

Para resolução desse problema foi criada uma solução em 8 etapas descritas abaixo:

**1. Túnel SSH para acesso ao banco de dados do sistema de gestão hospitalar:**

Um túnel SSH é uma conexão segura estabelecida entre duas máquinas através de uma conexão SSH. Ele cria um canal criptografado que permite que o tráfego de dados seja transmitido de forma segura entre as máquinas, mesmo quando atravessa redes não confiáveis, como a Internet.

Para termos acesso à máquina do hospital temos que usar o seguinte comando em um sistema Unix-Like:

```
ssh -L <porta_local>:<endereço_banco_de_dados>:<porta_banco_de_dados>  
<usuário>@<máquina_do_hospital>
```

**2. API de exportação de dados no sistema de gestão hospitalar:**

Desenvolvimento de uma API que permita exportar os dados de pacientes de forma segura. Pode ser acessada internamente por meio do passo 1, fornecendo os dados em formato JSON. Esse passo pode ser implementado usando diversas tecnologias, porém a que possuo mais familiaridade é o Node.js utilizando o framework do Express.js.

**3. Código/Script de importação no sistema de análise e cirurgias:**

O terceiro passo consiste no desenvolvimento de um programa no sistema de análise de cirurgia que seja capaz de conectar a API de exportação através do túnel SSH e importar os dados, de forma eficiente, garantindo a completude e a integridade dos dados importados, para um bando de dados local. Além disso, é recomendado que o programa seja configurado para ser executado periodicamente de forma automática, para que os dados sejam atualizados regularmente.

**4. Criptografia de dados durante a transferência:**

Durante a criação do Script e da transferência de dados é preciso garantir que os dados transferidos entre os sistemas sejam criptografados para proteger a confidencialidades das informações. Isso pode ser usando protocolos de criptografia como SSL/TLS para comunicação segura através do túnel SSH

**5. Mecanismo de sincronização e controle de duplicação:**

Implementar um mecanismo no sistema de análise de cirurgias para sincronizar periodicamente os dados importados do sistema de gestão hospitalar e garantir que não haja duplicação de registros. Isso pode envolver o uso de identificadores exclusivos para cada registro e a comparação de dados para identificar e eliminar duplicatas.

**6. Monitoramento e registro de atividades:**

Para manutenção do sistema é necessário implementar ferramentas de monitoramento para rastrear o desempenho e a integridade da integração, bem como registrar todas as atividades relacionadas à transferência de dados entre os sistemas. Isso permite detectar e resolver rapidamente quaisquer problemas que possam surgir durante o processo de integração.

**7. Backup e recuperação de dados:**

Estabelecer políticas de backups regulares para os dados importados garantindo que os dados estejam disponíveis e protegidos contra perda e corrupção ou qualquer outro tipo de problema.

**8. Teste e validação:**

Realizar testes abrangentes para garantir que a integração funcione conforme o esperado e atenda aos requisitos de segurança e qualidade

**Requisitos Funcionais:**

Requisitos Funcionais	Detalhes e Requisitos
1. Estabelecer um túnel SSH seguro	- Configurar uma conexão SSH entre a máquina do hospital e o sistema de análise de cirurgias.
	- Utilizar autenticação de chave SSH ou outros métodos de autenticação segura.
	- Garantir que a conexão seja criptografada para proteger a segurança dos dados durante a transferência.
2. Desenvolver uma API de exportação de dados	- Criar uma API no sistema de gestão hospitalar que permita exportar os dados de pacientes e cirurgias.
	- A API deve fornecer os dados em formato estruturado, como JSON.
3. Criar um script de importação no sistema de análise de cirurgias	- Desenvolver um script ou programa no sistema de análise de cirurgias que seja capaz de conectar-se à API de exportação do sistema de gestão hospitalar.
	- O script deve ser capaz de importar os dados de pacientes e cirurgias de forma eficiente, garantindo a completude e integridade dos dados.
	- Configurar o script para execução periódica e automática, garantindo a atualização regular dos dados.
4. Criptografar os dados durante a transferência	- Utilizar protocolos de criptografia como SSL/TLS para garantir a segurança dos dados durante a transferência através do túnel SSH.
5. Implementar um mecanismo de sincronização	- Implementar um mecanismo no sistema de análise de cirurgias para sincronizar periodicamente os dados importados do sistema de gestão hospitalar.

Requisitos Funcionais	Detalhes e Requisitos
e controle de duplicação	- Utilizar identificadores exclusivos e comparar dados para evitar duplicações de registros.
6. Monitorar e registrar atividades	- Implementar ferramentas de monitoramento para rastrear o desempenho e a integridade da integração.
	- Registrar todas as atividades relacionadas à transferência de dados entre os sistemas para identificar e resolver problemas.
7. Realizar backups regulares dos dados	- Estabelecer políticas de backup regulares para garantir a disponibilidade e proteção dos dados contra perda ou corrupção.
8. Realizar testes abrangentes	- Realizar testes abrangentes para garantir que a integração funcione conforme o esperado e atenda aos requisitos de segurança e qualidade.

#### Requisitos não funcionais:

Requisitos Não Funcionais	Detalhes e Requisitos
1. Segurança:	- Garantir que todas as comunicações entre os sistemas, incluindo o túnel SSH e a transferência de dados, sejam criptografadas utilizando protocolos como SSL/TLS.
	- Implementar medidas de autenticação forte, como o uso de chaves SSH, para garantir a identidade dos sistemas envolvidos na integração.
2. Desempenho:	- Assegurar que a integração seja rápida e eficiente, com tempos de resposta aceitáveis mesmo para grandes volumes de dados.
	- O tempo de latência na transferência de dados através do túnel SSH deve ser minimizado para garantir uma comunicação ágil entre os sistemas.
3. Disponibilidade:	- Garantir que os sistemas estejam disponíveis para acesso e operação durante o horário de funcionamento, com um tempo de inatividade mínimo.
	- Implementar estratégias de redundância e failover para garantir a disponibilidade contínua dos serviços.
4. Confiabilidade:	- Assegurar que a integração seja robusta e confiável, minimizando falhas e interrupções no fluxo de dados.

Requisitos Não Funcionais	Detalhes e Requisitos
	- Realizar testes de estresse e de recuperação para garantir que o sistema seja capaz de lidar com situações adversas sem comprometer a integridade dos dados.
5. Manutenção:	- Facilitar a manutenção e atualização do sistema, com uma arquitetura modular e de fácil manutenção.
	- Prover documentação detalhada e atualizada sobre a integração e seus componentes para auxiliar na resolução de problemas e na implementação de melhorias.
6. Escalabilidade:	- Garantir que a solução seja capaz de lidar com um aumento no volume de dados e usuários sem comprometer o desempenho ou a disponibilidade.
	- Utilizar uma arquitetura escalável que permita adicionar recursos adicionais conforme necessário para suportar o crescimento do sistema.
7. Compatibilidade:	- Certificar-se de que a solução seja compatível com as tecnologias e plataformas existentes nos sistemas envolvidos na integração.
	- Garantir a interoperabilidade entre os diferentes sistemas e versões de software para evitar conflitos e incompatibilidades.

Além das tabelas, foi criado um diagrama de caso de uso para representar esse cenário:

