CENTRO PAULA SOUZA

FACULDADE DE TECNOLOGIA DE FRANCA

“Dr. THOMAZ NOVELINO”

**TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

João Paulo Fernandes rodrigues

**A tecnologia digital de informação e comunicação para melhoria da qualidade de vida de pessoas diabéticas.**

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca - “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Claudio Eduardo Paiva

FRANCA/SP

2022

**A tecnologia digital de informação e comunicação para melhoria da qualidade de vida de pessoas diabéticas.**

**João Paulo Fernandes Rodrigues**

**Resumo**

O sistema multiplataformas tem como objetivo auxiliar no tratamento da doença diabetes mellitus, assumindo como principal função verificar o índice de glicemia e efetuar suas devidas correções através de sugestões quando necessário, tendo como base o cadastro das informações contidas em receitas médicas. O sistema permite a inserção de alimentos juntamente com seus devidos níveis de carboidratos e unidades de medida, com a finalidade de ser utilizada para realizar cálculos de correções alimentares mais complexas. O sistema também atende as necessidades de monitoramento da doença, gerando relatórios periódicos apresentando a média geral de glicemia, quantidade total de medicação consumida e total de exames realizados no período selecionado, no mesmo relatório estes mesmos índices citados anteriormente devem ser apresentados em quatro períodos diários de seis horas cada, com a finalidade de auxiliar ao profissional da saúde nas tomadas de decisões. O aplicativo possibilita a função de agendamento, onde o usuário pode definir dia e horário de suas consultas e exames que devem ser feitos em laboratórios, nessa mesma funcionalidade o sistema possibilita o registro de atividades físicas realizadas pelo usuário, com a finalidade de ajudar e incentivar o usuário no monitoramento de sua doença.

ATÉ 250 PALAVRAS.

**Palavras-chave:** Android, Controle, Diabetes, Insulina, Monitoramento.

***Abstract***

The multiplatform system aims to assist in the treatment of diabetes mellitus, assuming as its main function to check the glycemic index and make the necessary corrections through suggestions when necessary, based on the registration of information contained in medical prescriptions. The system allows the insertion of foods together with their appropriate levels of carbohydrates and units of measurement, in order to be used to perform calculations of more complex food corrections. The system also meets the needs for monitoring the disease, generating periodic reports showing the general average of blood glucose, total amount of medication consumed and total number of tests performed in the selected period, in the same report these same indices mentioned above must be presented in four daily periods of six hours each, with the purpose of assisting the health professional in decision making. The application allows the scheduling function, where the user can define the day and time of their consultations and exams that must be done in laboratories, in this same functionality the system allows the registration of physical activities performed by the user, in order to help and encourage the user in monitoring his illness.

***Keywords:*** *Android, Control, Diabetes, Insulin, Monitoring*

1 Introdução

A função da insulina é promover a entrada de glicose para as células do organismo de tal maneira que ela possa ser aproveitada para as diversas atividades celulares. A falta de insulina ou um defeito na sua ação promove acúmulos de glicose no sangue, esse acumulo é chamado de hiperglicemia. É de extrema importância tratar da hiperglicemia, tendo em vista que está associada a lesões da microcirculação, o que ocasiona um mal funcionamento em diversos órgãos como rins, olhos, os nervos e o coração.

Atualmente, aproximadamente 415 milhões de adultos apresentam Diabetes Mellitus (DM) em todo o mundo e 318 milhões de adultos possuem intolerância à glicose, com risco elevado de desenvolver a doença no futuro.

O Brasil é o quarto país com maiores taxas de DM na população adulta, com um total de 14,3 (12,9-15,8) milhões de pessoas de 20 a 79 anos com DM, com um gasto anual estimado de pelo menos US$ 21,8 bilhões (BRASil, 2017, p.09).

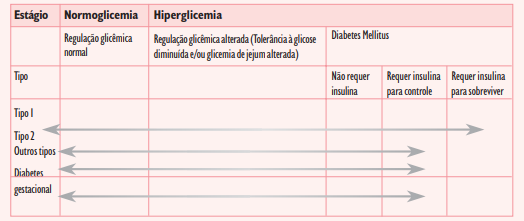
Os tipos de diabetes mais frequentes são o diabetes tipo 1 que corresponde a 10% do total dos casos, e o diabetes tipo 2 que corresponde a 90% dos casos.

O diabetes tipo 1, também conhecida como diabetes juvenil indica a deficiência absoluta do pâncreas na produção de insulina, nesse tipo especifico de diabetes o uso injetável de insulina passa a ser obrigatório e de extrema importância para que se possa prevenir cetoacidose, coma e morte.

O diabetes tipo 2 consiste no mal funcionamento do pâncreas que gera uma deficiência relativa a insulina. O diabetes tipo 2 em alguns casos é necessário o uso de insulina, porém ao contrário da diabetes tipo 1 o uso de insulina não visa evitar a cetoacidose, mais sim obter o controle de hiperglicemia, como pode-se ver na Figura 1. A cetoacidose nesse caso é rara, e quando presente geralmente está acompanhada de alguma infecção, trauma ou um estresse muito grave.

Outro tipo de diabetes encontrado com maior frequência e cuja etiologia ainda não está esclarecida é o diabetes gestacional, que, em geral, é um estágio pré-clínico de diabetes, detectado no rastreamento pré-natal. Outros tipos específicos de diabetes menos frequentes podem resultar de defeitos genéticos da função das células beta, defeitos genéticos da ação da insulina, doenças do pâncreas exócrino, endocrinopatias, efeito colateral de medicamentos, infecções e outras síndromes genéticas associadas ao diabetes (BRASIL, 2006, p.12).

**Figura 1 –** Requisitos para uso de insulina



**Fonte:** Brasília (2016)

É possível identificar pessoas portadoras de diabetes mellitus tipo 1 pelos seguintes sintomas: vontade de urinar com muita frequência, sede excessiva, fome em excesso, perda involuntária de peso, cansaço frequente, sonolência, coceira em todo corpo, infecções frequentes, irritabilidade e mudanças de humor repentinas.

Já as pessoas com diabetes tipo 2 geralmente apresentam os seguintes sintomas: aumento de sede, boca constantemente seca, vontade de urinar com muita frequência, cansaço frequente, visão turva ou embaçadas, feridas com a cicatrização muito lentas e infecções frequentes.

Com relação ao tratamento do diabetes devemos elencar como sendo essencial a alimentação e a pratica de exercícios físicos que ajudam na distribuição de glicose entre as células. A alimentação a quantidade energética ingerida deve ser adequada à atividade física e ser fracionada em 5 a 6 refeições/lanches diários. Para tanto, os pacientes devem ser encorajados a comer alimentos ricos em fibras, como frutas, verduras, legumes, feijões e cereais integrais.

Com relação aos diabéticos tipo 1, deve obrigatoriamente ser feita a aplicação de insulina para determinada quantidade de carboidratos, que será definido pelo médico.

As correções no nível de glicemia devem ser feitas com base a exames de sangue, onde são relatados o exato valor no momento do exame, a base para o cálculo de correção é definida obrigatoriamente pelo médico.

O diabetes apresenta altos índices de internações e morbi -mortalidade, o que ocasiona uma perda importante na qualidade de vida do portador de tal doença, isso ocorre devido a erros ao fazer as correções através de insulina e mal monitoramento da doença, devemos elencar também que pode ser um dos motivos algumas infecções encontradas nos pacientes. O diabetes é uma das principais causas de mortalidade juntamente com insuficiência renal, amputações de membros inferiores, cegueira e doenças cardiovasculares.

Estima-se que uma em cada 12 mortes em adultos no mundo possa ser atribuída ao DM, totalizando, aproximadamente, cinco milhões de casos ao ano, o que equivale a uma morte a cada seis segundos. A proporção de óbitos é ligeiramente maior em mulheres do que em homens. O gasto com DM, na maioria dos países, varia entre 5% e 20% das despesas globais em saúde (Brasília, 2017, p.09).

Como obter uma melhor qualidade de vida para diabéticos, pelo controle da quantidade de insulina a ser injetada no sangue, através de um aplicativo multiplataforma desenvolvido com Xamarin para facilitar a definição de horários e cálculos rápidos de alimentação a ser consumida, possibilitando o registro dos dados de medições para monitoramento do tratamento?

O aplicativo tem por objetivo evitar erros de cálculos referentes a insulina a ser aplicada, alertar o usuário de tomar os remédios nos devidos horários, fazer exames de correções, registrar observações referentes aos horários que forem feitos os exames e atividades físicas, lembrar sobre as próximas consultas e exames marcados, além de auxiliar o médico a obter um cronograma e monitoramento mais detalhado, dessa forma proporcionando uma melhor qualidade de vida para pessoas portadoras do diabetes mellitus.

**2 Levantamento de Requisitos**

2.1 Elicitação e especificação dos Requisitos

Este projeto foi desenvolvido a partir de técnicas de entrevistas com pessoas portadoras da doença “diabetes mellitus”, onde foram feitas reuniões presenciais destacando as principais dificuldades no controle e monitoramento da doença, também foi utilizado a técnica de etnografia, que durante um período pré-definido foi acompanhado o tratamento dessa doença com pessoas com maiores dificuldades de controle.

2.2 BPMN

**Figura 2:** BPMN

Diagrama

Descrição gerada automaticamente

2.3 Requisitos Funcionais

Usar modelo estudado nas aulas de engenharia de software.

**Quadro 1 –** Requisitos Funcionais do sistema

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RF001**-Cadastro de dados pessoais | Categoria:  ( ) Oculto  (X)Evidente | Prioridade:  (X) Altíssima  ( ) Alta  ( ) Média  ( ) Baixa | | |
| **Descrição**: O sistema deve gravar os dados pessoais de cada usuário sendo estes: Nome, e-mail, celular, tipo de diabetes que possuí. | | | | |
| **RF002**- Cadastro de medicamentos | Categoria:  ( ) Oculto  (X)Evidente | Prioridade:  ( ) Altíssima  ( ) Alta  ( ) Média  (X) Baixa | | |
| **Descrição:** O sistema deve registrar o nome dos medicamentos utilizados pelo usuário. | | | | |
| **RF003**- Cadastro de dosagens para correções. | Categoria:  ( ) Oculto  (X)Evidente | Prioridade:  ( ) Altíssima  (X) Alta  ( ) Média  ( ) Baixa | | |
| **Descrição**: O sistema deve registrar a quantidade de insulina referente a determinada quantidade de carboidratos, caso esteja especificado na receita médica.  Exemplo: 1 unidade para cada 15 gramas de carboidrato.  Obrigatoriamente o sistema deve registrar a quantidade de insulina referente ao total do valor de glicemia que deve ser controlado.  Exemplo: 1 unidade para cada 50 mg/dl. | | | | |
| **RF004**- Cadastro do exame de glicemia. | Categoria:  ( ) Oculto  (X)Evidente | | Prioridade:  (X) Altíssima  ( ) Alta  ( ) Média  ( ) Baixa | |
| **Descrição**: O sistema deve registrar dia, horário e o valor informado pelo usuário referente ao exame de glicemia, este valor deverá estar contido entre 21 mg/dl e 599 mg/dl, caso o valor seja inferior a 21 mg/dl o sistema apresentará as siglas LO, e caso o valor seja superior a 599 mg/dl o sistema apresentará as siglas HI. | | | | |
| **RF005-** Calcular média de glicêmia. | Categoria:  (X) Oculto  ( ) Evidente | | | Prioridade:  ( ) Altíssima  ( ) Alta  ( ) Média  (X) Baixa |
| **Descrição**: Caso seja solicitado o sistema deve fazer a média da glicemia em determinados períodos definidos pelo usuário.  Fórmula para média glicêmica:  Para cada exame registrado será somado o valor “1” em uma variável “X”.  Em uma segunda variável “Y” será somado o valor obtido através dos registros dos exames de glicemias do usuário.   Deste modo se o usuário registrar de seus exames em 3 horários diferentes, com os seguintes valore: 250; 150; 80 o sistema deverá somá-los e dividi-los pôr 3 chegando ao resultado de 160.   Média = Y/X. | | | | |
| **RF006**- Cálculo de correção de glicemia. | Categoria:  (X) Oculto  ( ) Evidente | | | Prioridade:  (X) Altíssima  ( ) Alta  ( ) Média  ( ) Baixa |
| **Descrição**: Com base nos dados obtidos através do cadastro do usuário, o sistema deve calcular a quantidade de medicamento a ser utilizado.  Desta forma se for registrado um valor hipotético de 200 mg/dl, o sistema deve fazer um cálculo objetivando chegar a um valor entre 70 à 150 mg/dl. | | | | |
| **RF007-** Cadastro de Alimento. | Categoria:  ( ) Oculto  (X) Evidente | | | Prioridade:  ( ) Altíssima  (X) Alta  ( ) Média  ( ) Baixa |
| **Descrição:** O usuário do sistema deverá inserir alimentos informando a quantidade de carboidratos e unidade de medida. | | | | |
| **RF008-** Cálculo para dosagem de alimentos. | Categoria:  (X) Oculto  ( ) Evidente | | | Prioridade:  ( ) Altíssima  (X) Alta  ( ) Média  ( ) Baixa |
| **Descrição:** O usuário deve informar quais alimentos foram consumidos e suas respectivas quantidades.  Hipoteticamente se o usuário registra um pão francês com equivalência a 15 gramas de carboidratos o sistema deverá definir 1 unidade de insulina para cada pão francês consumido. Este valor irá variar conforme o cadastro feito pelo usuário inicialmente ou conforme for atualizado os dados no sistema. | | | | |
| **RF009-** Sugerir a quantidade de medicamento a ser utilizado | Categoria:  (X) Oculto  ( ) Evidente | | | Prioridade:  (X) Altíssima  ( ) Alta  ( ) Média  ( ) Baixa |
| **Descrição:** Ao se obter os cálculos de correção de glicemia e correção alimentar, o sistema deverá somar os dois resultados e exibir para o usuário a quantidade recomendada de insulina, a sugestão não pode ser um número com casas decimais, deve ser um número inteiro.  Caso o resultado do exame de glicemia for um valor inferior a 70 mg/dl o sistema deve abortar a sugestão, exibindo uma mensagem relatando que o sistema só faz cálculos de sugestões para valores iguais ou superiores à 70 mg/dl. | | | | |
| **RF010-** Informar atividades físicas | Categoria:  ( ) Oculto  (X) Evidente | | | Prioridade:  ( ) Altíssima  ( ) Alta  ( ) Média  (X) Baixa |
| **Descrição:** O usuário pode informar sobre suas atividades físicas diária, onde o sistema deve registrar o dia, horário, nome da atividade realizada e observações feitas pelo usuário. | | | | |
| **RF011-** Monitorar a média de medicamentos consumidos. | Categoria:  (X) Oculto  ( ) Evidente | | | Prioridade:  ( ) Altíssima  ( ) Alta  (X) Média  ( ) Baixa |
| **Descrição:** O sistema deve assim que solicitado pelo usuário fazer o cálculo da média da quantidade de insulina aplicada por um determinado período.  O usuário informa o valor que foi aplicado, o sistema armazena o valor em uma variável “X” e soma o valor 1 em uma segunda variável “Y” para cada registro.  Média = X / Y. | | | | |
| **RF012-** Relatório | Categoria:  (X) Oculto  ( ) Evidente | | | Prioridade:  ( ) Altíssima  (X) Alta  ( ) Média  ( ) Baixa |
| **Descrição:** o usuário pode verificar todas as sugestões feitas pelo aplicativo através de um relatório.  O relatório deve conter a média geral de glicemia, quantidade total de exames registrados, total de dosagens aplicadas, média do total de dosagem aplicada, maior e menor índice de glicemia registrado no período selecionado. O sistema também deve gerar os mesmos índices citados anteriormente subdivididos em quatro períodos de 6 horas. | | | | |

2.4 Requisitos Não Funcionais

Usar modelo estudado nas aulas de engenharia de software.

**Quadro 2** – Requisitos Não Funcionais do sistema

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RNF001**-Login | O usuário deverá estar logado para utilizar o sistema. | Tipo Segurança | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |
| **RNF002**- Acessibilidade | Deve ser compatível para Android, IOS e Windows 10. | Tipo Compatibilidade | ( ) Desejável  (X) Obrigatório | ( ) Permanente  (X) Transitório |
| **RNF003-** Facilidade de uso. | O sistema deverá ser de fácil manuseio, tendo uma taxa de até 3 erros por dia no início de sua utilização. | Tipo Usabilidade | (X) Desejável  ( ) Obrigatório | ( ) Permanente  (X) Transitório |
| **RNF004-** Tempo estimado para acesso | O software deve conectar imediatamente o usuário, será estimado uma média de até 30 segundos de espera. | Tipo Desempenho | (X) Desejável  () Obrigatório | ( ) Permanente  (X) Transitório |
| **RNF005-** Tempo estimado para realizar cálculos | O software deverá apresentar os resultados dos cálculos em um tempo estimado de até 15 segundos. | Tipo Desempenho | (X) Desejável  ( ) Obrigatório | ( ) Permanente  (X) Transitório |
| **RNF006-** Disponibilidade de uso | O sistema deverá estar disponível durante as 24h do dia, não sendo necessário estar conectado a uma rede de internet. | Tipo Disponibilidade | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |
| **RNF007-** Identificação do usuário | O usuário deverá identificar qual tipo de diabetes que o usuário possui, se é tipo 1 ou tipo 2 | Tipo Usabilidade | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |
| **RNF008-** Entrega de relatório | O Relatório deve ser entregue assim que solicitado, sendo estimado um tempo de até 30 segundos de espera. | Tipo Usabilidade | ( ) Desejável  (X) Obrigatório | ( ) Permanente  (X) Transitório |
| **RNF09-** Segurança | O usuário apenas pode ver seus próprios dados, não sendo permitido ver dados de terceiros no aplicativo. | Tipo Segurança | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |
| **RNF010-** Armazenamento de dados | O sistema deve comunicar com o banco de dados SQLite. | Tipo Compatibilidade | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |
| **RNF011-** Linguagem de programação | O sistema deve ser desenvolvido utilizando a linguagem C#. | Tipo Compatibilidade | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |

**Quadro 3** – Matriz de rastreabilidade.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | RF1 | RF2 | RF3 | RF4 | RF5 | RF6 | RF7 | RF8 | RF9 | RF10 | RF11 | RF12 |
| RNF1 |  | X | X | X | X | X | X | X | X | X | X | X |
| RNF2 | X | X | X | X | X | X | X | X | X | X | X | X |
| RNF3 | X | X | X | X | X | X | X | X | X | X | X | X |
| RNF4 | X | X | X | X |  |  | X |  |  | X | X |  |
| RNF5 |  |  |  |  | X | X |  | X |  |  |  |  |
| RNF6 | X | X | X | X | X | X | X | X | X | X | X | X |
| RNF7 | X |  |  |  |  |  |  |  |  |  |  |  |
| RNF8 |  |  |  |  |  |  |  |  |  |  |  | X |
| RNF9 |  | X | X | X | X | X | X | X | X | X | X | X |
| RNF10 | X | X | X | X | X | X | X | X | X | X | X | X |
| RNF11 | X | X | X | X | X | X | X | X | X | X | X | X |

2.5 Casos de Uso

Índice de casos de uso e Diagrama de casos de uso

UC001 – Gerenciar dados.

UC002 – Informar o resultado do exame de glicemia.

UC003 – Gerenciar alimentos e dosagens.

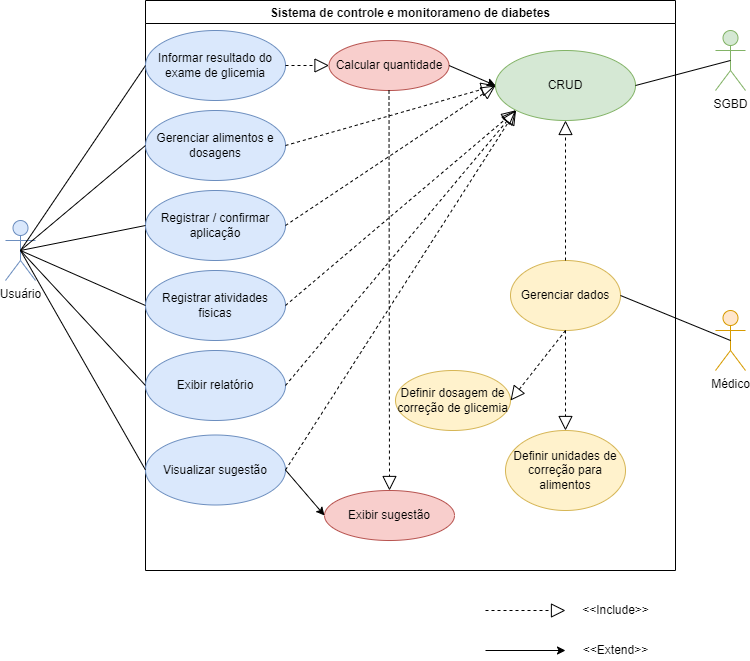
UC004 – Exibir sugestões.

UC005 – Registrar/Confirmar aplicação.

UC006 – Registrar atividades físicas.

UC007 – Exibir relatório.

