sistemas de Arquivos e Gestão de Dados
2. Particionamento, RAID e Unidades Lógicas
3. Gerenciamento de Memória e Performance
4. Arquiteturas de Processamento, Escalonamento e Concorrência
5. Segurança em Sistemas Operacionais (Hardening)
6. Sistemas Operacionais Embarcados e IoT
7. Segurança Cibernética, Redes e VPN
8. Legislação e Ética no Mundo Digital

A seguir, os temas serão detalhados para orientar a pesquisa e a apresentação de cada equipe. Lembrem-se de conectar os conceitos técnicos com casos reais do mercado e de preparar uma atividade prática que envolva a turma. Seu seminário deverá ser uma aula completa e interativa sobre o tema escolhido, mostrando domínio do assunto e capacidade de comunicação.

Serão avaliadas a autogestão no tempo de fala e a autonomia na busca por referências complementares. Usem ferramentas como o **Canva** ou **Prezi** para os slides e procurem exemplos reais de empresas que sofreram com os temas abordados.

Tema 1: Sistemas de Arquivos e Gestão de Dados

Este tema deixa de ser apenas "onde guardamos arquivos" para se tornar o estudo de como o software interage com a persistência de dados, garantindo integridade, velocidade e segurança.

1. Subtópicos para Pesquisa (O "O que pesquisar")

Para que a pesquisa seja completa, a equipe deve explorar estes quatro pilares:

A. Estrutura Interna e Arquitetura

- **Metadados e Inodes:** O que o SO guarda além dos dados (permissões, datas, proprietário).
- **Tabelas de Alocação:** Como o SO localiza um arquivo no disco físico (FAT, MFT).
- **Fragmentação:** Por que ocorre, como o SO lida com ela e por que SSDs não precisam de "desfragmentação" como os HDDs.

B. Sistemas Modernos e Comparativos

- **NTFS (Windows):** Recursos de compressão, cotas de disco e permissões avançadas.
- **EXT4 e Btrfs (Linux):** Por que o Linux é a escolha para servidores? Estudo sobre *Copy-on-Write* (CoW).
- **APFS (macOS/iOS):** Foco em criptografia nativa e otimização para memória Flash.

C. Confiabilidade e Integridade

- **Journaling (Diário):** Como o sistema evita a corrupção de dados após um desligamento repentino.
- **Técnicas de Snapshot:** Criar "fotos" do sistema de arquivos para recuperação imediata (essencial para servidores e backups).

D. Perspectiva do Desenvolvedor (O Diferencial)

- **I/O de Arquivos:** Como a escolha do sistema de arquivos impacta a velocidade de um Banco de Dados.
- **Abstração e VFS (Virtual File System):** Como o SO permite que seu código Python/Java leia um arquivo sem saber se ele está num HD, num SSD ou num pendrive.

FAQ (Guia de Estudo)

A equipe deve estar preparada para responder estas perguntas durante ou após o seminário:

1. **"Por que não consigo copiar um arquivo de 5GB para um pendrive formatado em FAT32, se ele tem 16GB livres?"**

- *Foco:* Limites técnicos de endereçamento (4GB por arquivo no FAT32).

2. **"O que é o 'Journaling' e como ele atua como uma 'caixa preta' de avião para o meu disco rígido?"**

- *Foco:* Integridade e recuperação de falhas.

3. **"Em um ambiente de servidores (Cloud), por que usamos sistemas de arquivos distribuídos como o NFS ou HDFS em vez de um sistema local simples?"**

- *Foco:* Escalabilidade e disponibilidade de dados.

4. **"Como o sistema de arquivos do Linux gerencia permissões (chmod/chown) e por que isso é vital para a segurança de uma aplicação web?"**

- *Foco:* Segurança da Informação e controle de acesso.

5. **"O que é o comando TRIM em SSDs e por que ele é gerenciado pelo Sistema de Arquivos?"**

- *Foco:* Ciclo de vida do hardware e performance.

