

Anotações dos comentários

- A ideia é ser algo simples sem utilizar nenhuma impresa real, apenas dados fictícios
- O armazenamento podemos inicialmente ter um "smartcontract" para salvar apenas os dados de uma consulta agendada, por exemplo
 - "André tem uma consulta às 9:00 com a senha 1 no laboratório XPTO"
 - "Rodrigo tem uma consulta às 9:30 com a senha 2 no laboratório XPTO"
 - "Thiago tem uma consulta às 10:00 com a senha 3 no laboratório XPTO" E os dados a serem salvos seria um hashmap<AccountID, Vec<{ticket:u32 agencia: AccountId, dataHoraConsulta:X, dataHoraDoTicket}>>
- Armazenamento dos dados poderíamos fazer uma "POC" que salva os arquivos no bucket AWS de forma local utilizando o LocalStack
- Front-end simple com HTML, CSS e JavaScript só para ter uma interação com a blockchain, sem nenhuma estilização muito complicada
- Registros médicos seriam os dados que salvaríamos on-chain, como o AccountID do médico, AccountId da agência, data da geração do registro, hash criptografado do arquivo salvo no bucket(se houver arquivo) e qualquer outro dado que seja necessário ou interessante de ser mantido
- Histórico de tickets salvos seria interessante para saber quando foi feita a consulta X na agência XPTO, por exemplo, o André fez a consulta X no dia 01/07/2010 às 10:30 com o ticket 15 na agência ImunoCorp(Nome aleatório aqui)
- O padrão FHIR é um padrão de salvamento e manipulação de dados médicos, ele é bem complexo e chato de se implementar, essa parte poderíamos ignorar nesse começo e implementar se realmente isso vier a se tornar uma aplicação no mundo real
- Os **record** são apenas uma sugestão da IA, vamos discutir os dados que deveria existir nele e ver quais deveríamos ter para uma primeira versão da aplicação, poderíamos ter apenas a parte de geração de ticket das consultas e histórico dele
- As **Regras de Negócio (Básicas)** precisamos discutir internamente o que precisamos fazer, eu deixaria essa parte de ver o que as empresas precisam para uma V2, a qual teria uma implementação no mundo real
- As senhas sequenciais, podemos ter a regra de negócio de gerar a senha a qualquer dia/hora mas o contador ser referente ao dia requisitado, no qual mudaríamos a estrutura do dado **{agency_code}-{YYYYMMDD}-{HHMM}-{seq:04d}-{YYYYMMDD}-{HHMM}** - Ex.: AG12-20251204-0932-0007-20251210-1030 (agência AG12, data de emissão 2025-12-04, hora de emissão 09:32, sequência 7, data de atendimento 2025-12-10, hora de atendimento 10:30)
- Na criptografia poderíamos ver de utilizar como salt os bites da data/hora que foi iniciado a geração o registro
- Os dados de auditoria eu vejo que seria básico e simples, com um evento do ink para quando o dado foi pesquisado, coisa bem simples e somente a emissão do evento, mas sim ele seria implementado futuramente
- A Governança vejo que teria uma conta Admin que controlaria a adição de pessoas, médicos e agências inicialmente, mas para cadastrar um registro no histórico do paciente o médico/agência faria uma "solicitação" para adicionar esses dados com a confirmação do usuário, ou o usuário teria de liberar o acesso à leitura e escrita no histórico dele para o médico/agência
- Geração do ticket concordo com o que está comentado
- Conflitos vejo que ficaria por conta do substrate mesmo

- Sobre o consenso podemos utilizar o PoA (Proof of Authority) que basicamente é um validador que o André realmente é o André e pronto, mais leve e mais rápido
- Sobre a parte de milestones vamos ignorar o que está escrito no PDF, mas vejo que utilizaremos o github para o versionamento e para o "deploy" seria tudo feito de forma manual e na máquina local mesmo, para termos um ambiente relevante funcional sem muito trabalho nesse começo, e se o projeto vier a virar uma solução real, ai sim teria que ter uma pipeline.