Documento de Sistema de software

Sistema para auto contenção, transporte e manutenção de dados médicos pessoais, com suporte por Blockchain e Polkadot.

Autores: André, Rodrigo, Thiago. Data: dezembro de 2025 – janeiro de 2026.

Tabela de revisões:

Status	Data	Autor
Formulação da primeira versão desse documento : V 1.0.	07/12/2025	Rodrigo
Melhorias de idéias após revisão	27/12/2025	André, Rodrigo,Thiago

Apelos do projeto:
- Dados sob soberania do próprio indivíduo.
- Dados médicos sob controle do paciente.
- **Portabilidade**, **confiabilidade**, **controle** e **autonomia** sobre dados pessoais de saúde.

Descrição do projeto.

Esse é um projeto de software para criar uma aplicação + infraestrutura capaz de armazenar dados médicos pessoais, de forma segura, confiável e editável somente através de consenso suportado por Blockchain Polkadot. Tal sistema proverá então a facilidade de salvar/guardar dados médicos pessoais, sem qualquer acesso de terceiros, a menos que isso seja consentido. E esse tipo de consentimento será dado somente pela pessoa dona dos dados, usando o artifício tecnológico de escrita e consulta em cadeia de blocos de informações provida pela Polkadot. Portanto, esse projeto proverá um mecanismo para as pessoas serem as únicas responsáveis e detentoras de seus próprios dados médicos e concentrados num banco de dados, sendo que a proteção dos dados (acesso e edição) estará garantida com o uso de *blockchain*. Grande parte dos dados, como laudos médicos, receitas médicas, diagnósticos e imagens de exames estarão em banco de dados (relacional ou NoSQL) obviamente, mas as possibilidades/permissões de criá-los, editá-los e mostrá-los serão regidas por software feito em Rust, que usará o Substrate Polkadot, para comprovar consentimentos necessários.

Dessa forma as pessoas poderão trocar de médicos, sempre que necessário e sem prejuízo de perdas de históricos médicos, porque as informações sobre a saúde delas estarão sempre armazenadas num banco de dados único e disponíveis a quem interessar, desde a primeira consulta médica de um paciente em sua vida, sob consentimento. Ou seja, um paciente “carregará” consigo o seu histórico médico e terá a confiança de que os dados não serão alterados ou vendidos por profissionais ilícitos. Porque o uso de *blockchain* garantirá que tais dados foram/serão acessados e editados somente em momentos adequados, com a ciência do próprio paciente (o verdadeiro dono dos dados). Isso significa que ao consultar um médico, o paciente precisará liberar o acesso do seu histórico ao médico, que poderá solicitar. Por outro lado o médico (ou qualquer profissional da saúde) estará confiante de que os dados são mesmo de quem declara ser dono.

Os dados serão alterados somente por profissionais da saúde, assim como já ocorre atualmente. Ou seja, qualquer profissional conseguirá alterar os dados médicos de um paciente somente mediante a prova de que o mesmo seja um profissional da saúde, perante o software a ser desenvolvido neste projeto. Esse tipo de prova também poderá ser suportado por *blockchain* e Polkadot, por exemplo, o que não é muito comum atualmente (2025), mas os detalhes desse mecanismo estão explicados na especificação dos requisitos desse projeto.

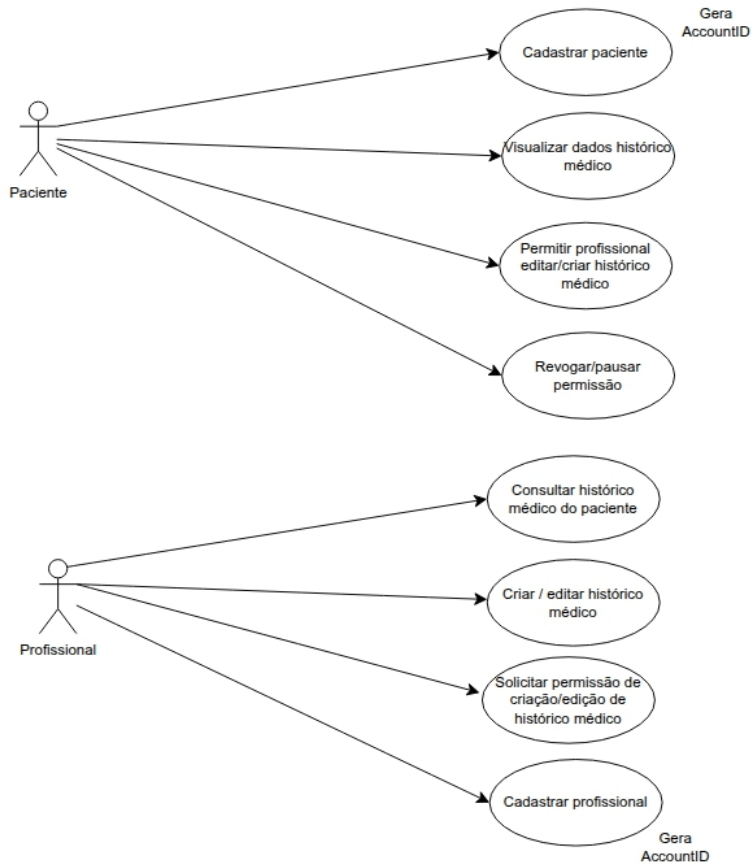
Por fim, o paciente confiará plenamente que seus dados estarão íntegros por todo tempo e o profissional da saúde confiará que os dados médicos do paciente são originados por outros profissionais e não o próprio paciente. Assim as duas partes (paciente e médico) confiarão em qualquer histórico médico. Então os históricos serão sempre válidos e protegidos.

Esse projeto adequa-se à LGPD porque provê:

- Consentimento explícito do titular
- Segurança e proteção dos dados sensíveis (saúde)
- Direito à portabilidade.
- Direito de revogação do consentimento.
- Rastreabilidade do uso dos dados.

Futuramente esse projeto poderá incluir a capacidade de gerar tickets numerados, por paciente, afim de organizar a ordem dessas pessoas em filas de atendimentos. O sistema garantirá a veracidade dos tickets e a validade de qualquer numeração estampada nos mesmos, bem como a relação deles com os pacientes.

Diagrama de Use cases.



Descrição das use cases.

No projeto, a *blockchain* não protege os dados médicos em si, mas protege as regras de acesso, as permissões concedidas pelo paciente e as provas de autoria e consentimento, tornando esses elementos imutáveis, auditáveis e independentes de uma autoridade central.

Cadastrar paciente:

Um paciente irá se cadastrar nesse sistema de software. Para o cadastro do paciente deverá ser fornecido algum documento valido emitido pelo governo, ex: CPF, e número de telefone (Inicialmente será contemplado somente o Brasil). O número de telefone será utilizado para o envio de SMS (ou MFA), para certificação da validade dos dados fornecidos.