Atividade Prática Sugerida

Para cumprir a **Capacidade Básica 7** (Operar SO por meio de linha de comando), a equipe pode demonstrar:

- **No Linux:** Usar o comando `df -h` e `mount` para mostrar onde os discos estão montados e qual o tipo de sistema de arquivos de cada partição.
- **No Windows:** Usar o comando `chkdsk` ou demonstrar a verificação de propriedades de uma unidade (verificando se é NTFS ou exFAT).

Relevância para o Mercado

Um desenvolvedor que entende sistemas de arquivos sabe configurar melhor um servidor Docker, entende por que um banco de dados está lento devido ao *overhead* de I/O e consegue diagnosticar problemas de permissão de escrita em ambientes de produção (Deploy).

Tema 2: Particionamento, RAID e Unidades Lógicas

Este tema foca em como "fatiamos" o hardware para que o Sistema Operacional possa organizar dados de forma estruturada e segura.

1. Subtópicos para Pesquisa (O "O que pesquisar")

A equipe deve aprofundar-se nos seguintes pilares técnicos:

A. Tabelas de Partição: O Mapa do Disco

- **MBR (Master Boot Record):** O padrão clássico. Limitações de 2TB e o máximo de 4 partições primárias. Por que está a cair em desuso?
- **GPT (GUID Partition Table):** O padrão moderno associado ao UEFI. Vantagens em redundância (cópia do cabeçalho no fim do disco) e suporte a discos gigantescos.
- **BIOS vs. UEFI:** Como o SO comunica com a placa-mãe para encontrar a partição de arranque (boot).

B. RAID (Redundant Array of Independent Disks)

- **RAID 0 (Striping):** Foco em performance pura. O que acontece se um disco falha?
- **RAID 1 (Mirroring):** Espelhamento para segurança. Qual o custo de armazenamento disso?
- **RAID 5 e 6:** O equilíbrio entre paridade, performance e segurança. Como o SO reconstrói dados de um disco morto?
- **RAID 10 (1+0):** A combinação de velocidade e redundância usada em bases de dados críticas.

C. Gerenciamento de Volumes Lógicos (LVM)

- **Abstração do Disco:** Diferença entre partição física e volume lógico.
- **Flexibilidade:** Como aumentar o tamanho de uma "unidade" (ex: o `/home` no Linux) em tempo real, adicionando um novo disco físico sem reinstalar o SO.

D. Armazenamento Definido por Software (SDS)

- Como as grandes clouds (AWS, Azure) gerenciam "discos virtuais" que não existem fisicamente como uma peça única, mas sim como pedaços de vários servidores.

FAQ (Guia de Estudo)

A equipe deve estar preparada para responder estas questões:

1. **"Se eu tenho dois SSDs de 500GB em RAID 0, o Windows verá 1TB ou 500GB? E se um queimar, perco tudo?"**

- *Foco:* Performance vs. Risco.

2. **"Por que é comum em servidores Linux separar as partições `/` (raiz), `/var` e `/home` em vez de deixar tudo num 'C:' único?"**

- *Foco:* Segurança, organização e prevenção de travamento do sistema por disco cheio.

3. **"O que é o 'Hot Swap' e como o RAID permite trocar um disco rígido sem desligar o servidor?"**

- *Foco:* Disponibilidade de sistemas (Uptime).

4. **"Posso converter um disco MBR para GPT sem formatar e perder meus dados?"**

- *Foco:* Utilitários de disco e riscos de conversão.

5. **"Qual a diferença entre RAID por Hardware (placa controladora) e RAID por Software (gerido pelo SO)?"**

- *Foco:* Custo, processamento e portabilidade.

Atividade Prática Sugerida

Para trabalhar as **Capacidades Básicas 2 e 3** (Configurar SO considerando disco e unidades), a equipe pode:

- **No Windows:** Abrir o "Gestor de Discos" (`diskmgmt.msc`) e mostrar como criar uma nova partição ou explicar a partição reservada pelo sistema.
- **No Linux:** Utilizar o comando `lsblk` para listar a árvore de dispositivos e `fdisk -l` para mostrar a tabela de partições.
- **Simulação:** Usar um simulador de RAID online ou mostrar um pequeno vídeo de uma reconstrução de RAID após falha.

Relevância para o Mercado

Um desenvolvedor de sistemas precisa saber que, se a sua aplicação escreve muitos logs, ela pode encher a partição do sistema e derrubar o serviço. Entender RAID e LVM permite que ele sugira arquiteturas de infraestrutura que suportem falhas de hardware sem que o código dele pare de funcionar.

Tema 3: Gerenciamento de Memória e Performance

Neste tema, os alunos descobrem que a memória RAM não é apenas um "depósito", mas um recurso escasso que o SO precisa distribuir com extrema precisão.

1. Subtópicos para Pesquisa (O "O que pesquisar")

A. Hierarquia e Organização

- **A Pirâmide de Memória:** Registradores, Cache (L1, L2, L3), Memória RAM e Memória Secundária. Por que existe essa diferença de velocidade e custo?
- **Alocação Contígua vs. Espalhada:** Como o SO decide onde colocar um novo programa que acabou de ser aberto.

B. Memória Virtual: O Truque de Mestre do SO

- **Paginação e Segmentação:** Como o SO faz o programa "acreditar" que tem mais memória do que realmente existe na RAM física.
- **Unidade de Gerenciamento de Memória (MMU):** O componente de hardware que traduz endereços lógicos em físicos.
- **Swap (Arquivo de Paginação):** O uso do disco como extensão da RAM. Quando isso é bom e quando destrói a performance?

C. Algoritmos de Substituição de Páginas

- Quando a RAM lota, quem sai? Estudo dos algoritmos: **FIFO** (Primeiro a entrar...), **LRU** (Menos usado recentemente) e **LFU**.
- **Thrashing (Hiperpaginação):** O colapso do sistema quando ele gasta mais tempo trocando dados com o disco do que processando.

D. Perspectiva do Desenvolvedor: Stack vs. Heap

- **Stack (Pilha):** Memória rápida, organizada e automática (variáveis locais).
- **Heap (Monte):** Memória dinâmica, onde o desenvolvedor pede espaço.
- **Garbage Collection:** Como linguagens modernas (Java, C#, Python) ajudam o SO a limpar a memória "suja" que o programa não usa mais.

FAQ (Guia de Estudo)

A equipe deve estar preparada para responder estas questões técnicas:

1. **"Por que meu computador fica extremamente lento quando abro 50 abas no navegador, mesmo se eu tiver um processador muito rápido?"**

- *Foco:* Esgotamento da RAM física e início do uso excessivo de Swap (disco).

2. **"O que é um 'Memory Leak' (Vazamento de Memória) e como ele pode derrubar um servidor ao longo de alguns dias?"**

- *Foco:* Falha do software em devolver a memória para o SO.

3. **"Qual a diferença entre um erro de 'Stack Overflow' e um erro de falta de memória (Out of Memory)?"**

- *Foco:* Diferença entre o limite da pilha de execução e o limite total do sistema.

4. **"Como o SO impede que o Processo A consiga ler os dados (como senhas) que estão na memória do Processo B?"**

- *Foco:* Proteção e isolamento de memória.

5. **"Vale a pena desativar o arquivo de paginação (Swap) se eu tiver muita memória RAM (ex: 64GB)?"**

- *Foco:* Discussão técnica sobre como o Windows/Linux gerenciam o Kernel.

Atividade Prática Sugerida

Para trabalhar a **Capacidade Básica 3** (Configurar SO considerando memória), a equipe pode:

- **Monitoramento Real:** Abrir o "Monitor de Recursos" no Windows ou o comando `htop` / `free -m` no Linux e explicar o que é memória "Cacheada", "Disponível" e "Em Uso".
- **Demonstração de Stress:** Usar uma pequena ferramenta ou script que consome RAM progressivamente para mostrar o gráfico de uso subindo e o impacto na resposta do sistema.

Relevância para o Mercado

Um desenvolvedor que não entende gerenciamento de memória cria códigos que consomem recursos desnecessários. No mundo da **Cloud computing**, onde você paga pelo uso de memória, entender esse tema economiza milhares de reais para as empresas.

Tema 4: Arquiteturas de Processamento, Escalonamento e Concorrência

Este tema aborda como o SO serve de ponte entre diferentes tecnologias de hardware e como ele faz a "mágica" de rodar centenas de tarefas simultâneas num processador que só faz uma coisa por vez.

1. Subtópicos para Pesquisa (O "O que pesquisar")

A. Arquiteturas de Processadores (Hardware)

- **x86 vs. x64:** A transição dos 32 para os 64 bits. O que mudou no endereçamento e nos registadores?
- **ARM:** Por que é a arquitetura dominante em smartphones e agora em laptops (Apple M1/M2/M3)? Estudo sobre eficiência energética e arquitetura RISC.
- **CISC (Complex Instruction Set) vs. RISC (Reduced Instruction Set):** Como a filosofia de design do chip afeta a velocidade do SO.
- **Arquiteturas de Servidores e Cloud:** Introdução a tecnologias como **EPYC** (AMD) e processadores baseados em **RISC-V** (o futuro do hardware aberto).

B. O Escalonador de Processos (Scheduler)

- **Estados do Processo:** Pronto, Executando e Bloqueado (Espera).
- **Algoritmos de Escalonamento:** * **Round Robin:** O sistema de fatias de tempo (Time Slice).
- **Prioridade:** Como o SO decide que o antivírus é menos importante que a interface do utilizador.
- **Escalonamento em Multicore:** Como o SO distribui tarefas entre 8, 16 ou mais núcleos de CPU.

C. Multitarefa e Concorrência

- **Processos vs. Threads:** A diferença entre isolamento total e partilha de memória.
- **Context Switch (Troca de Contexto):** O custo de performance quando o SO troca de uma tarefa para outra.
- **Sincronização:** Por que precisamos de **Mutex** e **Semáforos** para evitar que dois processos alterem o mesmo dado ao mesmo tempo?

FAQ (Guia de Estudo)

A equipe deve dominar estas questões para o seminário:

1. **"Um software compilado para Windows x64 roda num Windows ARM (como o do Surface Pro)? Porquê?"**

- *Foco:* Incompatibilidade de conjunto de instruções e emulação.

2. **"O que acontece se dois 'threads' tentarem incrementar a mesma variável ao mesmo tempo? Como o SO resolve isso?"**

- *Foco:* Condição de Corrida (Race Condition) e exclusão mútua.

3. **"Por que os computadores modernos têm tantos 'núcleos' em vez de apenas um processador muito, muito rápido?"**

- *Foco:* Limites térmicos (calor) e paralelismo.

4. **"O que é um 'Processo Zombie' e por que ele ainda aparece na lista de tarefas se já terminou?"**

- *Foco:* Comunicação entre processo pai e filho e libertação de recursos.

5. **"Qual a diferença entre Multiprogramação e Multiprocessamento?"**

- *Foco:* Execução alternada (perceção de simultaneidade) vs. Execução paralela real.

Atividade Prática Sugerida

Para trabalhar a **Capacidade Básica 7** (Operar SO por linha de comando) e **Conhecimento 2.2** (Threads), a equipe pode:

- **Exploração de CPU:** No Windows, usar o "Gestor de Tarefas" (separador Desempenho > CPU) para mostrar os núcleos lógicos. No Linux, usar o comando `lscpu` para detalhar a arquitetura.
- **Visualização de Processos:** Usar o comando `top` ou `htop` no Linux para mostrar a coluna de "Prioridade" (PR) e "Nice" (NI), explicando como o SO altera a prioridade dos processos.
- **Demonstração de Threads:** Mostrar um pequeno código em Python ou C que cria múltiplos threads e observar o uso da CPU subir no monitor de recursos.

Relevância para o Mercado

Para um programador, entender arquiteturas é vital para o **Deploy**. Saber se a aplicação vai rodar num servidor Intel ou num servidor ARM (mais barato na AWS/Azure) pode reduzir custos de infraestrutura em 40%. Além disso, entender concorrência evita bugs críticos em sistemas que atendem muitos utilizadores simultâneos.

Tema 5: Segurança em Sistemas Operacionais (Hardening)

Neste tema, o foco é a segurança **interna** e estrutural do SO. É o momento de entender como o sistema se autoprotege e como o administrador "tranca as portas" contra invasores.

1. Subtópicos para Pesquisa (O "O que pesquisar")

A. Controle de Acesso e Permissões

- **Matriz de Acesso:** Como o SO decide quem pode ler, escrever ou executar um arquivo.
- **Princípio do Menor Privilégio:** Por que nunca devemos rodar aplicações do dia a dia como "Administrador" ou "Root".
- **ACLs (Access Control Lists):** Diferença entre as permissões básicas do Linux (rwx) e as permissões avançadas do Windows (NTFS).

B. Sistemas Operacionais de Segurança (Hardened OS)

- **SELinux e AppArmor:** Como o Linux usa "rótulos" para impedir que um serviço web acesse pastas que não deveria (Mesmo que o serviço seja hackeado).
- **Sistemas Focados em Segurança:** Breve estudo sobre o **OpenBSD** (famoso pelo foco em código seguro) e o **Tails** (foco em amnésia e privacidade).

C. Isolamento e Sandbox

- **Conceito de Sandbox:** Como o SO isola um processo para que, se ele falhar ou for infectado, não comprometa o restante do sistema.
- **Conteinerização (Introdução ao Docker):** Como os containers aproveitam o kernel do SO para criar isolamento leve.

D. Hardening (Fortalecimento do Sistema)

- **Redução da Superfície de Ataque:** Desativação de serviços inúteis, fechamento de portas lógicas e remoção de softwares redundantes.
- **Atualizações de Kernel:** A importância dos patches de segurança (Ex: Entender o que foi o ataque *WannaCry* e como o patch do SO teria evitado).

FAQ (Guia de Estudo)

A equipe deve estar preparada para debater estas situações:

1. **"Qual a diferença real entre um usuário comum, um Administrador (Windows) e um Superusuário/Root (Linux)?"**

- *Foco:* Hierarquia de poder e riscos associados.

2. **"Se um vírus infecta um programa que está rodando dentro de um 'Sandbox', ele consegue chegar aos meus arquivos pessoais?"**

- *Foco:* Eficácia do isolamento de processos.

3. **"Por que dizem que o Linux é 'mais seguro' que o Windows? Isso é um fato técnico ou apenas um mito?"**

- *Foco:* Modelo de permissões nato vs. popularidade e engenharia social.

4. **"O que é 'Exploit' e como o SO usa técnicas como ASLR (Address Space Layout Randomization) para confundir hackers?"**

- *Foco:* Tecnologias modernas de proteção de memória contra invasões.

5. **"Por que manter o sistema atualizado é mais importante do que ter um bom antivírus?"**

- *Foco:* Correção de vulnerabilidades de "Dia Zero" (Zero-day).

Atividade Prática Sugerida

Para trabalhar a **Capacidade Básica 3** (Configurar SO considerando usuários e permissões), a equipe pode:

- **No Linux:** Demonstrar o uso do comando `chmod` e `chown`. Mostrar o que acontece quando um usuário sem permissão tenta ler o arquivo `/etc/shadow`.
- **No Windows:** Mostrar como criar um usuário "Padrão" (sem poderes de Admin) e como o UAC (User Account Control) atua como uma barreira de segurança.
- **Auditoria:** Mostrar como visualizar os logs de segurança (Visualizador de Eventos no Windows ou `/var/log/auth.log` no Linux) para ver tentativas de login falhas.

Relevância para o Mercado

Um desenvolvedor que ignora segurança cria sistemas vulneráveis. Entender as permissões do SO é essencial para configurar servidores web, APIs e bancos de dados. No mercado atual, a filosofia **DevSecOps** exige que o desenvolvedor saiba proteger o ambiente onde seu código será executado.

Tema 6: Sistemas Operacionais Embarcados e IoT

Este tema explora sistemas projetados para tarefas específicas, onde a eficiência de hardware e a velocidade de resposta são mais importantes do que a versatilidade.

1. Subtópicos para Pesquisa (O "O que pesquisar")

A. Sistemas de Tempo Real (RTOS)

- **Determinismo:** O que significa um sistema ser determinístico e por que isso é vital para um airbag ou um braço robótico?
- **Hard Real-Time vs. Soft Real-Time:** A diferença entre sistemas onde o atraso causa uma catástrofe e sistemas onde o atraso é apenas um incômodo (como streaming de vídeo).
- **Escalonamento em RTOS:** Como o SO garante que a tarefa mais importante nunca seja interrompida por uma menos crítica.

B. Sistemas Operacionais para IoT (Internet of Things)

- **Restrições de Recursos:** Como rodar um SO em dispositivos com apenas alguns Kilobytes de RAM.
- **Sistemas Leves:** Estudo sobre o **FreeRTOS**, **Contiki** ou **Zephyr**.
- **Gerenciamento de Energia:** Técnicas de *Duty Cycling* e *Deep Sleep* para fazer a bateria de um sensor durar anos.

C. Arquitetura e Kernel Embarcado

- **Microkernel vs. Monolítico:** Por que sistemas embarcados preferem arquiteturas mínimas e modulares?
- **Abstração de Hardware (HAL):** Como o SO embarcado conversa com sensores (temperatura, acelerômetro) e atuadores (motores).

D. Casos de Uso e Mercado

- **Automotivo:** O papel do SO em carros autônomos.
- **Wearables:** Como o SO de um Smartwatch difere do SO de um smartphone.
- **Drones e Robótica:** O uso do **ROS (Robot Operating System)** como uma camada sobre o SO.

FAQ (Guia de Estudo)

A equipe deve estar preparada para responder estas questões técnicas:

1. **"Por que não instalamos o Windows ou o Linux comum em um roteador de internet ou em uma geladeira inteligente?"**

- *Foco:* Overhead de recursos e necessidade de funções que esses aparelhos não usam.

2. **"O que acontece se o escalonador de um sistema de freio ABS falhar por 100 milissegundos?"**

- *Foco:* A importância da latência zero em sistemas críticos.

3. **"Qual a diferença entre um Microcontrolador (Arduino/ESP32) e um Microprocessador (Raspberry Pi) sob a ótica do Sistema Operacional?"**

- *Foco:* Bare-metal (sem SO) vs. Sistemas com Kernel.

4. **"Como o SO embarcado lida com a conectividade (Wi-Fi/Bluetooth) sem drenar toda a bateria do dispositivo?"**

- *Foco:* Pilhas de protocolos otimizadas.

5. **"Por que a segurança em dispositivos IoT é considerada um dos maiores desafios da tecnologia atual?"**

- *Foco:* Dispositivos simples que se tornam "portas de entrada" para ataques em redes corporativas.

Atividade Prática Sugerida

Para trabalhar as capacidades de **instalação e configuração**, a equipe pode:

- **Simulação:** Mostrar o funcionamento de um simulador de RTOS (como o do Wokwi para ESP32 rodando FreeRTOS) ou explicar como o **Raspberry Pi OS Lite** (versão sem interface gráfica) é usado em projetos IoT.
- **Demonstração de Monitoramento:** Mostrar como o consumo de memória de um SO embarcado é estável e minúsculo comparado a um SO de desktop.

Relevância para o Mercado

O mercado de IoT e sistemas embarcados está em explosão. Um desenvolvedor de sistemas que entende as limitações de um RTOS está pronto para trabalhar na **Indústria 4.0**, em empresas de logística, saúde (dispositivos médicos) e automação residencial. É uma das áreas que melhor remunera devido à sua alta especialização.

Tema 7: Segurança Cibernética, Redes e VPN

Este tema aborda a proteção dos dados enquanto eles "viajam" e como garantir que apenas as pessoas certas acedam aos recursos da empresa remotamente.

1. Subtópicos para Pesquisa (O "O que pesquisar")

A. Fundamentos de Redes e Protocolos Seguros

- **O Modelo OSI e a Segurança:** Em que camadas atuam o Firewall, o TLS e o IPsec?
- **Protocolos Seguros (HTTPS, SSH, SFTP):** Como a criptografia transforma o tráfego legível em dados cifrados.
- **Certificados Digitais e SSL/TLS:** Como o SO e o navegador verificam se um site é "quem diz ser".

B. VPN (Virtual Private Network)

- **Conceito de Túnel:** Como a VPN cria uma rede privada sobre uma infraestrutura pública (Internet).
- **Protocolos de VPN:** Estudo comparativo entre **OpenVPN**, **L2TP/IPsec** e o moderno **WireGuard** (focado em performance).
- **Casos de Uso:** Acesso remoto para funcionários (Remote Access) e ligação entre escritórios (Site-to-Site).

C. Firewalls e Perímetros de Defesa

- **Firewall de Host vs. Firewall de Rede:** A diferença entre o Windows Defender/Iptables e um equipamento dedicado (Fortigate, pfSense).
- **IDS (Detecção de Intrusão) e IPS (Prevenção):** Como o SO ou a rede podem identificar comportamentos suspeitos automaticamente.
- **DMZ (Zona Desmilitarizada):** Como isolar servidores públicos da rede interna da empresa.

D. Ameaças e Vetores de Ataque em Rede

- **Ataques de Interceptação (Man-in-the-Middle):** Como funcionam e como a criptografia os impede.
- **DDoS (Negação de Serviço):** Como o SO e a infraestrutura podem tentar mitigar ataques de sobrecarga.
- **Criptografia Simétrica vs. Assimétrica:** Entender como chaves públicas e privadas protegem as comunicações.

FAQ (Guia de Estudo)

A equipe deve ser capaz de explicar estes pontos à turma:

1. **"Se eu usar uma VPN, fico 100% anónimo na internet e protegido contra qualquer vírus?"**

- *Foco:* Desmitificar a VPN (ela protege o caminho, não necessariamente o conteúdo ou a identidade).

2. **"Qual a diferença prática entre o HTTP e o HTTPS no que toca ao que um 'hacker' pode ver numa rede Wi-Fi pública?"**

- *Foco:* Criptografia de transporte de dados.

3. **"O que é a Autenticação de Dois Fatores (2FA) e como o SO pode integrar isso para tornar o acesso mais seguro?"**

- *Foco:* Camadas extras de segurança além da palavra-passe.

4. **"Por que razão as empresas exigem o uso de VPN para aceder ao código-fonte ou às bases de dados quando o programador está em 'Home Office'?"**

- *Foco:* Extensão do perímetro de segurança da empresa.

5. **"O que acontece quando um certificado SSL de um site expira? Por que o SO nos impede de entrar no site?"**

- *Foco:* Cadeia de confiança e autoridade certificadora (CA).

Atividade Prática Sugerida

Para trabalhar as **Capacidades Básicas de Segurança**, a equipe pode demonstrar:

- **Análise de Certificado:** Abrir o cadeado no navegador num site conhecido e mostrar quem emitiu o certificado e a validade.
- **Verificação de Portas:** No Windows (PowerShell) ou Linux, usar o comando `netstat -an` ou `ss -tulpn` para mostrar que portas o SO tem "abertas" à espera de ligações de rede.
- **Demonstração de VPN:** Mostrar a mudança do endereço IP público antes e depois de ligar uma VPN (pode ser uma extensão gratuita ou um cliente de desktop).

Relevância para o Mercado

No cenário atual de **Trabalho Remoto**, saber configurar e utilizar VPNs é uma competência básica. Para um programador, entender como os dados viajam de forma segura entre o *Front-end* e o *Back-end* (via TLS/SSL) é crucial para evitar fugas de dados que podem custar milhões em multas e danos à reputação.

Tema 8: Legislação e Ética no Mundo Digital

Este tema aborda o enquadramento jurídico que regula a profissão de TI e o uso das tecnologias. A equipe deve apresentar estes tópicos de forma clara, pois trata-se do primeiro contacto de muitos alunos com o Direito Digital.

1. Subtópicos para Pesquisa (Roteiro Obrigatório)

A. Proteção de Dados: LGPD (Brasil) e GDPR (Europa)

- **Objetivos:** O que visam proteger e quem são os "titulares dos dados".
- **Conceitos Chave:** Dados pessoais vs. Dados sensíveis. O que é o consentimento?
- **Impacto no Desenvolvimento:** Como o *Privacy by Design* (privacidade desde a conceção) obriga o programador a pensar na segurança antes de escrever a primeira linha de código.
- **Penalidades:** As multas milionárias e o dano à reputação das empresas.

B. Marco Civil da Internet (Lei 12.965/2014)

- **Princípios:** Neutralidade da rede (por que o seu fornecedor de internet não pode privilegiar um site em detrimento de outro), liberdade de expressão e privacidade.
- **Responsabilidade:** Quando é que uma plataforma (como o Facebook ou um site que vocês criem) é responsável pelo conteúdo que os utilizadores publicam?
- **Influência Global:** Como esta lei brasileira serviu de modelo para outros países.

C. Marco Legal da Ciência, Tecnologia e Inovação

- **Fomento:** Como a lei facilita a cooperação entre universidades e empresas para criar novas tecnologias.
- **Desafios:** A burocracia vs. a agilidade necessária para o mercado de inovação.

D. Crimes Cibernéticos: Lei Carolina Dieckmann (Lei 12.737/2012)

- **Contexto:** O caso real que deu origem à lei e a necessidade de tipificar crimes digitais.
- **Condutas Criminosas:** O que constitui "invasão de dispositivo informático" e quais são as penas.
- **Impacto na Segurança:** Como a lei reforça a importância das permissões de acesso que estudámos no Tema 5.

E. Propriedade Intelectual e Direitos Autorais

- **Proteção de Software:** O software é protegido como "obra literária". O que significa o Direito de Autor no código-fonte?
- **Licenciamento:** Diferença entre Software Proprietário, Open Source (GPL, MIT) e o impacto da pirataria no mercado de trabalho de TI.

FAQ (Guia de Estudo)

A equipe deve dominar estas questões para responder aos colegas:

1. **"Se eu criar uma aplicação que guarda o e-mail e a palavra-passe dos utilizadores, quais são as minhas obrigações perante a LGPD?"**

- *Foco:* Armazenamento seguro, transparência e direito de exclusão.

2. **"O Marco Civil da Internet permite que o governo aceda às minhas conversas privadas sem ordem judicial?"**

- *Foco:* Proteção da privacidade e o papel do poder judiciário.

3. **"Se eu entrar no computador de um colega sem autorização, mas não apagar nada, estou a cometer um crime?"**

- *Foco:* Lei Carolina Dieckmann e a definição de invasão.

4. **"Posso copiar um trecho de código de um projeto no GitHub e usar na minha aplicação comercial? O que diz a lei de direitos autorais?"**

- *Foco:* Licenciamento de software e ética profissional.

5. **"Por que razão a GDPR europeia afeta empresas brasileiras que nunca saíram do Brasil?"**

- *Foco:* Extraterritorialidade e a economia digital globalizada.

Atividade Prática Sugerida

Para trabalhar as **Capacidades Socioemocionais** (Ética e Responsabilidade), a equipe pode:

- **Análise de Termos de Uso:** Escolher uma aplicação popular (ex: Instagram, TikTok ou uma IDE) e encontrar uma cláusula que fale sobre o uso de dados dos utilizadores.
- **Simulação de Caso:** Apresentar um cenário de "vazamento de dados" numa empresa fictícia e perguntar à turma: "De acordo com a LGPD, o que esta empresa deveria ter feito para evitar isto?".
- **Checklist de Conformidade:** Criar um panfleto digital simples com "5 passos para o meu software estar dentro da lei".

Relevância para o Mercado

Um Desenvolvedor de Sistemas que não conhece as leis digitais é um risco para a empresa. Em 2026, a conformidade legal é um requisito de contratação. Entender estes marcos legais permite que o profissional proteja a si mesmo, à sua empresa e, principalmente, aos utilizadores finais.