**Figura 3:** Diagrama de Casos de Uso



Especificação de cada um dos casos de uso

**Quadro 5** – Use Case Cadastrar Usuários

|  |  |
| --- | --- |
| **Caso de Uso –** Gerenciar dados. | |
| **ID** | UC 001 |
| **Descrição** | Este caso de uso tem como objetivo gerenciar datas, horários e dosagens de cada medicação, referentes a agendamentos de consultas, exames a ser feitos em laboratórios. |
| **Ator Primário** | Paciente |
| **Pré-condição** | É obrigatório que os dados sejam preenchidos com base na receita médica mais recente. |
| **Cenário Principal** | 1. O use case inicia após o usuário selecionar a opção de agendamento. 2. O sistema carrega um formulário de cadastro de horários. 3. O paciente escolhe entre a opção de agendar um evento ou atualizar evento. 4. O paciente defini os horários e dias quando necessário. 5. O sistema valida as definições. 6. O Sistema armazena os dados. 7. O sistema encerra o caso de uso. |
| **Pós-condição** | Nenhuma |
| **Cenário Alternativo** | 4a – Horário não informado.  4a.1 O sistema informa ao usuário que este campo é obrigatório e deve ser preenchido.  4a.2 Retorna ao passo 3 do cenário principal.  4b – Horário definido não existe.  4b. 1 O Sistema informa ao usuário que o horário a ser definido não existe.  4b. 2 Retorna ao passo 3 do cenário principal.  4c – Evento sem dia marcado.  4c. 1 O sistema informa ao usuário que para se cadastrar um evento deve-se definir um dia e um horário.  4c. 2 Retorna ao passo 3 do cenário principal. |

|  |  |
| --- | --- |
| **Caso de Uso –** Informar o resultado do exame de glicemia | |
| **ID** | UC 002 |
| **Descrição** | Este caso de uso tem por objetivo registrar o resultado obtido através do exame de glicemia. |
| **Ator Primário** | Paciente |
| **Pré-condição** | Deverá ser cadastrado no dia e no horário definidos pelo usuário. |
| **Cenário Principal** | 1. O use case inicia ao selecionar a opção cadastro de exames. 2. O sistema carrega o formulário de cadastro. 3. O usuário informa o dia, o horário e o resultado do exame de glicemia. 4. O sistema valida os campos. 5. O sistema armazena os dados e informa ao usuário que os dados foram salvos. 6. O sistema encerra o caso de uso. |
| **Pós-condição** | Nenhuma |
| **Cenário Alternativo** | 4a – Dia e horário não preenchidos ou inválidos  4a.1 O sistema informa que o atributo dia e horário são obrigatórios e que deveram ser preenchidos com valores válidos.  4a. 2 Retorna ao passo 3 no cenário principal.  4b – Exame de destro inválido ou não preenchido.  4b. 1 O sistema informa que o exame de destro deve ser um valor válido, e que este campo é obrigatório.  4b. 2 Retorna ao passo 3 no cenário principal. |

|  |  |
| --- | --- |
| **Caso de Uso –** Gerenciar alimentos e dosagens**.** | |
| **ID** | UC 003 |
| **Descrição** | Este caso de uso tem por objetivo gerenciar a quantidade de insulina referentes aos carboidratos de alimentos cadastrados no sistema. |
| **Ator Primário** | Paciente |
| **Pré-condição** | Nenhuma. |
| **Cenário Principal** | 1. O use case inicia após o usuário selecionar a opção de contagem de carboidratos. 2. O sistema carrega o formulário de seleção de alimentos. 3. O usuário seleciona/cadastra um alimento. 4. O sistema abre um formulário de adição de alimentos. 5. O usuário preenche os campos destacando quais alimentos foram consumidos e sua respectiva quantidade. 6. O sistema valida os campos. 7. O sistema calcula a quantidade total de carboidratos informadas armazena os dados obtidos. 8. O sistema verifica se o paciente faz uso de insulina injetável, caso faça o valor obtido dos carboidratos é convertido para unidades do medicamento. 9. O sistema informa ao usuário que foi concluído essa etapa com sucesso. 10. O sistema encerra o caso de uso. |
| **Pós-condição** | Nenhuma |
| **Cenário Alternativo** | 6a – Alimento não cadastrado.  6a.1 O sistema informa ao usuário que o alimento informado não está cadastrado no banco de dados.  6a.2 Retorna para o passo 3 do cenário atual. |

|  |  |
| --- | --- |
| **Caso de Uso – Exibir sugestões.** | |
| **ID** | UC 004 |
| **Descrição** | Este caso de uso tem por objetivo sugerir a quantidade correta de da medicação a ser utilizada. |
| **Ator Primário** | Paciente |
| **Pré-condição** | Os dados serão apresentados somente após ser executado pelo menos o UC002. |
| **Cenário Principal** | 1. O use case inicia após o usuário informar o resultado do exame de destro e/ou os alimentos consumidos. 2. O sistema valida os campos obrigatórios 3. O sistema deverá calcular a quantidade de insulina/comprimido, tendo como referência o resultado do destro e a quantidade de carboidratos ingeridos. 4. O sistema sugere a quantidade de medicamento. |
| **Pós-condição** | Nenhuma. |
| **Cenário Alternativo** | 2a – Resultado do destro não informados  2a. 1 O sistema informa que o atributo resultado de destro é obrigatório.  2b. 2 Retorna ao passo 1 no cenário principal |

|  |  |
| --- | --- |
| **Caso de Uso –** Registrar/Confirmar aplicação | |
| **ID** | UC 005 |
| **Descrição** | Este caso de uso tem por finalidade fazer o cadastro das quantidades das medicações feitas pelo usuário |
| **Ator Primário** | Paciente |
| **Pré-condição** | Deve ter sido apresentado a sugestão ao usuário. |
| **Cenário Principal** | 1. O use case inicia após o sistema exibir a sugestão ao usuário.   1. O sistema abre um formulário de observações. 2. O usuário, anota a quantidade que foi aplicada caso seja diferente da sugestão. 3. O sistema carrega uma caixa de confirmação. 4. O usuário confirma a aplicação. 5. O sistema salva os dados. 6. O sistema encerra o caso de uso. |
| **Pós-condição** | Nenhuma |
| **Cenário Alternativo** | 4a – Dados não confirmados.  4a.1 O sistema encerra o caso de uso e retorna uma tela de notificação. |

|  |  |
| --- | --- |
| **Caso de Uso –** Registrar atividades físicas. | |
| **ID** | UC 006 |
| **Descrição** | Este caso de uso tem por objetivo monitorar as atividades físicas realizadas pelo usuário. |
| **Ator Primário** | Paciente |
| **Pré-condição** | Nenhuma. |
| **Cenário Principal** | 1. O use case inicia após o sistema alertar o usuário de suas atividades físicas. 2. O sistema carrega o formulário de cadastro. 3. O usuário informa qual atividade física foi realizada (caminhada, corrida, futebol, luta, etc.) e também informa o período que foi realizada a atividade. 4. O sistema valida os campos obrigatórios. 5. O sistema salva os dados no banco de dados e informa ao usuário que os dados foram salvos. 6. O sistema encerra o caso de uso. |
| **Pós-condição** | Nenhuma |
| **Cenário Alternativo** | 1a – Usuário ignora o alerta.  1a – O sistema encerra o caso de uso.  4a – Campos Atividade realizada não preenchido.  4a.1 O sistema informa ao usuário que este campo é obrigatório e deve ser preenchido.  4a.2 Retorna ao passo 3 do cenário principal. |

|  |  |
| --- | --- |
| **Caso de Uso – Exibir relatório** | |
| **ID** | UC 007 |
| **Descrição** | Este caso de uso tem por objetivo permitir que o usuário possa monitorar a doença. |
| **Ator Primário** | Paciente |
| **Pré-condição** | Deverá conter pelo menos um dado cadastrado. |
| **Cenário Principal** | 1. O use case inicia após o usuário selecionar a opção relatório. 2. O sistema valida os dados que deveram conter no relatório. 3. O sistema gera o relatório. 4. O sistema encerra o caso de uso. |
| **Pós-condição** | Nenhuma |
| **Cenário Alternativo** | 2a – Nenhum exame de destro registrado.  2a.1 O sistema informa ao usuário que é necessário ter pelo menos um exame de destro cadastrado para gerar o relatório  2a.2 Retorna ao passo 1 do cenário principal. |

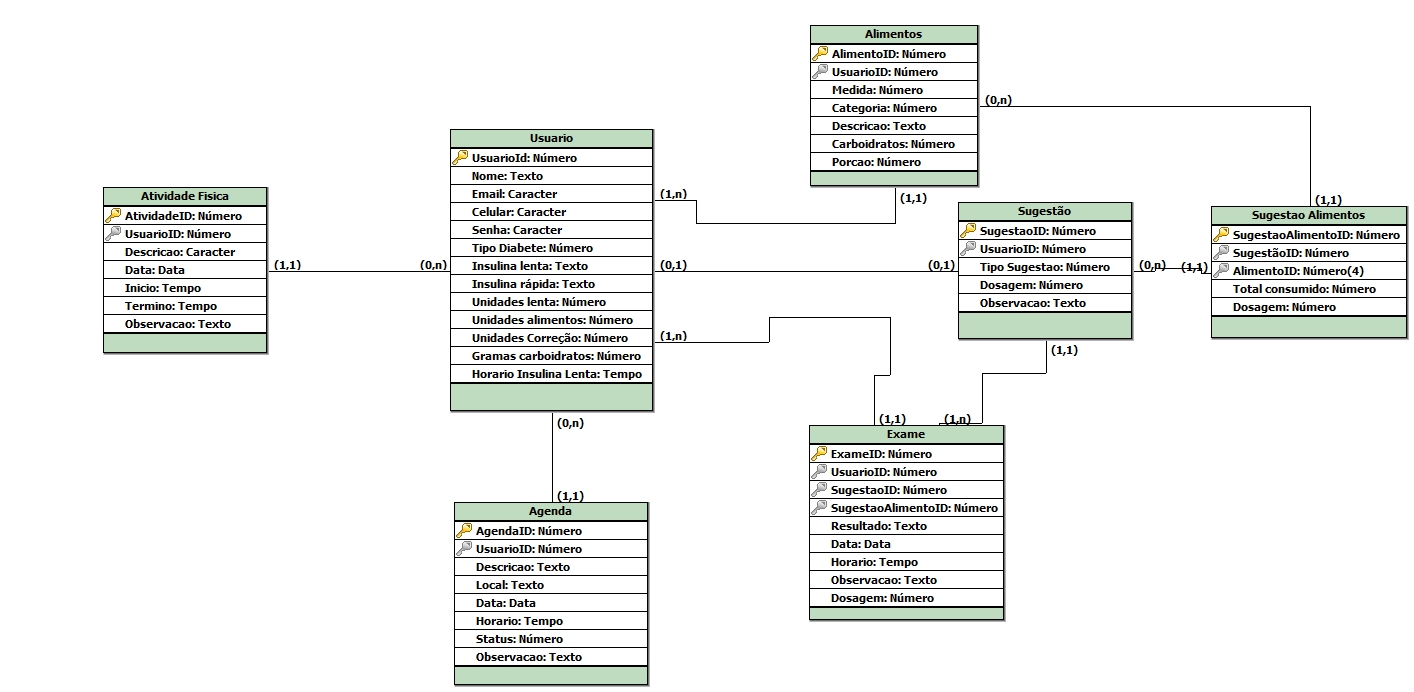
**Quadro 6** – Matriz de rastreabilidade RF x UC

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | RF 1 | RF 2 | RF 3 | RF 4 | RF 5 | RF 6 | RF 7 | RF 8 | RF 9 | RF 10 | RF 12 |
| UC1 |  | X | X |  | X |  | X | X |  |  | X |
| UC2 | X |  |  |  |  |  |  |  |  |  |  |
| UC3 |  |  | X |  |  | X | X | X |  |  |  |
| UC4 |  |  |  |  |  |  |  | X |  |  |  |
| UC5 |  |  | X |  | X |  | X | X | X |  |  |
| UC6 |  |  |  |  |  |  |  |  | X |  |  |
| UC7 |  |  |  |  |  |  |  |  |  |  | X |

2.6 Diagrama Entidade-Relacionamento

Diagrama que representa a modelagem do banco de dados.

**Figura 4:** Diagrama Entidade-Relacionamento



**3 Ferramentas e Métodos ou Desenvolvimento**

3.1 Ferramentas

3.1.1 Implementação

A linguagem de programação escolhida para o desenvolvimento do projeto foi a C#, utilizando-se a IDE Visual Studio Community 2019, por facilitar os back-ups incrementaís feitos no projeto através da plataforma Git-Hub, além de propiciar a instalação completa do framework Xamarin-Forms na versão 5.0.0.2115.

A escolha do framework Xamarin-Forms se tornou necessária para distribuir a mesma linha de código para diversas plataformas diferentes, sendo elas o sistema operacional Windows 10, android e IOS, tendo o principal foco na plataforma android.

Foi utilizado o banco de dados SqLite na versão 1.8.116, que possibilita a gravação dos dados no próprio dispositivo do usuário, dessa forma não sendo necessário a utilização de internet para usufruir dos benefícios do sistema, sendo está a principal razão para o seu uso.

3.1.2 Línguagem de programação C#.

C# é uma linguagem orientada a objetos desenvolvida por Anders Hejlsberg para a Microsoft, fazendo parte da aplicação .NET. Em primeira instância a linguagem foi batizada como Cool no final do século XX, mais tarde após a virada do século a linguagem Cool foi renomeada para C#.

Após o renome da linguagem, a Microsot submeteu-se à ECMA(European Computer Manufacturers Association), associação cujo objetivo é a padronização de sistemas de informação. Em 2001, a ECMA aprovou o C# e a linguagem recebeu a especificação ECMA-334. Mais tarde, em 2003, tornou-se padrão da ISSO, recebendo a especificação de ISSO/IEC 23270 (COIMBRA DE ARAÚJO, 2013).

O C# surgiu com o propósito de tornar menos rígido o desenvolvimento de aplicativos, a linguagem foi baseada nas linguagens de programação, Java, C e C++, juntando diversos recursos de cada linguagem citada, se tornando mais simples e flexível introduzindo diversos elementos como: expressões lambda, tipos primitivos com valores nulos, delegações e acessos diretos a memória.

Em relação a linguagem C, o C# tem foco na compilação de soluções de alto nível, já a linguagem C é voltada para o desenvolvimento de baixo nível. Outras comparações, de acordo com MSDN (2012), o C#, simplifica de modo significativo a complexidade do C++ e introduz novos elementos não disponíveis no Java, tais como: tipos primitivos não nulo, delegações, expressões lambda e acesso direto à memória.

MSDN (2012) ressalta que, o processo de compilação do C# é mais simples e mais flexível comparado ao C++ ou Java, pois elimina-se a necessidade de arquivos de cabeçalhos separados e não há a necessidade da declaração de métodos e tipos em uma ordem específica. Um arquivo contendo códigos de programação podem ser definidos quaisquer números de classes, estruturas, interfaces e eventos que forem necessários. (COIMBRA DE ARAÚJO, 2013).

3.1.3 Banco de dados SQLite.

Na prática o SQLite se assemelha à um Sistema de Gerenciamento de Banco de Dados, possibilitando fazer inserções, alterações e exclusões de dados, o SQLite é formado por um conjunto de bibliotecas escritas em C, com a finalidade de ser inserido em diversos programas escritos em diversas linguagens diferentes.

A principal diferença com outros bancos de dados, é que a manipulação dos dados pode ser feita sem que seja preciso acessar um SGBD (sistema de gerenciamento de banco de dados), ou seja, todas as instruções podem ser feitas através do código fonte, inclusive a criação do banco de dados e tabelas, sendo esse o principal motivo por ser implementada em aplicativos móveis (Android, IOS).

Além disso, o Android oferece suporte completo ao banco, através de uma API com um rico conjunto de classes e métodos que abstraem as complexidades dos códigos SQL. Assim, não precisamos montas a cláusula SQL inteira para atualizar uma linha na tabela, ou ainda, para fazer uma pesquisa na mesma. O Android nos fornece um método, onde passando alguns parâmetros obtemos um apontador para os dados retornados, podendo navegar pelo resultado como se estivéssemos escolhendo uma folha em um arquivo. (CRIS BRITO, 2015).

3.1.3 Xamarin.Forms.

O Xamarin é uma plataforma de desenvolvimento utilizada para criar sistemas em diversas plataformas diferentes, o framework é uma extensão da plataforma de desenvolvimento .NET que é composta por diversas bibliotecas, ferramentas e linguagens de programação para desenvolver diversos aplicativos.

O Xamarin.Forms trata-se de uma estrutura do Xamarin, que tem por finalidade disponibilizar bibliotecas e ferramentas para o desenvolvimento de aplicativos móveis. O Xamarin.Forms é uma estrutura de código aberto da Microsoft, tendo seu principal objetivo a criação de aplicativos para Android, IOS e sistemas operacionais Windows através do .NET, através de uma única base de códigos compartilhados.

3.2 Métodos ou Desenvolvimento

Foi utilizado o patern MVVM (Model-View-Viewmodel) para o desenvolvimento do aplicativo, esse patern foi criado em 2005 por John Gossman, sendo ele um dos arquitetos do WPF e Silverlight da Microsoft. O partner se assemelha ao padrão MVC (Model-View-Controller), contudo o MVVM procura estabelecer uma separação mais apurada de responsabilidades em uma aplicação WPF e Silverlight.

Em termos gerais a View é responsável por definir o front-end do sistema, que faz uma interligação com a ViewModel através de um componente denominado de DataContext. A ViewModel tem como sua principal responsabilidade apresentar uma lógica de apresentação para a View, sendo assim fazendo a interligação entre a View e a Model. A Model tem a responsabilidade de encapsular a lógica e dados, são as classes que são utilizadas no desenvolvimento do sistema.

**Figura 5 –** View

Texto

Descrição gerada automaticamente

**Fonte:** Autor

A figura a cima mostra um breve exemplo da construção do front-end em uma View, nesse exemplo nos mostra a construção da tela de login, onde é possível perceber na linha 15 a manipulação de imagem, sendo definido altura e largura, também nota-se que todos elementos estão contidos dentro de um frame, que permitir destacar toda área contida nele, também causando efeitos 3D nas plataformas Android e IOS.

A próxima figura nos mostra um exemplo de uma ViewModel. É possível perceber na linha 18 a interface INotifyPropertyChanged, que é responsável por proporcionar um mecanismo unificado para implementar um único evento para todos os eventos dos objetos utilizados, sendo assim apresenta uma interação de forma rápida e eficiente ao usuário. Como exemplo temos as validações que são feitas de forma dinâmica apresentando a mensagem de erro e impossibilitando o envio do formulário antes mesmo do usuário clicar no botão de confirmação.

**Figura 6 –** ViewModel

Texto

Descrição gerada automaticamente

**Fonte:** Autor

A seguinte figura nos mostra um exemplo de uma model, que é responsável por encapsular todos os dados utilizados. Através da model definimos qual tabela será utilizada no banco de dados e quais são as entidades.

**Figura 6 –** Model

Texto

Descrição gerada automaticamente

**Fonte:** Autor

Para que fosse possível integrar a mesma linha de código para diversas plataformas diferentes foi-se implementado uma interface, com a finalidade de direcionar os dados para a plataforma correta.

**Figura 6 –** Interface

Texto

Descrição gerada automaticamente

**Fonte:** Autor

A interface se comunica com as classes contidas em cada plataforma, dessa maneira possibilitando a reutilização do mesmo código, em cada classe é efetuado o código para cada tipo de plataforma retornando um resulta, no caso acima temos a manipulação de dados no banco de dados SQLite.

**Figura 6 –** Classe Da plataforma Android

Texto

Descrição gerada automaticamente

**Fonte:** Autor

**4 Resultados e Discussão** (este item é obrigatório)

Apresentar *prints* do sistema, explicando cada funcionalidade que foi implementada. Caso o sistema tenha sido implantado em algum usuário, coletar e descrever informações sobre o processo de implantação e os benefícios levantados pelo usuário sobre a utilização do software.

**Considerações finais**

Relembrar quais foram objetivos iniciais, o que foi de fato desenvolvido, quais foram os principais desafios e quais serão os projetos futuros que poderão ser realizados.

**Referências**

BRASIL. Ministério da Saúde. **Diabetes Mellitus.** Brasília, 2006.

BRASIL. Ministério da Saúde. **Relatório de Recomendações.** Brasília, 2009.

BRASIL. Ministério da Saúde. **RASTREAMENTO E DIAGNÓSTICO DE DIABETES MELLITUS GESTACIONAL NO BRASIL**. Brasília, 2017.

**Manual oficial de contagem de carboidratos regional** / Sociedade Brasileira de Diabetes, Departamento de Nutrição. – Rio de Janeiro: Dois C: Sociedade Brasileira de Diabetes, 2009 il.

COIMBRA DE ARAÚJO, Everson. **Artigo Invista em você! Saiba como a DevMedia pode ajudar sua carreira. A evolução da linguagem de programação C#**. [*S. l.*], 2013. Disponível em: https://www.devmedia.com.br/a-evolucao-da-linguagem-de-programacao-csharp/28639. Acesso em: 15 out. 2021.

CRIS BRITO, Robson. **Utilizando SQLite em aplicativos Android**. [*S. l.*], 2015. Disponível em: https://www.devmedia.com.br/utilizando-sqlite-em-aplicativos-android/32117. Acesso em: 15 out. 2021.