Visualizar dados de histórico médico:

O paciente poderá visualizar seu histórico médico sempre que necessário. Atualmente as pessoas precisam guardar dados médicos impressos, enquanto mudam de médicos ao longo da vida. Essa é a forma comum de manterem consigo um histórico, até esse momento (2025). Os pacientes não carregarão consigo alguma *media* digital, para manter o banco de dados médicos, mas terão acesso a esse banco sempre que desejado. A visualização do histórico será via interface de software e o banco estará instalado em algum computador ainda a definir. Um paciente não conseguirá editar dados vistos por ele mesmo.

Permitir profissional criar ou apenas ver histórico médico:

Sempre que um profissional da saúde requerer acesso a dados médicos de um paciente, esse último deverá consentir isso. O consentimento será feito através do uso de chaves alfanuméricas criptográficas geradas/suportadas por código Rust. As mesmas serão validadas e relacionadas com o determinado paciente. Por exemplo, um *smart contract* pode verificar se uma chave de consenso é pertencente/relacionada a um paciente. Nesse caso o paciente teria que passar a chave ao profissional. Por outro lado, pode-se pensar numa função de *smart contract* que apenas libere acesso aos dados para um AccountID de um profissional específico. Ou seja, o paciente usa essa função e passa para ela o AccountID do médico. Isso poderia liberar o médico para acessar certos dados do paciente, talvez dados da mesma especialidade médica do profissional. Quando um profissional de saúde conseguir acesso ao histórico médico de uma pessoa, ele poderá editar os dados. Mas, dependendo do caso, o consentimento será apenas para leitura dos dados já existentes e nenhuma modificação será possível.

Revogar/Pausar permissão:

Cada paciente poderá revogar a permissão de acesso aos seus dados médicos, por meio do software programado em Rust no Polkadot SDK. A revogação ou pausa de permissão (ação alvo dessa *Use Case*) impedirá que um profissional da saúde altere o histórico médico do paciente.

Consultar histórico médico do paciente:

Um profissional da saúde fará consulta ao histórico médico de um paciente. Para tal ele precisará do consentimento do próprio paciente. O consentimento será garantido como explicado acima. O médico precisará fornecer sua identificação dentro desse sistema, que poderá ser o AccountId ou uma chave própria pública. Nesse caso de uso o médico poderá ler dados existentes do histórico, mas não poderá alterá-los.

Criar histórico médico:

Um profissional da saúde fará a adição dos dados no histórico do paciente. Se não houver dado no histórico (*blank*), ele escreverá os primeiros dados a pertencerem ao histórico. Para tudo isso, o profissional deverá conseguir o consentimento do paciente. Caso o paciente não esteja acessível

(por motivo de saúde, invalidez, acidente, etc.), o profissional da saúde usará seus sistemas tradicionais de software, para iniciar seu atendimento. O consentimento será dado conforme já comentado, também via software programado em Rust. Vide detalhes dos diagramas de sequência.

Solicitar permissão de criação de dados no histórico médico:
Qualquer profissional da saúde prestes a editar/criar dados de histórico médico de paciente fará a requisição do consentimento para isso. Para tal, médico e paciente trocarão suas chaves públicas via esse sistema. A camada em Polkadot será exatamente a parte desse projeto que garantirá a funcionalidade da seguraça necessárias para crição de consensos regidos por lógicas apoiadas em chaves.

Cadastrar profissional:
Permite um profissional da saúde cadastrar-se no sistema. O profissional citará dados de identidade, encontrados em seus documentos fornecidos pelo governo. Ex: CPF, número de identidade. E citará seu número de celular, para recebimento de SMS ou execução do algoritmo de MFA. Motivo: validar os dados citados. Ele também deverá citar seu número de registro em qualquer concelho que pertença, ou entidade criada para profissionais da sua área médica (CRM).

Todos os dados de histórico médico não serão gravados em blocos de informações numa *blockchain*. Eles serão gravados em banco de dados isolado da *chain*. Dados críticos como chaves privadas que identificam pessoas, códigos hashes para validar informações salvas em banco e chaves para encriptar/decriptar dados nunca estarão salvos em banco de dados. Essa medida aumentará a segurança em manipular tais dados. Os dispositivos dos usuários poderão conter chave privada. Se um dispositibo for atacado e uma chave privada for acessada, as outras chaves privadas dos outros usuários ainda estarão seguras. Ou seja, tem-se que evitar dados críticos concentrados num único banco de dados.

Justificativa do Uso de Blockchain no Sistema Proposto

Embora os dados médicos propriamente ditos não sejam armazenados diretamente em uma *blockchain*, o uso dessa tecnologia no sistema proposto é fundamental para garantir confiança, integridade, auditabilidade e descentralização no controle de acesso e na manipulação desses dados sensíveis. A *blockchain* atua como uma camada de governança e consenso, responsável por assegurar que regras, permissões e provas associadas ao uso dos dados médicos não possam ser fraudadas ou alteradas de forma indevida.

Regras Imutáveis de Acesso e Edição de Dados

No sistema proposto, uma das principais regras é que um profissional da saúde somente pode criar ou alterar dados médicos de um paciente mediante a existência de um consentimento válido, ativo e não expirado concedido pelo próprio paciente. Essa regra é implementada diretamente na lógica executada pela *blockchain* (*runtime* desenvolvido com *Substrate*), tornando-se imutável após sua publicação.

A utilização da *blockchain* garante que essa regra seja aplicada por consenso entre os nós da rede, impedindo que administradores do sistema, desenvolvedores ou operadores de infraestrutura alterem seu funcionamento de forma unilateral. Assim, elimina-se a necessidade de confiar em uma autoridade central para fazer cumprir as regras do sistema, reduzindo significativamente o risco de manipulação indevida de dados médicos.

Gerenciamento de Permissões Baseado em Consentimento

Outro aspecto central do projeto é o controle de permissões de acesso aos dados médicos, que são sempre concedidas pelo paciente. Cada permissão é registrada na blockchain por meio de uma transação que associa o paciente, o profissional da saúde, o tipo de acesso permitido (leitura ou edição) e um prazo de validade definido.

Esse mecanismo assegura que:

- √ Nenhum profissional da saúde possa acessar dados sem autorização explícita do paciente;
- √ Nenhuma permissão possa ser estendida, modificada ou reativada sem o consentimento do titular dos dados;
- √ O término ou a revogação de uma permissão seja automaticamente respeitado por todo o sistema.

A *blockchain* torna essas permissões auditáveis e invioláveis, garantindo que o histórico de autorizações concedidas e revogadas permaneça íntegro ao longo do tempo.

Provas de Autoria, Integridade e Consentimento

A *blockchain* também é utilizada como um mecanismo de prova. Sempre que um profissional da saúde cria ou altera um registro médico, um código *hash* representando esse conteúdo é registrado na *blockchain*, juntamente com a identificação criptográfica do profissional e o momento da operação.

Esse registro permite provar, de forma inequívoca:

Qual profissional foi responsável pela criação ou alteração de um dado médico;

Quando essa ação ocorreu;

Que havia um consentimento válido do paciente no momento da operação;

Que os dados armazenados fora da *blockchain* não foram alterados posteriormente, pois qualquer modificação seria detectada pela divergência do *hash*.

Dessa forma, a *blockchain* fornece garantias de integridade, autoria e não repúdio, características essenciais em sistemas que lidam com informações médicas sensíveis.

Detecção e Prevenção de Acessos Indevidos

Além de impedir ações não autorizadas, o sistema proposto também se beneficia da capacidade da *blockchain* de registrar tentativas de acesso inválidas. Caso um profissional tente acessar ou modificar dados após a expiração ou revogação de um consentimento, a transação será rejeitada automaticamente pela lógica do consenso, podendo ainda gerar registros auditáveis da tentativa (isso pode ser um requisito futuro).

Isso contribui para aumentar a transparência do sistema e fortalecer a confiança dos pacientes, que passam a ter garantias técnicas de que seus dados não podem ser utilizados fora das condições previamente autorizadas.

Síntese do Papel da Blockchain no Projeto

Em síntese, no sistema proposto, a blockchain não é utilizada como um repositório de dados médicos, mas como um mecanismo descentralizado de **controle de regras, permissões e provas**. Ela assegura que o paciente mantenha a soberania sobre seus dados, que os profissionais da saúde atuem dentro de limites claramente definidos e que todas as interações relevantes sejam registradas de forma imutável, auditável e resistente a fraudes.

Esse uso da *blockchain* está alinhado tanto com os princípios técnicos de sistemas distribuídos confiáveis quanto com os requisitos legais de proteção de dados sensíveis previstos na LGPD.

Requisitos funcionais.

Req_FN001 - Cadastrar paciente

Requisito:	Detalhamento para implementação:
O sistema deve prover a funcionalidade de **cadastro de um novo paciente** por meio de uma interface gráfica (WEB e/ou aplicativo móvel).	Durante o cadastro, o paciente deverá informar **nome completo, número de telefone celular, CPF e endereço de e-mail**. Esses dados serão utilizados exclusivamente para fins de identificação inicial e validação do usuário no sistema, podendo ser verificados por mecanismos auxiliares como SMS ou MFA. No momento do cadastro, deverá ser criado um **AccountId no ambiente Substrate/Polkadot**, que representará de forma única o paciente no sistema *blockchain*. Esse AccountId será utilizado pelo *runtime* Substrate para controle de permissões, registro de consentimentos e rastreabilidade das ações associadas ao paciente. O sistema deverá gerar um **par de chaves criptográficas assimétricas (chave pública e chave privada)** associado ao paciente, seguindo os algoritmos padrão suportados pelo Substrate/Polkadot. A **chave privada**: **não deverá ser armazenada em blockchain nem em banco de dados centralizado**; deverá permanecer **exclusivamente sob posse do paciente**, armazenada de forma segura no dispositivo utilizado no cadastro; será utilizada para **assinar digitalmente transações**, como a concessão ou revogação de consentimento de acesso aos dados médicos. A **chave pública**: poderá ser registrada no ambiente Substrate, associada ao AccountId do paciente; será utilizada para **verificação de assinaturas digitais**, identificação criptográfica do paciente e encapsulamento (wrap) de chaves simétricas destinadas ao próprio paciente; **não será derivada de dados pessoais**, como CPF ou telefone, nem constituirá *hash* desses dados, garantindo assim a não exposição de informações sensíveis. A autenticação do paciente no sistema será realizada por meio da **assinatura digital de transações**, utilizando sua chave privada, permitindo que o Substrate valide a autoria das ações sem nunca ter acesso à chave privada em si. O AccountId do paciente deverá ser utilizado como referência única no ambiente Polkadot, não substituindo nem expondo diretamente sua identidade civil, que permanecerá tratada apenas em camadas off-chain do sistema. A geração das chaves criptográficas e do AccountId deverá ocorrer com o

	suporte do **ecossistema Substrate/Polkadot**, respeitando as boas práticas de segurança e soberania do usuário sobre suas credenciais criptográficas.

Req_FN002 - Confirmar com número de celular ou e-mail

Requisito:	Detalhamento para implementação:
O sistema deve prover a funcionalidade de **validação dos dados informados no cadastro** de usuários, sejam eles pacientes ou profissionais da saúde, por meio de mecanismos de **verificação em dois fatores (2FA)**.	Durante o processo de cadastro, o usuário deverá confirmar **ao menos um meio de contato válido**, sendo **número de telefone celular ou endereço de e-mail**. A validação será realizada mediante o envio de um **código temporário de verificação**, que deverá ser informado pelo usuário na interface gráfica (WEB e/ou aplicativo móvel) para que o cadastro seja concluído. Caso a validação seja realizada por **SMS**, o sistema deverá enviar um código de verificação para o número de telefone informado. Caso a validação seja realizada por **e-mail**, o fluxo de verificação deverá ser equivalente ao do SMS, exigindo a inserção do código recebido. Caso o usuário não forneça um número de telefone nem um endereço de e-mail válidos, o processo de cadastro **não deverá prosseguir**. A validação por SMS ou e-mail tem como objetivo **confirmar a posse do meio de contato informado**, reduzindo riscos de cadastros fraudulentos e fortalecendo a confiabilidade do sistema, sem substituir os mecanismos criptográficos de autenticação baseados em assinaturas digitais. O envio de mensagens de SMS e e-mails poderá ser implementado utilizando **serviços de terceiros**, como plataformas de computação em nuvem (por exemplo, AWS ou Firebase), escolhidas pela facilidade de integração e confiabilidade operacional.

Req_FN003 - Salvar cadastro em banco de dados

Requisito:	Detalhamento para implementação:

O sistema deve prover a funcionalidade de **persistir os dados cadastrais dos usuários** (pacientes e profissionais da saúde) em um **banco de dados off-chain**, logicamente isolado da blockchain.	Os dados pessoais de cadastro, como nome e informações de contato, deverão ser armazenados em tabelas do banco de dados. A **chave primária (PK)** dessas tabelas deverá ser a **chave pública do usuário**, gerada durante o processo de cadastro por meio de código Rust executado no ambiente Substrate. O **AccountID** do usuário deverá ser mantido exclusivamente no contexto do ambiente Polkadot/Substrate, sendo utilizado para identificação da conta *blockchain*, assinatura de transações e controle de permissões *on-chain*, **não sendo utilizado como chave primária no banco de dados**. Após o armazenamento dos dados cadastrais no banco de dados, o sistema deverá gerar um **código *hash* criptográfico** representando esses dados. Esse *hash* deverá ser **registrado on-chain**, associado à chave pública do usuário, servindo como uma **prova de integridade e veracidade** das informações armazenadas *off-chain*. Em qualquer momento futuro, o *hash* armazenado na *blockchain* poderá ser recalculado a partir dos dados recuperados do banco de dados e comparado ao valor *on-chain*, permitindo verificar se os dados pessoais foram alterados indevidamente fora das regras do sistema. A separação entre banco de dados *off-chain* e *blockchain* tem como objetivo **reduzir custos de armazenamento *on-chain***, preservar a privacidade dos dados pessoais e, ao mesmo tempo, manter garantias criptográficas de integridade e imutabilidade por meio da *blockchain*.

Req_FN004- Criptografar dados em banco de dados

Requisito:	Detalhamento para implementação:
O sistema deve prover a funcionalidade de **armazenar os dados do histórico médico de forma criptografada no banco de dados**, garantindo a confidencialidade de informações sensíveis.	Antes de qualquer operação de persistência ou atualização do histórico médico, os dados deverão ser **criptografados off-chain**, no software do usuário autorizado (profissional da saúde ou paciente), utilizando um **algoritmo de criptografia simétrica comum a todo o sistema**. O paciente irá criptografar somente seus dados de cadastro e nada mais, antes de salvá-los no banco de dados. Para cada operação de salvamento ou atualização do histórico médico, o sistema deverá **gerar uma nova chave de criptografia simétrica**, distinta das anteriores. Essa chave tornará automaticamente inválida qualquer chave utilizada em gravações anteriores. A geração da chave de criptografia deverá ser controlada pelo código Rust executado no ambiente Substrate, que será responsável por: autorizar a operação de salvamento conforme regras de consentimento e tempo de validade; disponibilizar a chave de criptografia ao usuário autorizado, de forma **criptografada com sua chave pública**, para uso exclusivo *off-chain*. Como consequência, profissionais da saúde que não possuam autorização válida ou que tenham utilizado chaves anteriores **não conseguirão descriptografar versões atualizadas do histórico médico**, mesmo que

	tenham tido acesso no passado. Esse mecanismo de renovação contínua de chaves tem como objetivo: aumentar a segurança dos dados médicos; garantir o controle de acesso baseado em consentimento; impedir que médicos antigos tenham acesso a dados criados ou modificados por profissionais posteriores. Os detalhes completos desse fluxo estão descritos nos diagramas de sequência da modelagem UML do sistema.

Req_FN005 - Usar chave para cripografar/descriptografar dados

Requisito:	Detalhamento para implementação:
O sistema deve prover a funcionalidade de **geração e gerenciamento seguro de chaves criptográficas simétricas** utilizadas na criptografia e descriptografia dos dados de histórico médico armazenados no banco de dados off-chain.	Sempre que for necessária a criação ou alteração de informações no histórico médico de um paciente, o sistema deverá **gerar uma nova chave de criptografia simétrica**, exclusiva para aquela operação de salvamento. A geração dessa chave deverá ser realizada de forma segura pelo **código Rust executado no ambiente Substrate (Polkadot)**, após a validação das regras de consentimento, autenticação e tempo de validade. Cada nova chave de criptografia gerada deverá **invalidar automaticamente a chave utilizada anteriormente**, garantindo que apenas a versão mais recente do histórico médico possa ser descriptografada com a chave atual, e impedindo o acesso a dados atualizados por usuários que possuam apenas chaves antigas. A chave de criptografia **não deverá ser armazenada em texto claro na *blockchain***. Em vez disso, o sistema deverá: disponibilizar a chave de criptografia apenas aos usuários autorizados (médico atual e/ou paciente); transmitir essa chave de forma segura, **criptografada com a chave pública do destinatário**, para uso exclusivo *off-chain*; manter na *blockchain* apenas as informações necessárias para controle de acesso, autorização e validação, sem expor segredos criptográficos. Sempre que um usuário autorizado precisar **ler ou alterar** os dados do histórico médico, os dados armazenados no banco de dados deverão ser recuperados e **descriptografados *off-chain***, utilizando a chave de criptografia correspondente à última versão válida do histórico. Após a descriptografia, o sistema deverá utilizar o **código *hash* previamente registrado na *blockchain*** para validar a integridade e a veracidade dos dados recuperados. Esse *hash* deverá corresponder **exclusivamente aos dados mais recentes do histórico médico**, ou seja, à última alteração realizada, garantindo que o conteúdo não tenha sido adulterado fora das regras do

	sistema. Os fluxos completos de geração, distribuição, invalidação de chaves e validação de dados estão detalhados na modelagem UML por meio dos diagramas de sequência. Cada vez que os histórico for criptografado, toda a informção presente nele será criptografada novamente. Ou seja, todos os dados de toda a vida do paciente.

Req_FN006 - Gerar AccountID

Requisito:	Detalhamento para implementação:
O sistema deve prover a funcionalidade de **criação de um AccountID** para cada usuário que se cadastrar na aplicação (paciente ou profissional da saúde). Esse AccountID será criado automaticamente no momento do registro do usuário no ambiente Polkadot, conforme os mecanismos padrão da plataforma.	O AccountID será utilizado **exclusivamente no contexto do Substrate/Polkadot**, principalmente para: identificação do usuário em transações *on-chain*; assinatura e validação de operações no runtime; associação do usuário aos custos computacionais e taxas de transação (fees) no ambiente *blockchain,* se isso se aplicar. O AccountID **não precisa ser armazenado no banco de dados *off-chain***, uma vez que sua função está restrita à camada *blockchain*. A relação entre usuários e seus dados persistidos *off-chain* deverá ser realizada por meio da **chave pública criptográfica do usuário**, gerada durante o cadastro. A chave pública do usuário será utilizada como **chave primária (PK)** para os dados armazenados no banco de dados, reduzindo a necessidade de exposição e manipulação direta do AccountID fora da *blockchain*. Essa decisão tem como objetivo: minimizar riscos associados à exposição do AccountID; reduzir a superfície de ataque contra a identidade *on-chain* do usuário; manter uma clara separação entre identidade *blockchain* (*on-chain*) e persistência de dados sensíveis (*off-chain*). O AccountID poderá exercer múltiplas funções dentro do sistema, relacionadas às regras de consenso, autorização e rastreabilidade no Substrate, e por isso deverá ser tratado como um identificador sensível, com uso restrito à camada *blockchain*.

Req_FN007- Exibir histórico médico do paciente

Requisito:	Detalhamento para implementação:

O sistema deve prover a funcionalidade de **exibição do histórico médico ao usuário paciente**, por meio de interface gráfica (WEB e/ou APP), garantindo confidencialidade, integridade e controle exclusivo do titular dos dados.	Para que o paciente visualize seu histórico médico, deverá existir uma funcionalidade que permita a **recuperação dos dados médicos armazenados no banco de dados *off-chain***, desde que o paciente esteja devidamente autenticado no sistema. A autenticação do paciente será realizada por meio da **assinatura digital de uma transação**, utilizando sua chave privada local, sem que essa chave seja jamais transmitida ou armazenada fora do dispositivo do usuário. A liberação do acesso ao histórico médico será realizada por meio de uma **transação no ambiente Substrate**, assinada pelo paciente, a qual será validada pelo código Rust executado no runtime da *blockchain*. O runtime verificará a assinatura utilizando a chave pública registrada, concedendo o consentimento necessário para a leitura dos dados. Após a autorização on-chain, o software do paciente poderá: recuperar os dados criptografados do banco de dados off-chain; obter a chave simétrica necessária para a descriptografia, de forma segura; descriptografar os dados localmente (*off-chain*); validar a integridade dos dados utilizando o **código *hash* previamente registrado na** *blockchain*, garantindo que o conteúdo não foi adulterado. Os dados exibidos ao paciente deverão estar **estritamente disponíveis em modo somente leitura**. O paciente **não poderá editar, alterar ou excluir** quaisquer informações do seu histórico médico. A modificação ou criação de dados no histórico médico será permitida exclusivamente a profissionais da saúde devidamente cadastrados e autenticados no sistema, mediante consentimento explícito do paciente, validado por regras de autorização implementadas no código Rust executado no ambiente *blockchain*.

Req_FN008 - Gerar consentimento de edição de dados

Requisito:	Detalhamento para implementação:

Para que dados do histórico médico de um paciente possam ser criados, editados ou alterados por um profissional da saúde, o paciente deverá **gerar explicitamente um consentimento de edição** por meio do sistema.	A geração do consentimento implicará na execução de uma **transação assinada** pelo paciente no ambiente Substrate, validada pelo código Rust executado no runtime da *blockchain*. Essa transação servirá como prova inequívoca da autorização concedida pelo titular dos dados. O fluxo de geração do consentimento deverá ocorrer da seguinte forma: O profissional da saúde deverá disponibilizar sua **chave pública** ao paciente. Opcionalmente, o paciente poderá solicitar e consultar a **chave pública do profissional da saúde**, a fim de verificar seu perfil e identidade no sistema. O paciente, por meio da interface do software, decide liberar o acesso ao seu histórico médico para um profissional específico. O sistema registra *on-chain* uma **associação entre a chave pública (ou AccountID) do profissional da saúde e a chave pública do paciente**, contendo as permissões concedidas e o tempo de validade do consentimento. Essa associação será mantida como **estado *on-chain***, sendo pública, auditável e imutável enquanto válida, conforme as regras da *blockchain*. Em nenhum momento o sistema armazenará ou manipulará chaves privadas no runtime. O consentimento deverá possuir um **prazo de validade definido pelo paciente**, por exemplo, o equivalente à duração de uma consulta médica. Após o término desse prazo, o consentimento será automaticamente considerado expirado, impossibilitando o profissional da saúde de solicitar a gravação de quaisquer alterações no histórico médico do paciente. O controle temporal do consentimento deverá ser implementado com base na **contagem de blocos da** *blockchain*, garantindo determinismo, previsibilidade e independência de relógios externos. Qualquer tentativa de salvamento de dados após o vencimento do consentimento deverá ser rejeitada automaticamente pelo código Rust executado no runtime. Os detalhes completos desse fluxo estão descritos nos diagramas de sequência apresentados na modelagem UML do projeto.

Req_FN009 -Gerar consentimento de apenas visualização de dados

Requisito:	Detalhamento para implementação:
O sistema deve permitir que o paciente **gere consentimento para visualização do histórico médico** por parte de um profissional da saúde, com **permissão exclusiva de leitura**, sem possibilidade de criação, edição ou alteração de dados.	A geração desse consentimento deverá ocorrer por meio de uma **transação assinada pelo paciente** no ambiente Substrate, validada pelo código Rust executado no runtime da *blockchain*, de forma análoga ao processo de geração de consentimento para edição de dados. O consentimento de visualização deverá registrar *on-chain*:

	a chave pública ou AccountID do paciente; a chave pública ou AccountID do profissional da saúde autorizado; o **tipo de permissão concedida**, explicitamente definida como *somente leitura*; opcionalmente, um período de validade para a visualização, conforme configuração do paciente. Diferentemente do consentimento para edição, o consentimento de visualização **não concederá permissão de escrita em nenhuma circunstância**, independentemente do tempo de validade configurado. Qualquer tentativa do profissional da saúde de salvar, alterar ou criar dados no histórico médico do paciente deverá ser automaticamente rejeitada pelo código Rust executado no runtime da *blockchain*. O controle dessas permissões será mantido como **estado *on-chain***, garantindo rastreabilidade, auditabilidade e impossibilidade de elevação indevida de privilégios. Os detalhes completos do fluxo de geração e uso desse consentimento estão descritos nos diagramas de sequência apresentados na modelagem UML do projeto.

Req_FN010- Controlar tempo de consentimentos de edição de dados

Requisito:	Detalhamento para implementação:
O sistema deve permitir que o usuário (paciente) **controle o tempo de validade de um consentimento** concedido a um profissional da saúde.	O prazo de validade do consentimento deverá ser **configurável exclusivamente pelo paciente**, apenas no momento da criação do consentimento. Após criado, o tempo de validade não poderá ser alterado, garantindo previsibilidade e segurança no controle de acesso. A validade do consentimento será controlada por meio de um **contador de tempo executado no Substrate**, com o estado do consentimento sendo mantido on-chain na *blockchain*. Esse contador será utilizado para verificar, a cada tentativa de ação, se o prazo ainda está vigente. Qualquer transação de salvamento, criação, edição ou exclusão de dados do histórico médico, executada via Substrate e controlada pelo código Rust no runtime da *blockchain*, **não terá efeito** caso o prazo de validade do consentimento tenha expirado (*timeout*). Nessas condições, a transação deverá ser automaticamente rejeitada. Esse mecanismo permitirá ao paciente manter controle total sobre **quando e por quanto tempo** seus dados de histórico médico poderão ser modificados por um profissional da saúde. Um exemplo típico de uso desse recurso é a definição do prazo de validade do consentimento como o tempo estimado de uma consulta médica. Os detalhes completos do fluxo de criação, validação temporal e invalidação automática do consentimento estão descritos nos diagramas de sequência da modelagem UML do sistema.

Req_FN011- Revogar/pausar consentimentos existentes

Requisito:	Detalhamento para implementação:
O sistema deve permitir que o usuário (paciente) **revogue ou pause qualquer consentimento** previamente concedido para acesso ao seu histórico médico.	O paciente poderá revogar **apenas os consentimentos que ele próprio concedeu**, sendo tecnicamente impossível alterar ou interferir em consentimentos pertencentes a outros usuários. Essa restrição deverá ser garantida pelo controle de acesso implementado no Substrate e validado por transações na *blockchain*. Ao revogar um consentimento, o sistema deverá **definir imediatamente o tempo de validade desse consentimento como zero**, caracterizando um estado de *timeout* instantâneo. A partir desse momento, qualquer tentativa de um profissional da saúde de salvar, criar, editar ou excluir dados do histórico médico do paciente deverá ser automaticamente bloqueada pelo código Rust executando no ambiente Substrate, sem que nenhuma alteração seja persistida no banco de dados. Alternativamente, o paciente poderá **pausar um consentimento ativo**. Durante o período em que o consentimento estiver pausado: A contagem regressiva do tempo de validade ficará suspensa; O profissional da saúde não conseguirá realizar operações de salvamento ou modificação no histórico médico do paciente. Ao despausar o consentimento, a contagem regressiva do tempo de validade será retomada **a partir do ponto exato em que havia sido interrompida**, preservando o tempo restante originalmente configurado no momento da criação do consentimento. Os estados de revogação, pausa e retomada do consentimento deverão ser registrados e controlados por meio de transações no Substrate, mantendo o estado atualizado on-chain na *blockchain*. Os fluxos completos desse comportamento estão detalhados nos diagramas de sequência da modelagem UML do sistema.

Req_FN012 - Impedir alteração de dados inseridos por outros profissionais

Requisito:	Detalhamento para implementação:
O sistema deve impedir que um profissional da saúde **altere dados médicos que tenham sido registrados por outro profissional**.	Quando um médico solicitar salvar alterações em um histórico médico, o sistema verificará se a chave pública de médico anteriormente associada a esses dados é igua àquela do médico corrente. Se for igual => o mesmo médico está alterando informações que ele mesmo criou no passado e portanto as alterações poderão ser salvas (isso é uma edição). Caso contrário nada será alterado (edição bloqueada automaticamente). Esse controle será feito na camada do *substrate*. Veja diagrama de sequência para mais detalhes. Nesse caso, os dados escritos pelo médico criarão um novo registro no histórico médico. Ou seja, os dados criados por outro profissional da saúde anterior permanecerão inalterados. Os dados novos serão uma criação, não uma

	edição. Justificativa: mater um histórico real das informações médicas de um paciente , ao longo da vida dele, ao passar por diferentes médicos.

Req_FN013 - Mostrar dados de histórico médico para profissional da saúde

Requisito:	Detalhamento para implementação:
O sistema deverá prover a funcionalidade de **exibir os dados médicos (histórico) de um paciente a um profissional da saúde**, desde que exista um **consentimento válido previamente concedido pelo paciente**.	Um médico poderá solicitar ao sistema o acesso ao histórico médico de um paciente. Esse acesso somente será permitido se: o paciente tiver concedido explicitamente o consentimento para visualização e/ou edição dos dados; e o consentimento ainda estiver dentro do seu prazo de validade (*timeout*), conforme definido no momento da concessão. **Fluxo de execução** O médico solicita o acesso ao histórico médico informando: a **chave pública do paciente**; sua **identidade autenticada**, comprovada por assinatura com sua chave privada (de forma transparente ao usuário). O módulo executando no Substrate verifica, na *blockchain*, se existe uma **associação válida entre o médico e o paciente**, criada previamente por meio de um consentimento ativo e ainda não expirado. Caso a associação seja válida, o sistema autoriza o acesso ao histórico médico. Caso contrário, o acesso é negado. Uma vez autorizado, o sistema (módulo em Rust no Substrate) retorna ao software do médico: a **chave simétrica cifrada** correspondente à última versão válida do histórico médico (chave de criptografia); o **hash de verificação** associado aos dados armazenados no banco de dados. O software do médico: recupera os dados do histórico médico no banco de dados; descriptografa os dados utilizando a chave simétrica recebida (já decifrada com a chave pública dele mesmo); valida a integridade e a autenticidade dos dados por meio do hash fornecido.

	Após a validação bem-sucedida: o médico poderá **visualizar os dados médicos** do paciente; se o consentimento permitir edição, o médico poderá **criar novos registros ou editar registros que ele próprio tenha criado anteriormente**, respeitando as regras de autoria e tempo de validade do consentimento;

Req_FN014 -Criar / alterar dados de histórico médico

Requisito:	Detalhamento para implementação:
O sistema deverá prover as condições necessárias para que um **profissional da saúde** possa **criar ou alterar o histórico médico de um paciente**, desde que exista um **consentimento válido** previamente concedido pelo paciente.	Após criar ou alterar informações médicas no histórico, o médico acionará a funcionalidade de **salvar os dados** por meio da interface gráfica do seu software. **Fluxo de execução** Ao solicitar o salvamento, o sistema (módulo executando no Substrate) verifica se existe um **consentimento ativo e não expirado** que autorize o médico corrente a realizar alterações no histórico médico do paciente. Em seguida, o sistema verifica se já existem dados no histórico médico do paciente: Caso **não existam dados prévios**, os dados informados pelo médico serão tratados como uma **criação de novo registro** no histórico. Caso existam dados prévios, o sistema analisa se a **chave pública do médico corrente** corresponde à chave associada ao último registro criado por um profissional da saúde. O resultado dessa verificação determina o tipo de operação: Se a chave públcia do médico corrente **for a mesma** associada ao registro anterior, os dados serão **editados**, substituindo o conteúdo do último registro criado por esse mesmo médico=> Médico altera seu próprio texto já existente. Se a chave pública **for diferente**, isso indica que o paciente está sendo atendido por um **novo profissional**, e os dados informados serão salvos como um **novo registro no histórico médico**, sem alterar registros anteriores. Nesse caso, trata-se de uma **criação**, e não de uma edição. Antes do salvamento efetivo dos dados: o sistema gera uma **nova chave simétrica de criptografia**, específica para esse novo estado do histórico médico; Essa chave é enviada do software em Rust para o software do

	médico, cifrada com a chave pública do médico. é gerado também um **hash criptográfico** referente exclusivamente aos dados que estão sendo gravados nesse momento. O software do médico: utiliza a chave simétrica gerada para **criptografar os dados do histórico**; Mas essa chave é decifrada com a chave privada do médico. envia os dados criptografados para armazenamento no banco de dados externo à *blockchain*. A **chave simétrica cifrada** e o **hash de verificação**: também ficam associados ao paciente e à versão mais recente do histórico médico; Nesse caso a chave simétrica é cifrada com a chave pública do paciente. são registrados e mantidos exclusivamente no Substrate (na *blockchain*), **nunca no banco de dados**, garantindo integridade, rastreabilidade e controle de acesso. **Justificativa** Esse mecanismo garante que: médicos não alterem informações criadas por outros profissionais; o histórico médico seja **imutável por autoria**, mantendo registros consistentes ao longo do tempo; cada alteração ou criação gere um novo estado criptograficamente protegido, com renovação de chaves e validação de integridade.

Req_FN015 - Salva dados de históricos em banco de dados

Requisito:	Detalhamento para implementação:
O sistema deve prover as condições necessárias para que um **profissional da saúde** possa **salvar o histórico médico de um paciente**, desde que exista um **consentimento válido** concedido previamente pelo paciente.	Assim que o médico decidir salvar as informações registradas no histórico do paciente, o sistema executará o seguinte fluxo: O sistema gera uma **nova chave simétrica de criptografia**, específica para essa operação de salvamento. Cada salvamento gera obrigatoriamente uma nova chave. A chave anteriormente utilizada torna-se automaticamente

	inválida para novos acessos. O **software do médico** utiliza essa nova chave para **criptografar todo o histórico médico do paciente**, ou seja, o conjunto completo de dados médicos acumulados até aquele momento (ao longo da vida), e não apenas as informações criadas ou alteradas na sessão atual. Os dados criptografados são então **gravados no banco de dados**, que permanece isolado da *blockchain*, contendo apenas: o *payload* criptografado do histórico médico; um identificador do profissional da saúde que realizou a gravação (por exemplo, a **chave pública do médico** ou um identificador derivado dela). **Observação**: a chave privada do médico nunca é armazenada, nem no banco de dados nem na blockchain. Ela é utilizada exclusivamente no lado do usuário para autenticação e assinatura de transações. O sistema gera um **hash criptográfico** referente ao estado mais recente do histórico médico, permitindo a validação futura da integridade dos dados armazenados. A **chave simétrica de criptografia**, devidamente **cifrada para os destinatários autorizados** (no mínimo o paciente e o médico corrente), bem como o **hash de validação**, são registrados no Substrate, ficando disponíveis na *blockchain* apenas como **metadados criptográficos**, nunca em formato legível. **Justificativa** Esse fluxo garante que: cada versão do histórico médico possua uma chave de criptografia própria; médicos antigos não consigam acessar dados atualizados por novos profissionais; a autoria dos registros médicos seja rastreável sem violar a segurança das chaves privadas; o banco de dados nunca armazene informações sensíveis em claro; a *blockchain* seja utilizada apenas para controle de acesso, integridade e auditoria, reduzindo custos e exposição de dados.

Req_FN016 - Relacionar dados em banco de dados com informação hash

Requisito:	Detalhamento para implementação:
Os dados médicos **criados ou alterados por um profissional da saúde** deverão ser validados por meio de um **código hash criptográfico**, com o objetivo de garantir a **integridade** e a **imutabilidade lógica** dessas informações.	Após o médico concluir a edição ou criação de informações no histórico médico do paciente, o sistema deverá: Gerar um **código hash** a partir **exclusivamente dos dados médicos escritos ou alterados na última operação** realizada por esse médico. Associar esse código hash ao respectivo registro do histórico médico, juntamente com os metadados de autoria e controle de acesso, mantendo esse hash registrado no Substrate (*blockchain*). Quando os dados médicos forem recuperados posteriormente do banco de dados: O software do médico ou do paciente deverá descriptografar os dados utilizando a chave simétrica válida correspondente à última gravação. O sistema recalculará o hash sobre os dados recuperados referentes à última edição registrada. O hash recalculado será comparado com o hash armazenado na *blockchain*. Caso os valores coincidam, ficará comprovado que os dados **não foram alterados fora do fluxo controlado do sistema**.

Req_FN017 -Manter informação hash em blockchain

Requisito:	Detalhamento para implementação:
O **código *hash*** das informações médicas deverá ser mantido na *blockchain*, de forma que seja possível, a qualquer momento, obter o ***hash* verdadeiro e confiável** para fins de validação dos dados armazenados no banco de dados *off-chain*.	A *blockchain* será utilizada como fonte de referência imutável para esse código *hash*, garantindo que: o *hash* não possa ser alterado após sua criação; o valor armazenado seja confiável para auditoria e verificação de integridade; qualquer tentativa de adulteração dos dados no banco de dados possa ser detectada. Cada código *hash* deverá estar **associado a um único paciente**, e essa associação também deverá ser mantida na *blockchain*. O *hash* registrado corresponderá **exclusivamente aos dados médicos gerados ou alterados na última operação válida**, conforme definido nos fluxos de consentimento e salvamento do histórico médico. A *blockchain* **não armazenará dados médicos em si**, mas apenas: o código *hash*;

	os identificadores necessários para associá-lo ao paciente; e os metadados de controle definidos no runtime (como permissões e validade de consentimento).

Req_FN018 -Criar solicitação de acesso a dados médicos de históricos

Requisito:	Detalhamento para implementação:
O sistema deve prover uma forma de um profissional da saúde solicitar o consentimento para acessar dados de histórico médico.	Esse requisito está totalmente coberto pelas funcionalidade definidas em outros requisitos acima.

Req_FN019 -Cadastar profissional da saúde.

Requisito:	Detalhamento para implementação:
O sistema deve prover a funcionalidade de **cadastrar um novo profissional da saúde** por meio de uma interface gráfica (WEB e/ou APP).	Durante o cadastro, o profissional deverá informar, no mínimo: nome completo; número de telefone celular atual; e-mail; CPF; CRM ou outro identificador profissional válido emitido por órgão regulador da sua área. Após a validação desses dados, o sistema deverá **criar uma identidade criptográfica** para o profissional da saúde, composta por: um **par de chaves criptográficas** (chave pública e chave privada), geradas de forma segura; um **AccountID** no ambiente Polkadot, associado a essa identidade. **Geração e Armazenamento das Chaves** A geração das chaves criptográficas deverá ocorrer por meio de **código Rust executando no Substrate Polkadot**. A **chave privada não deverá ser armazenada na** *blockchain* nem em bancos de dados centralizados. A chave privada deverá permanecer **exclusivamente no dispositivo do profissional**, sob responsabilidade do próprio usuário. A **chave pública** poderá ser registrada na *blockchain* e associada ao

AccountID do profissional.

As chaves **não são derivadas uma da outra**; ambas fazem parte de um par criptográfico gerado conjuntamente por algoritmo padrão (ex.: Ed25519).

Utilidade da Chave Pública

A chave pública do profissional da saúde terá as seguintes finalidades:

identificar unicamente o médico dentro do sistema;

permitir que pacientes consultem o perfil do profissional;

ser utilizada como identificador lógico (por exemplo, chave de referência em banco de dados);

participar dos processos de concessão de consentimento, juntamente com a chave pública do paciente;

permitir que o runtime no Substrate valide assinaturas feitas pelo profissional.

Utilidade da Chave Privada

A chave privada do profissional da saúde será utilizada para:

autenticar o profissional no sistema por meio de assinaturas digitais;

assinar transações enviadas ao Substrate, como solicitações de leitura ou gravação de histórico médico;

comprovar, de forma criptográfica, a autoria das alterações realizadas em dados médicos.

O uso da chave privada será sempre **transparente ao usuário**, ocorrendo internamente no software do profissional.

Utilidade do AccountID

O AccountID será utilizado para:

representar o profissional no ambiente Polkadot;

identificar o responsável por transações executadas no Substrate;

permitir o controle de permissões, custos e rastreabilidade de operações on-chain (se isso se tornar necessário)

O AccountID **não precisa ser usado como chave primária no banco de dados**, sendo reservado principalmente para interações com o ambiente Polkadot.

Considerações de Segurança

Nenhuma chave privada deverá ser armazenada em *blockchain*.

	A *blockchain* será utilizada apenas para armazenar **chaves públicas, *hashes* e metadados de controle**, nunca segredos criptográficos. A perda da chave privada implica na impossibilidade de o profissional autenticar-se ou assinar transações, conforme os princípios de segurança do sistema. Mas ele poderá se autenticar novamente, para continuar usando o sistema.

Req_FN020 -Registrar qual profissional editou qual dado, para rastreamento.

Requisito:	Detalhamento para implementação:
O sistema deve prover um mecanismo de **carimbo de autoria** para cada registro criado no histórico médico de um paciente.	Ao gravar novas informações médicas (criação ou edição válida de dados), o sistema deverá: associar o registro do histórico médico à **chave pública do profissional da saúde responsável** pela criação daquela informação; registrar essa associação como metadado do histórico médico, permitindo identificar futuramente **quais profissionais contribuíram para o histórico ao longo da vida do paciente**. A **chave privada do médico nunca deverá ser armazenada**, nem no banco de dados nem na *blockchain*. A comprovação de autoria será feita por meio de: O identificador do profissional associado a cada registro poderá ser: a chave pública do médico; **Justificativa** Esse mecanismo garante que: a autoria de cada informação médica seja rastreável; nenhum segredo criptográfico seja armazenado; o histórico médico preserve valor jurídico, auditável e cronológico; a *blockchain* seja utilizada para garantir veracidade e não repúdio, sem violar princípios básicos de segurança.

Requisitos não funcionais.

Req_NF021 - usar banco de dados firebase

Requisito:	Detalhamento para implementação:

Req_NF022- uso de bolt.new

Requisito:	Detalhamento para implementação:

Requisitos futuros.

Req_FT023- função para geração de tickets numerados de ordenação em filas.

Requisito:	Detalhamento para implementação:

Req_FT024- implementação de front-end para celuar.

Requisito:	Detalhamento para implementação:

Req_FT025 - instalação de banco de dados em nuvem (AWS S3, etc).

Requisito:	Detalhamento para implementação:

Req_FT026- Definição das tabelas definitivas do banco de dados, para que as informações estejam organizadas por especialidades médicas.

Requisito:	Detalhamento para implementação:

Organizar as informações no histórico, por especialidade.	Isso permitirá que um médico especialista veja rapidamente o histórico de outros especialistas da mesma área.

Req_FT027- Usar QRCode para passar chaves públicas entre pessoas.

Requisito:	Detalhamento para implementação:

Req_FT028 - Mostrar dados de histórico médico filtrado por ano.

Requisito:	Detalhamento para implementação:

Req_FT029- orgaizar histórico em ordem decrescente cronológica (pilha).

Requisito:	Detalhamento para implementação:

Req_FT030 - Capturar fotografia da pessoa que fazem cadastro.

Requisito:	Detalhamento para implementação:

Req_FT031 - atualizar número de celuar e/ou e-mail

Requisito:	Detalhamento para implementação:
Permitir usuário atualizar seu número de celular ou e-mail registrados.	

Req_FT032 - Permitir manipular dinheiro na Polkadot

Requisito:	Detalhamento para implementação:

para a execução de transações.	Poderíamos conversar com o Gustavo e com o Lohann sobre esse ponto da utilização dos DOTs da Polkadot, mas vejo que poderia ser utilizado como um medidor de confiabilidade do médico/instituição médica (ideia do ChatCGT)

Req_FT033 - Rastrear ou listar médicos que alteraram um histórico médico.

Requisito:	Detalhamento para implementação:
Listar para um usuário do sistema a lista de médicos anteriores do histórico.	

Req_FT034 - Configuração do tempo limite para alterar um histórico.

Requisito:	Detalhamento para implementação:
Permitir o paciente configurar quanto tempo vai durar a concessão de alteração do seu histórico.	Inicialmente esse tempo pode ser um valor fixo. Ex: 3 horas. Ou N blocos minerados.

Req_FT035 - Gerenciamento de chaves / recuperação (UX e segurança)

Requisito:	Detalhamento para implementação:
	Usuários seguram chaves privadas; se perdê-las, perdem acesso. Pensar em recuperação (social recovery, custodial opcional com nível de consentimento, ou mecanismos de recuperação por verifiable credentials) sem quebrar o modelo de soberania. (Ideia do ChatGPT)

Req_FT036 - Armazenamento de chaves no dispositivo

Requisito:	Detalhamento para implementação:
Armazenar com segurança.	OK desde que seja em keystore cifrado (ex.: Android Keystore, iOS Secure Enclave, ou arquivo cifrado com senha). Para médicos (uso profissional) considerar suporte a hardware wallets ou HSM. (Ideia do ChatGPT)

Req_FT037 - Direito de esquecimento (ideia do ChatGPT)

Requisito:	Detalhamento para implementação:

Req_FT038 - **Auto $ustentabilidade do projeto**

Requisito:	Detalhamento para implementação: