

INSTITUTO FEDERAL DO RIO GRANDE DO NORTE - IFRN
CAMPUS PARNAMIRIM - IFPAR
COORDENAÇÃO DO CURSO DE TECNOLOGIA EM SISTEMAS PARA
INTERNET
SEMINÁRIO DE ORIENTAÇÃO DE PROJETO INTEGRADOR
Docente: Valério Gutemberg de Medeiros Júnior

Dia - a - Dia:

Aplicação de controle Glicêmico.

Luiz Gustavo da Rocha Ferreira

Parnamirim - RN julho/2023

1. ABSTRACT:	4
2. RESUMO:	5
3. INTRODUÇÃO:	7
2.1 JUSTIFICATIVA:	7
2.2 OBJETIVO:	7
2.3 PÚBLICO-ALVO:	7
4. Desenvolvimento:	9
4.1. ORDEM DE PRIORIDADE DO DESENVOLVIMENTO.	11
4. FUNCIONAMENTO:	13
4. 1. DESCRIÇÃO DOS USUÁRIOS:	13
4. 1.1. Dados de Entrada:	14
4. 1.2. Processamento dos Dados:	14
4. 1.3. Saída de Dados:	14
4. 1.4, Resultados Desejados:	15
4. 1.5. Resultados Indesejados:	15
5. REQUISITOS:	16
5. 1. Levantamento de Dados:	16
5. 2. Sobre as Técnicas de Levantamento de Dados.	17
5.3. 1. Requisitos Funcionais:	18
6. REQUISITOS NÃO-FUNCIONAIS.	19
7. DIAGRAMAS.	20
7.1. DIAGRAMA DE USO DE CASO:	20
7.2. DIAGRAMA DE COMPONENTES:	21
7.2.1 DIAGRAMA USUÁRIO-MÉDICO:	21
7.2.2 DIAGRAMA USUÁRIO-PACIENTE:	22
7.2.3 DIAGRAMA USUÁRIO-CLÍNICA:	23
7.3. DIAGRAMA DE ATIVIDADE:	24
7.4 DIAGRAMA DE CLASSE:	25
8.PROTOTIPAÇÃO:	26
8.1. FRONT END LOGIN USUÁRIO CLÍNICA.	26
8.1. FRONT END DASHBOARD USUÁRIO CLÍNICA.	27
8.2. BACK END	28

8.2.1 API	28
8.3 BANCO DE DADOS	29
8.3.1 DIAGRAMA DE DESENVOLVIMENTO DE BANCO DE DADOS.	29
8.3.2 SCRIPT DO BANCO DE DADOS COM MYSQL.	30
9. RESULTADOS:	30
CONCLUSÃO:	32
REFERÊNCIA:	33

1. ABSTRACT:

The "Dia-a-Dia" project is a mobile application developed to assist in the monitoring and management of diabetes, with the aim of improving the quality of life for patients. The application allows users to record and organize their daily blood glucose data, creating a comprehensive mapping of health information. With a focus on usability and functionality, the app enables users to track their glucose levels, meals, and physical activities. It also provides analysis and visualization features, including graphs and reports, to aid users and healthcare professionals in monitoring and making informed decisions regarding treatment. Developed using multi-platform frameworks such as Ionic and Flutter, the application ensures compatibility with Android, iOS, and web platforms. Integration with glucose monitoring devices, personalized notifications, and secure data storage further enhance the user experience. The "Dia-a-Dia" project aims to provide diabetic patients with a comprehensive and accessible tool for disease monitoring, simplifying daily glucose management, and promoting effective treatment outcomes.

2. RESUMO:

O projeto Dia-a-Dia é uma aplicação multiplataformas desenvolvida para auxiliar no monitoramento e gerenciamento da diabetes, visando melhorar a qualidade de vida dos pacientes. Através dessa aplicação, os usuários podem registrar e organizar seus dados glicêmicos diários, criando um mapeamento detalhado de suas informações de saúde.

A finalidade principal do Dia-a-Dia é oferecer aos pacientes diabéticos uma ferramenta prática e intuitiva para registrar suas medições de glicemia, medicamentos e outros fatores relevantes. A aplicação permite acompanhar as variações glicêmicas ao longo do tempo, identificando padrões e tendências, e fornecendo um feedback personalizado para auxiliar na tomada de decisões relacionadas ao tratamento.

As funcionalidades principais do aplicativo incluem o registro de glicemias, permitindo inserir valores obtidos por meio de medidores de glicose. A aplicação também oferece recursos de análise e visualização dos dados registrados, como gráficos e relatórios, que auxiliam tanto os pacientes no autocontrole quanto os médicos no acompanhamento do tratamento.

Em termos de tecnologias utilizadas, o "Dia-a-Dia" é uma aplicação multiplataforma desenvolvida com Ionic e Flutter. Ela utiliza MySQL, FastAPI, Matplotlib e Pandas. Essa solução permite a criação de aplicativos nativos para Android e iOS a partir de um único código-fonte, além de contar com uma interface web para Desktop. A aplicação se destaca pela integração com dispositivos de monitoramento de glicemia, oferecendo a importação automática de dados, notificações personalizadas para lembrar sobre medições e medicamentos, e um armazenamento seguro dos dados do usuário.

Em suma, o Dia-a-Dia busca proporcionar aos pacientes diabéticos uma ferramenta completa e acessível para o monitoramento de sua doença, facilitando o

gerenciamento diário da glicemia e promovendo uma maior integração entre os dados do usuário e os profissionais de saúde, visando um tratamento mais eficaz e resultados positivos.

3. INTRODUÇÃO:

2.1 JUSTIFICATIVA:

O projeto Dia-a-Dia surge no contexto de uma sociedade que enfrenta desafios relacionados à diabetes e à qualidade da alimentação. Com o aumento significativo dos casos de diabetes, conforme relatado pela CNN (BARRETO, 2021), torna-se essencial buscar soluções para auxiliar no monitoramento e tratamento dessa doença crônica.

A má alimentação é uma das possíveis causas dos maus hábitos que contribuem para o aumento dos casos de diabetes. Diante desse cenário, é importante conscientizar a população sobre a importância de hábitos alimentares saudáveis e oferecer ferramentas que facilitem o monitoramento da glicemia e o controle da doença.

2.2 OBJETIVO:

O objetivo principal do projeto Dia-a-Dia é desenvolver uma aplicação multiplataforma que permita aos usuários monitorar e organizar seus dados glicêmicos diários. Através dessa ferramenta, busca-se estabelecer um mapeamento detalhado das informações de saúde dos pacientes, auxiliando-os no autogerenciamento da diabetes e proporcionando uma melhor qualidade de vida.

2.3 PÚBLICO-ALVO:

O público-alvo do projeto Dia-a-Dia são os pacientes diabéticos, tanto jovens como adultos, os casos clínicos classificados como diabetes tipo 1, tipo 2 e gestacional, que necessitam de um acompanhamento constante da glicemia e de uma organização eficiente dos dados relacionados à sua saúde. Além dos pacientes,

profissionais da área da saúde, como médicos e enfermeiros, também poderão se beneficiar da aplicação, pois ela possibilita um acompanhamento mais preciso do tratamento e o compartilhamento de informações relevantes, seja por meio de relatórios produzidos na própria aplicação, ou até mesmo por monitoramento nos dados fornecidos pela aplicação.

Ao disponibilizar uma aplicação prática, intuitiva e completa, o projeto Dia-a-Dia visa contribuir para o controle e tratamento adequado da diabetes, promovendo uma maior conscientização sobre a importância de hábitos saudáveis e um monitoramento eficiente da glicemia. Possibilitando diagnósticos mais precisos e promovendo melhor qualidade de vida para seus usuários.

4. Desenvolvimento:

O projeto Dia-a-Dia está sendo desenvolvido seguindo uma metodologia ágil, que permitiu uma abordagem iterativa e incremental para a criação da aplicação. A metodologia adotada envolveu as seguintes etapas:

1. Análise de requisitos: Inicialmente, foram realizadas análises detalhadas dos requisitos funcionais e não funcionais da aplicação. Isso incluiu a identificação das principais funcionalidades, interfaces e integrações necessárias para atender às necessidades dos usuários.

2. Projeto de interface: A partir dos requisitos estabelecidos, foi realizado o projeto de interface da aplicação, considerando a usabilidade, a navegabilidade e a experiência do usuário. Foram criados protótipos e fluxos de interação para garantir uma interface intuitiva e amigável.

3. Desenvolvimento: As tecnologias Ionic e Flutter para criar a aplicação multiplataforma. O Ionic foi utilizado para a criação da interface mobile, enquanto o Ionic foi empregado para o desenvolvimento da interface web para Desktop. Foram utilizadas as linguagens HTML, CSS e JavaScript para a criação das telas e lógica de programação.

4. Integrações e recursos: Nessa etapa a proposta é implementar as integrações com dispositivos de monitoramento de glicemia, para que seja possível a importação automática de dados para a aplicação. Também serão desenvolvidas funcionalidades utilizando o MySQL como banco de dados - ainda em fase de desenvolvimento.

5. Testes e validação: Ao longo do processo de desenvolvimento, serão realizados testes unitários e testes de integração para garantir o bom funcionamento da aplicação. Além disso, serão conduzidos testes de usabilidade com usuários reais, permitindo a identificação e correção de possíveis problemas e ajustes na interface e nas funcionalidades.

6. Entrega e avaliação: Após a conclusão do desenvolvimento e dos testes, a aplicação Dia-a-Dia será entregue aos usuários para avaliação e feedback. Com base

nesses retornos, serão realizadas melhorias e ajustes finais antes da disponibilização do produto final.

Para auxiliar no desenvolvimento, estão sendo utilizadas ferramentas como ambientes de desenvolvimento integrado (IDEs), ferramentas fornecidas pela JetBrains, as quais são:

1. Webstorm, para desenvolvimento de interface e aplicação web;
2. Datagrip, para desenvolvimento de banco de dados;
3. Dataspell, para análise de dados e produção de relatórios e formato de tabelas dos mais diversos formatos entre eles csv e excel;
4. Android Studio, para a codificação e depuração do código. Também foram utilizadas bibliotecas e frameworks, como Matplotlib e Pandas, para análise e visualização de dados.
5. Pycharm Professional, para desenvolvimento de gráficos e “plotagem” de gráficos e API;
6. Fastapi: framework web de alto desempenho e fácil desenvolvimento, baseado na linguagem de programação Python fornecendo uma estrutura sólida para a criação de endpoints RESTful, permitindo o processamento de requisições e respostas de forma eficaz. Considerado a integração do Fastapi com outras bibliotecas e ferramentas do ecossistema Python.
7. Mysql: Com confiabilidade, desempenho robusto e recursos avançados de segurança, o MySQL permite o armazenamento seguro e eficiente dos dados glicêmicos. Compatibilidade com as tecnologias utilizadas e escalabilidade.
8. Matplotlib: Biblioteca utilizada para visualização de dados no Python. No projeto Dia-a-Dia, o Matplotlib desempenha um papel importante ao permitir a criação de gráficos e visualizações dos dados glicêmicos dos pacientes diabéticos;
9. Ionic: Framework de desenvolvimento de aplicativos móveis multiplataforma que utiliza tecnologias web, como HTML, CSS e JavaScript, para criar aplicativos nativos para Android, iOS e Desktop. No projeto Dia-a-Dia, o Ionic desempenha um papel central, permitindo a criação de uma aplicação mobile e desktop intuitiva e responsiva. Além disso, a capacidade de desenvolver para múltiplas

plataformas a partir de um único código-fonte é extremamente vantajosa, possibilitando uma maior eficiência no desenvolvimento e manutenção da aplicação.

10. O Flutter é um framework de desenvolvimento de aplicativos multiplataforma que utiliza a linguagem Dart. No projeto Dia-a-Dia, o Flutter desempenha um papel crucial ao permitir a criação de interfaces nativas e responsivas para Android, iOS e web a partir de um único código-fonte, garantindo uma experiência consistente e eficiente para os usuários.

A metodologia ágil adotada permitirá um desenvolvimento mais flexível e adaptável, garantindo a entrega de um produto de qualidade que atende às necessidades dos usuários.

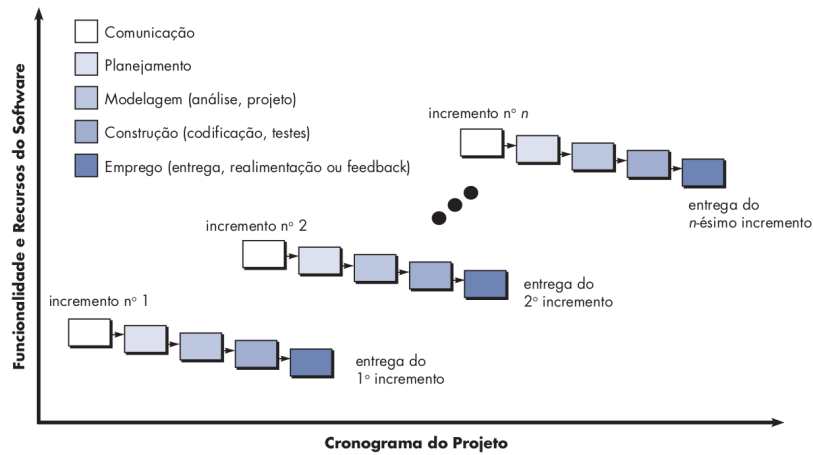
4.1. ORDEM DE PRIORIDADE DO DESENVOLVIMENTO.

Compreendendo que a objetividade da aplicação é bastante específica, sem que haja a necessidade de grandes mudanças com relação à proposta inicial, o modelo de desenvolvimento que melhor se enquadra no processo de desenvolvimento do Dia-a-Dia é o Modelo Iterativo/Incremental. Segundo Pressman:

O Modelo incremental combina elementos dos fluxos de processos lineares e paralelos, à medida que o tempo vai avançando. Cada sequência linear gera “incrementais” (entregáveis/aprovados/liberados) do software de maneira similar aos incrementais gerados por um fluxos de processos evolucionários.

pg.25

segue a imagem:



Considerando que o primeiro incremento é um produto essencial. Isto é, os requisitos básicos são atendidos. Esse produto essencial é utilizado pelo cliente e como resultado do feedback se dará prosseguimento ao próximo incremento. Levando em consideração que esse modelo de desenvolvimento tem baixo custo operacional em relação às outras implementações.

4. FUNCIONAMENTO:

Após diagnosticado por seu médico, o usuário será convidado pela clínica ou mesmo pelo médico a fazer o download da aplicação, que poderá ser disponibilizada nas plataformas do Play Store (Google), Apple Store(Apple) ou disponibilizada nos sites das clínicas por meio de link para download, ou Código QR nas clínicas e consultórios.

Após fazer o login, o usuário irá fazer seu cadastro e por meio de código da clínica ele terá seu cadastro ativado junto ao seu médico. Feito esse procedimento, o usuário já terá seus dados cadastrados na clínica de sua consulta. Para fins práticos, serão descritos apenas os procedimentos de uso relacionados aos usuários-clínicos e não ao relacionado às clínicas e o cadastramento dessas, considerando que para essa finalidade a Equipe de suporte fará essa atribuição.

Diariamente os usuários poderão inserir o número de seus índices glicêmicos, deixando para a aplicação notificar ao usuário-paciente a hora de fazer a medição, seja no modelo invasivo na epiderme, ou no modelo adesivo. Após a aferição o usuário poderá digitar o valor na aplicação por meio do teclado numérico. Em seguida, a aplicação retorna uma mensagem positiva, quando o procedimento obtiver sucesso.

Já no segundo dia, assim como nos dias seguintes, a aplicação irá notificar a hora da aferição e após o procedimento acima ser concluído o usuário terá a sua disposição um gráfico que se tornará mais preciso ao passar dos dias.

Na consulta seguinte o médico poderá acessar o portal da aplicação que terá um gráfico detalhado da taxa glicêmica e a partir desses dados o médico poderá fazer o prognóstico de tratamento.

4. 1. DESCRIÇÃO DOS USUÁRIOS:

Para melhor entendimento da aplicação, consideramos os seguintes atores no

processo de fornecimento e consumo de dados.

- **usuário-paciente:** Pode ser compreendido como os indivíduos que fazem uso da aplicação de forma pessoal e personalizada;
- **usuário-médico:** Pode ser entendido como os consumidores da base de dados da aplicação, estes são os médicos que serão cadastrados e que terão acesso a base de dados usuários-paciente;
- **usuário-clínica:** Para fins de cadastramento e controle tanto dos usuários-pacientes, como dos usuários-médicos. As clínicas são importantes para fins burocráticos e de controle de consultas(Funcionalidade não disponível na aplicação).

4. 1.1. Dados de Entrada:

Para cadastro, o usuário poderá fornecer as informações pessoais, tais como: nome, endereço, telefone, e-mail, assim como informações relacionadas aos serviços de saúde, medidas, e índices ligados triglicerídeos assim por diante.

Em relação ao uso de rotina, o usuário terá apenas que informar sua taxa glicêmica diária.

4. 1.2. Processamento dos Dados:

Após a confirmação referente ao cadastro, os dados do usuário serão enviados para o banco de dados que armazenará todas as informações referentes ao controle do índice de glicemia na corrente sanguínea, na aplicação do Dia-a-Dia terá um gráfico que considerará o índice de açúcar e o dias aferições.

4. 1.3. Saída de Dados:

Com base nos dados obtidos a aplicação terá duas saídas de dados, uma referente ao gráfico que poderá ser acessado na própria aplicação e outra saída,

nesse caso, clínica que será acessada pelo médico no exame de rotina, ou em algum caso específico.

4. 1.4, Resultados Desejados:

Considerando que o usuário-clínico corresponda perfeitamente a relação de uso de 100% (cem por cento) de uso, os dados esperados serão um base de dados com os diários de aferição, consideram que a medida aceita pela OMS referente a tabela:

Glicemia de jejum normal: inferior a 99 mg/dL;
Glicemia de jejum alterada: entre 100 mg/dL e 125 mg/dL;
Diabetes: igual ou superior a 126 mg/dL;
Glicemia de jejum baixa ou hipoglicemia: igual ou inferior a 70 mg/dL.

4. 1.5. Resultados Indesejados:

Considerando o pior cenário, o usuário não faz uso da aplicação no Dia-a-Dia, ela notificará a clínica, caso o usuário entre em consentimento. Assim, o usuário-paciente poderá ser notificado por meio de outras aplicações, por exemplo, WhatsApp a respeito da importância do controle glicêmico diário.

5. REQUISITOS:

5. 1. Levantamento de Dados:

Para o levantamento de Dados, considerando o público que irá consumir a aplicação elencamos o seguinte panorama:

- **Sobre o usuário-paciente:**
 - **Sobre o público:**

A aplicação terá objetivo clínico, deve-se considerar que a usabilidade da aplicação deve ser abrangente e o público alvo é composto por uma escala etária bastante ampla, compreendendo que a Diabetes é uma doença que atinge todas as idades.
 - **sobre as funcionalidades:**

A aplicação deve ser rápida, sua utilidade não pode ser custosa na relação uso e tempo de uso.
- **Da utilização dos dados:**
 - **Com base nos usuários-médicos:**

A aplicação deve considerar a relação entre usuários-médicos e quais características possam contribuir para o uso efetivo da aplicação. Assim, será considerado para a plataforma as funcionalidades que mais são adequadas pelos especialistas da saúde.
 - **com base nos usuários-clínicas:**

Aplicação deve ser estabelecer uma relação positiva entre usuário-clínico e a interface para a base de dados. Considerando a relação tempo de entrada de dados e tempo de usabilidade desses dados.

5. 2. Sobre as Técnicas de Levantamento de Dados.

Considerando os elementos anteriores, faz-se necessário aplicar as seguintes técnicas para levantamento de dados. Os dados serão levantados de forma híbrida. Contudo as técnicas abaixo serão as mais pertinentes para a viabilidade da aplicação.

1. Entrevista:

2. Observação:

3. Análise de Conteúdo:

O modelo de levantamento consiste em estabelecer um possível diálogo com os três grupos de usuários a fim de evitar funcionalidades indesejadas. Para que esse modelo seja efetivo, considerando as desvantagens existentes nesse modelo de levantamento, serão considerados os elementos julgados mais importantes na aplicação, no que diz respeito à Observação, considerando a rotina clínica como modelo a ser observado. Analisar de forma “natural” como ocorre o fluxo de dados e como os dados clínicos se tornam pertinentes para o controle usuário-médico. Por último e não menos importante, a Análise de Conteúdo servirá de modelo comparativo, considerando o grande número de aplicações relacionadas à saúde. O feedback nas plataformas de downloads servirão de espelho para observar as relações usuário-paciente e usuário-médico.

O modelo observado é a aplicação Glic, que faz a medição dos dados Glicêmicos e por meio de uma comunidade de médicos e nutricionistas cadastrados faz prognósticos para os portadores de Diabetes.

5.3. 1. Requisitos Funcionais:

Considerando as descrições dos serviços que serão fornecidos pelo sistema e refletem as necessidades dos clientes.

Usuário - Paciente:

A aplicação permitirá que o usuário-paciente faça seu cadastro, o qual deverá conter, nome completo, telefone email e tipo sanguíneo e tipo de diabetes, após cadastro ativado o usuário-paciente poderá fornecer seus dados glicêmicos de acordo com as recomendações médicas ou determinação médica. Ao usuário-cliente fica determinado fornecer os dados para a construção de uma tabela gráfica de acordo com os dados fornecidos.

Usuário - Médico:

A aplicação dará acesso a esse usuário para fazer consulta a respeito de um **usuário-paciente**, assim como, sua frequência e assiduidade com o fornecimento de dados. As informações que estarão disponíveis aos médicos serão preferencialmente relacionadas à proposta da aplicação em si.

Usuário - Clínica:

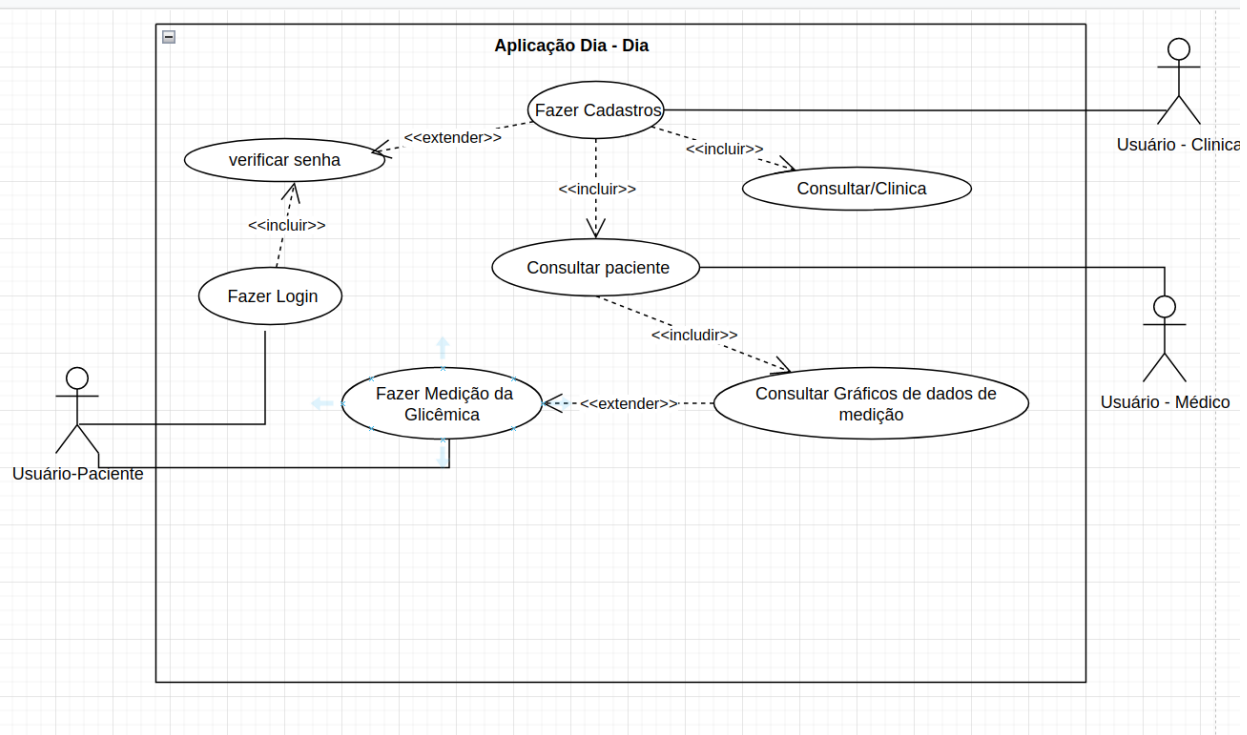
A aplicação fornecerá acesso aos dados dos **usuários-pacientes**, seja dados clínicos ou médicos, cabendo as clínicas a seleção **usuários-pacientes** para o grupo atendido pelos **usuário-médico**. Aos **usuários - clínicos** ficam as atribuições de marcação de consultas e a notificação aos pacientes a respeito dos exames solicitados e fornecimento de dados necessários para as consultas médicas.

6. REQUISITOS NÃO-FUNCIONAIS.

- Confiabilidade;
- Usabilidade;
- Escalabilidade;
- Portabilidade;
- Interoperabilidade;
- Legais;

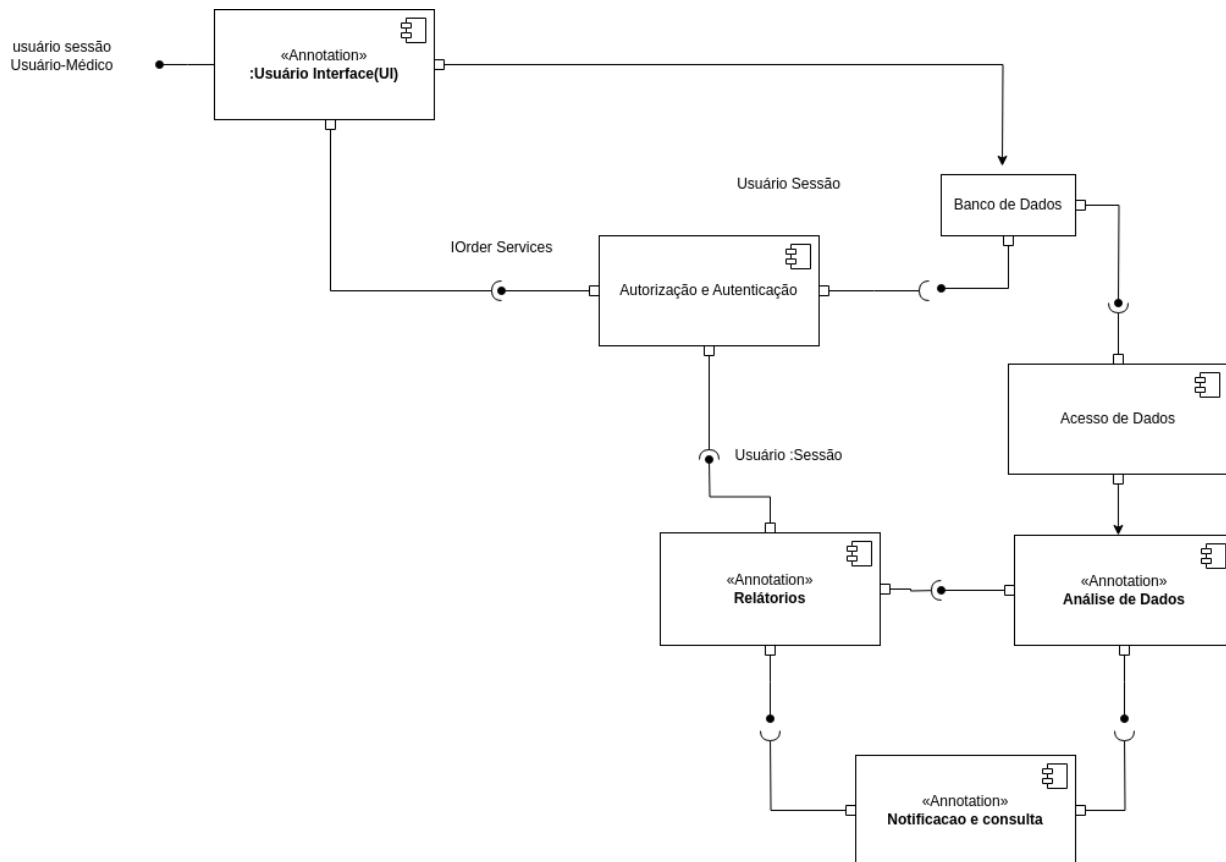
7. DIAGRAMAS.

7.1. DIAGRAMA DE USO DE CASO:

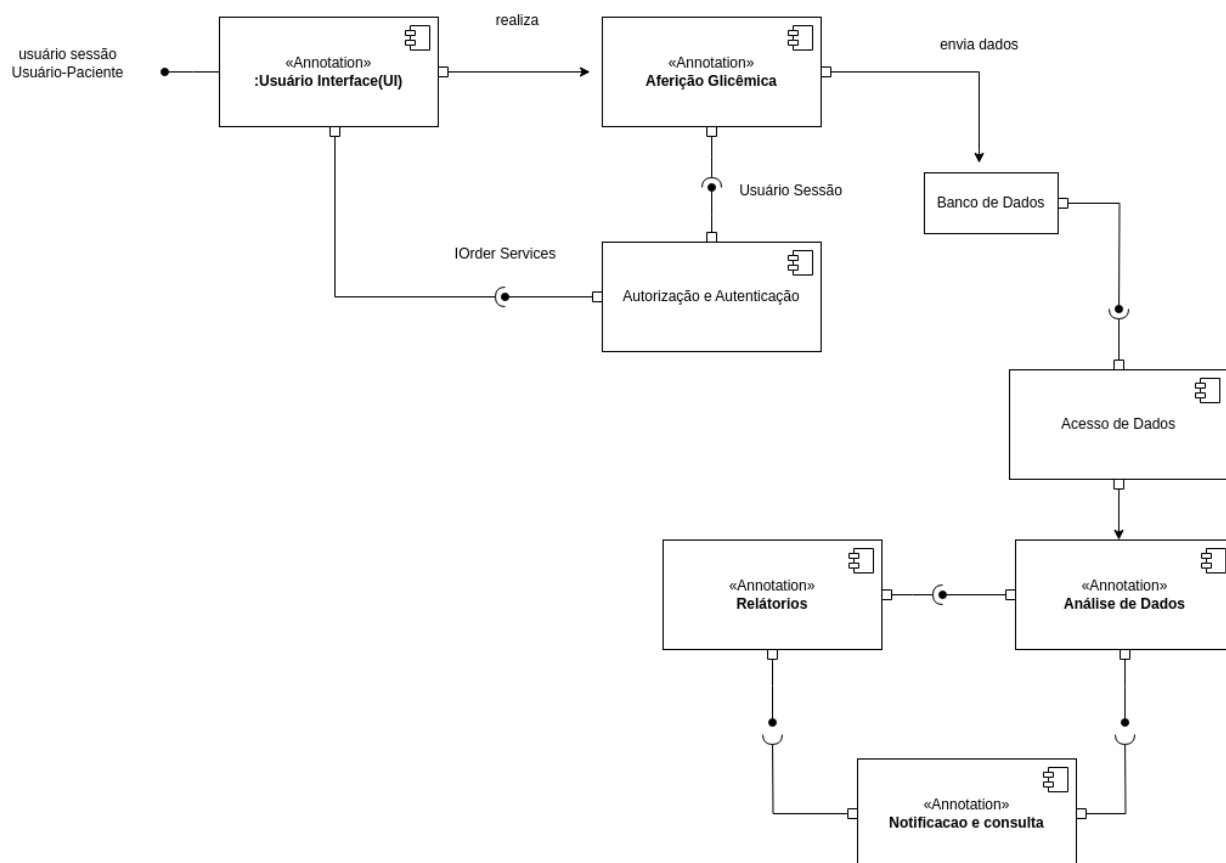


7.2. DIAGRAMA DE COMPONENTES:

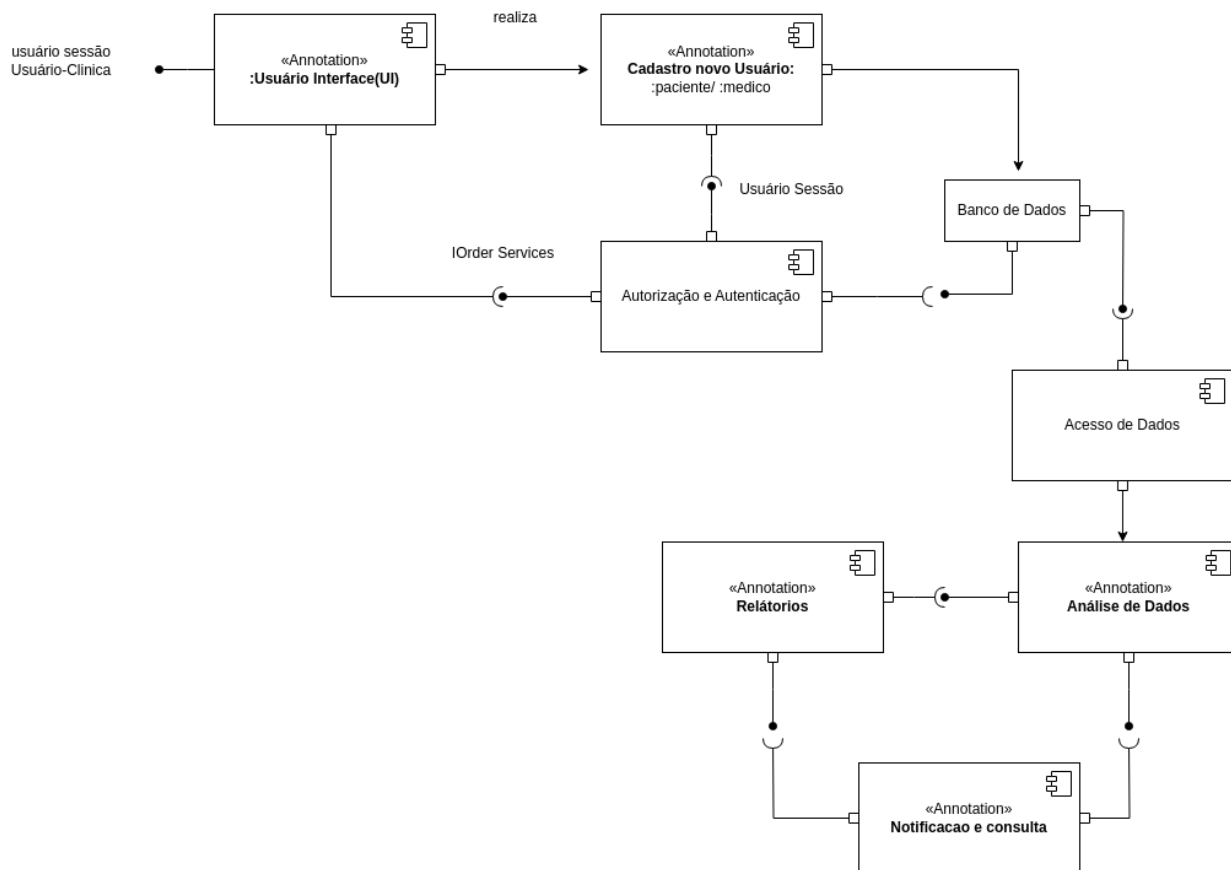
7.2.1 DIAGRAMA USUÁRIO-MÉDICO:



7.2.2 DIAGRAMA USUÁRIO-PACIENTE:

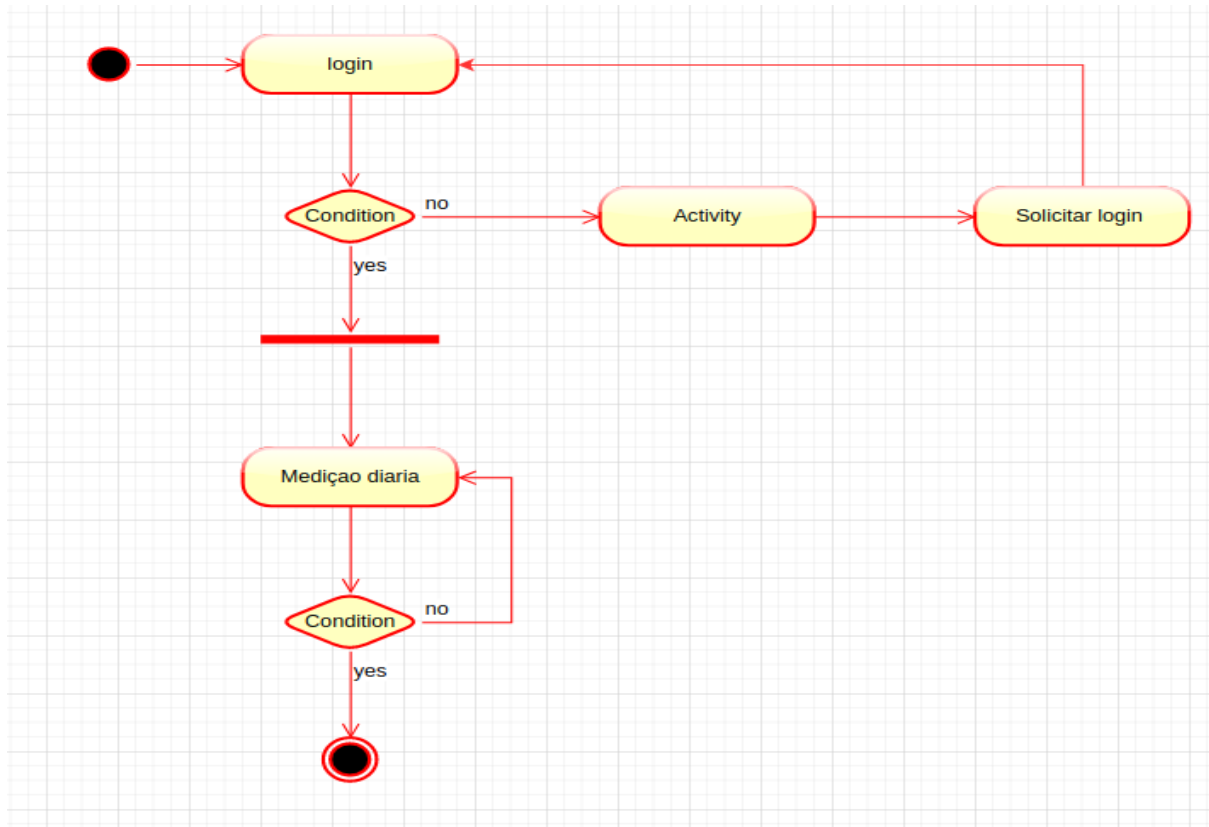


7.2.3 DIAGRAMA USUÁRIO-CLÍNICA:



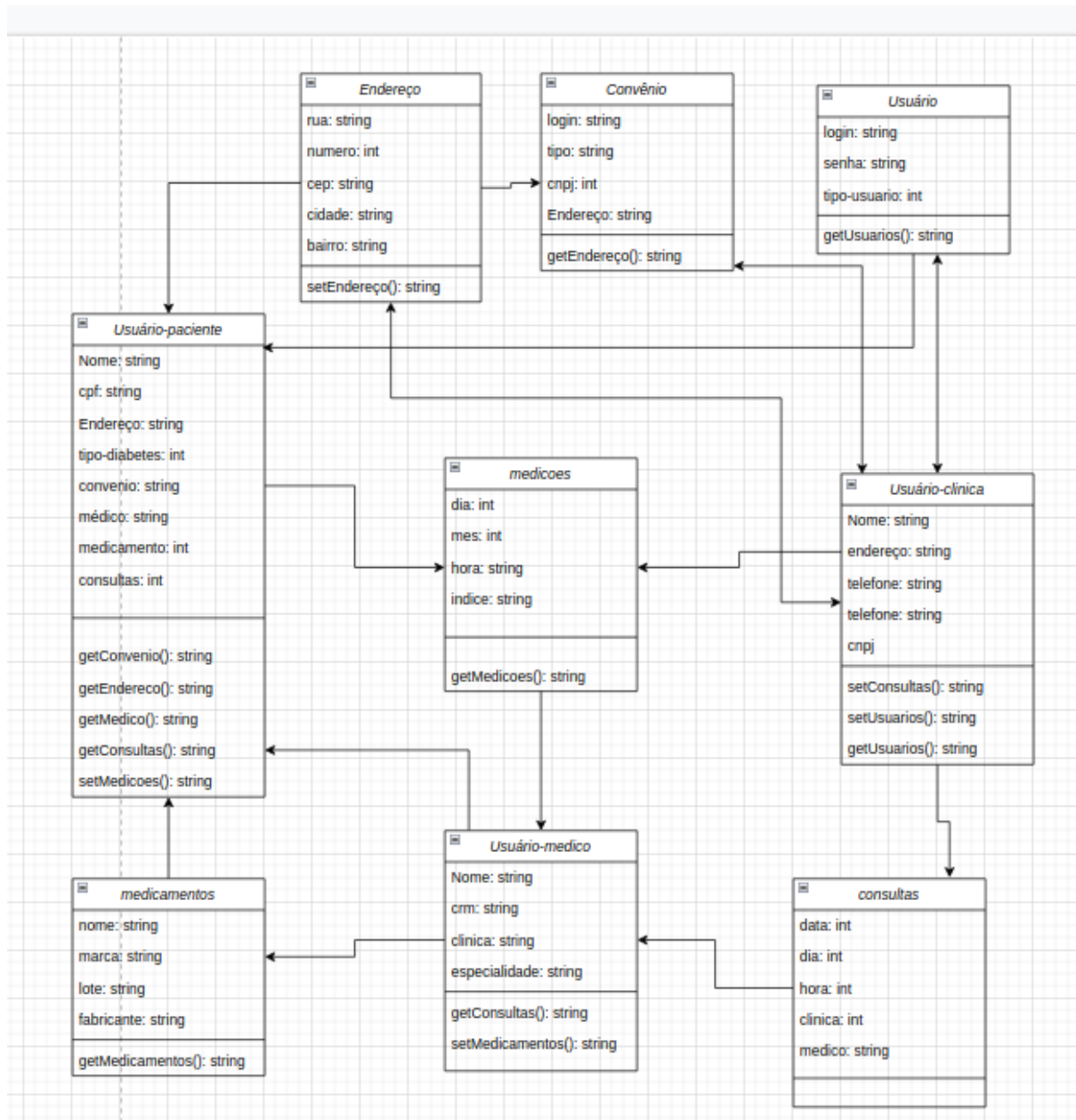
7.3. DIAGRAMA DE ATIVIDADE:

- usuário paciente.



Na etapa de prototipação mobile, foi focado na usabilidade, sendo elencado apenas o diagrama de atividade do usuário-paciente.

7.4 DIAGRAMA DE CLASSE:



8.PROTOTIPAÇÃO:

8.1. FRONT END LOGIN USUÁRIO CLÍNICA.

O projeto adotará uma abordagem em Angular e Typescript para a prototipação da aplicação desktop, buscando integrar os componentes curriculares do período de 2022.2. A demonstração será feita com o uso de Reactive, destacando a usabilidade e as funcionalidades da aplicação.

```

1 import React from 'react';
2 import logo from '../styles/img/logoGrande.png';
3 import '../styles/login.css';
4 import {Link} from "react-router-dom";
5
6
7 const LoginIndex = () =>{
8
9   return (
10     <header className="headerPosition">
11       <div className="LoginIndex1">
12         <div>
13           <h1 className="LoginHeader1">DIA A DIA</h1>
14           <h3 className="LoginHeader2"> Índice Glicêmico</h3>
15           <div className="loginBox">
16             <p className="loginText"> login</p>
17             <input placeholder="Email" type="text" className="loginInputText"/>
18             <br/>
19             <input placeholder="Senha" type="text" className="loginInputText2"/>
20             <br/>
21             <Link to={"/dashboard"}>
22               <input value="login" type="submit" className="loginButton"/>
23             </Link>
24             <p className="rescuepass"> Esqueceu a senha?</p>
25             <button className="fakeLink">Clique Aqui!</button>
26           </div>
27         </div>
28       </div>
29       <div className="loginIndex2">
30         <img className="logoGrande" src={logo} alt="logo D"/>
31       </div>
32     </div>
33     <div className="rectangle">
34
35   </div>

```

8.1. FRONT END DASHBOARD USUÁRIO CLÍNICA.

Uma breve demonstração do front-end do Dashboard do usuário na clínica está disponível no Github do projeto, onde todo o código está disponível para acesso.

```

1 import React from 'react';
2 import '../styles/dashboard.css';
3 import {Link} from "react-router-dom";
4 import logoDash from '../styles/img/LogoGrande.png';
5 import usuarioexemplo from '../styles/img/borat.jpeg';

3 usages  ▲ GustavoRocha
const DashBoard = () =>{
  const current : Date = new Date();
  const date : string = `${current.getDate()}/${current.getMonth()+1}/${current.getFullYear()} - ${current.getHours()}:${current.getMinutes()}:${current.getSeconds()}`;
  console.log(date);
  return(
    <div className="dashboard">
      <div className="sidebar">
        <h1>
          <Link to="/" className="Links">
            Dia a Dia
          </Link>
        </h1>

        <div className="menu">

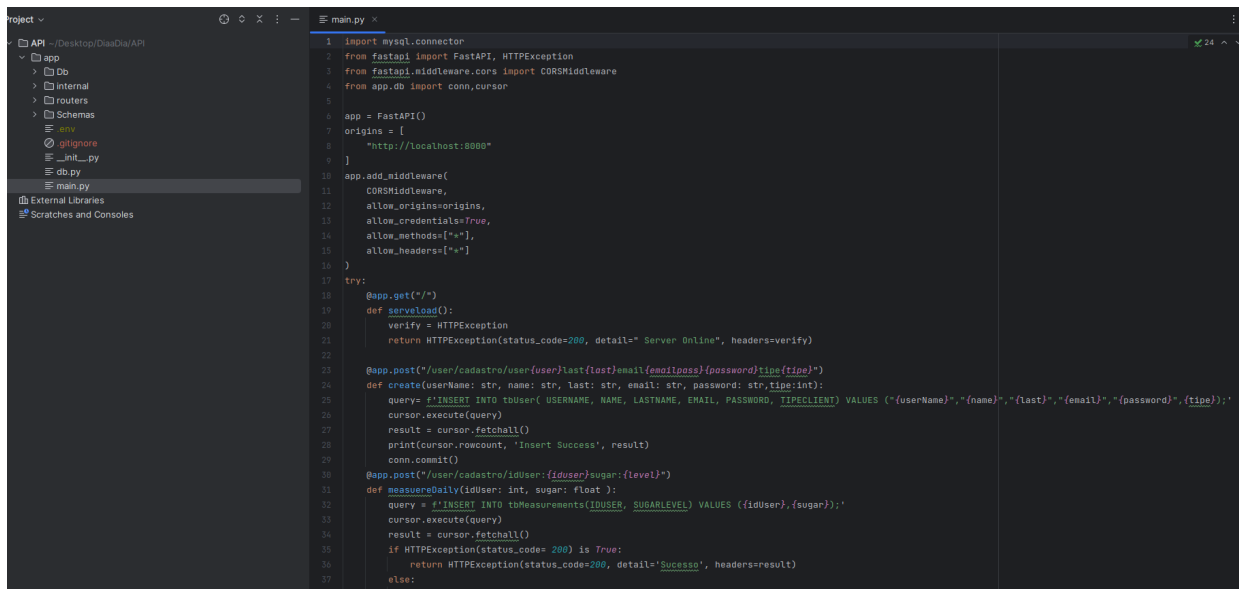
          <h3 className="menutitulo"> <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"
                                className="bi bi-columns-gap" viewBox="0 0 16 16">
              <path
                d="M6 1v3H1v1h5z" data-bbox="6 1 16 16"/>
            </h3>
          <h3 className="menutitulo"> MENU</h3>
          <p className="textosidebar">
            <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"
              className="bi bi-envelope-at" viewBox="0 0 16 16">
              <path
                d="M2 2a2 2 0 0 0 2 2v8.01A2 2 0 0 2 14 5.5a.5.5 0 0 0 0 1H2a1 1 0 0 1-.966-.741L5.64-3.471L8 9.583L7-4.2V8.5a.5.5 0 0 0 1 0V4a2 2 0 0 0 2 2" data-bbox="2 2 16 16"/>
              <path
                d="M14.247 14.247c1.01 0 1.587-.857 1.587-2.025v-.21c15.834 10.43 14.64 9 12.52 9h-.035c10.42 9 10.36 9 12.432v.214c9 14.82 10.438 1" data-bbox="14.247 14.247 16 16"/>
            </svg> Emails</p>
          </div>
        </div>
      </div>
    </div>
  );
}

```

8.2. BACK END

8.2.1 API

Uma breve demonstração de uma API em desenvolvimento usando FastAPI e MySQL está disponível. Através dessa API, é possível acessar e manipular dados armazenados em um banco de dados MySQL. A demonstração destaca o uso do FastAPI para a criação de rotas e endpoints, além da integração com o MySQL para o armazenamento e recuperação dos dados.



```

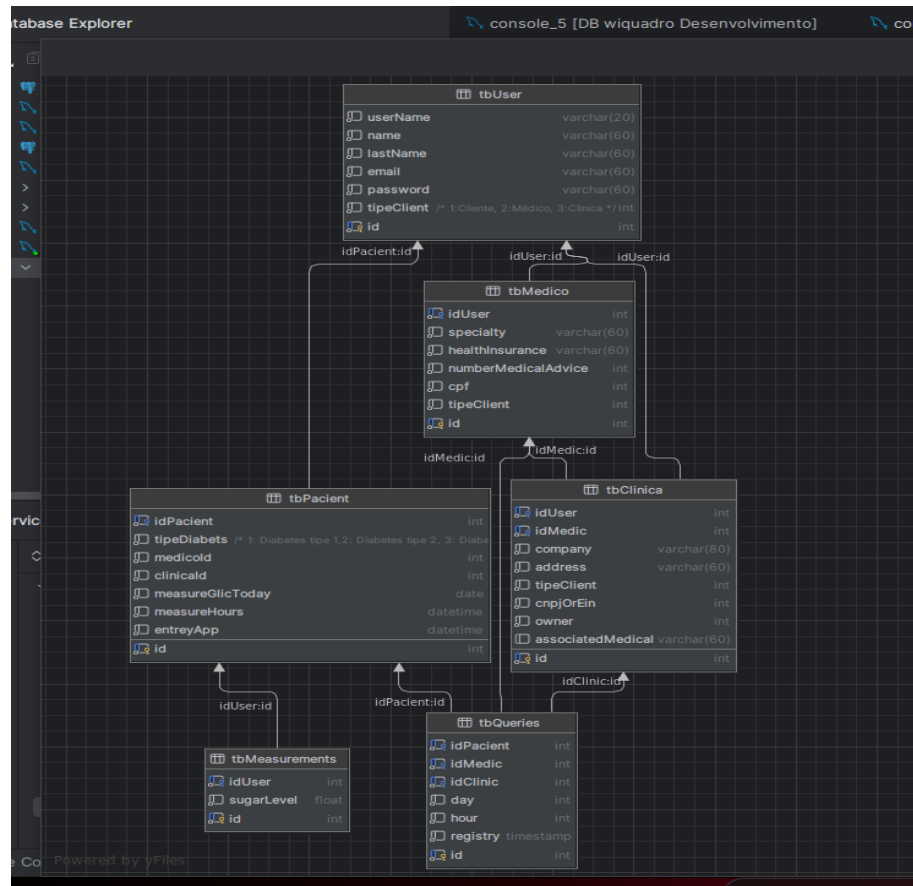
1 import mysql.connector
2 from fastapi import FastAPI, HTTPException
3 from fastapi.middleware.cors import CORSMiddleware
4 from app.db import conn, cursor
5
6 app = FastAPI()
7 origins = [
8     "http://localhost:8080"
9 ]
10
11 app.add_middleware(
12     CORSMiddleware,
13     allow_origins=origins,
14     allow_credentials=True,
15     allow_methods=["*"],
16     allow_headers=["*"]
17 )
18
19 try:
20     @app.get("/")
21     def serverload():
22         verify = HTTPException
23         return HTTPException(status_code=200, detail="Server Online", headers=verify)
24
25     @app.post("/user/cadastro/user={user}last={last}email={email}password={password}time={time}")
26     def create(userName: str, name: str, last: str, email: str, password: str, time: int):
27         query = f"INSERT INTO tbuser( USERNAME, NAME, LASTNAME, EMAIL, PASSWORD, TIPECLIENT) VALUES ('{userName}', '{name}', '{last}', '{email}', '{password}', {time});"
28         cursor.execute(query)
29         result = cursor.fetchall()
30         print(cursor.rowcount, 'Insert Success', result)
31         conn.commit()
32
33     @app.post("/user/cadastro/iduser={iduser}sugar={level}")
34     def measureDaily(idUser: int, sugar: float):
35         query = f"INSERT INTO tbmeasurements(IDUSER, SUGARLEVEL) VALUES ({idUser}, {sugar});"
36         cursor.execute(query)
37         result = cursor.fetchall()
38         if HTTPException(status_code=200) is True:
39             return HTTPException(status_code=200, detail="Sucesso", headers=result)
40         else:
41             return HTTPException(status_code=500, detail="Erro")

```

8.3 BANCO DE DADOS

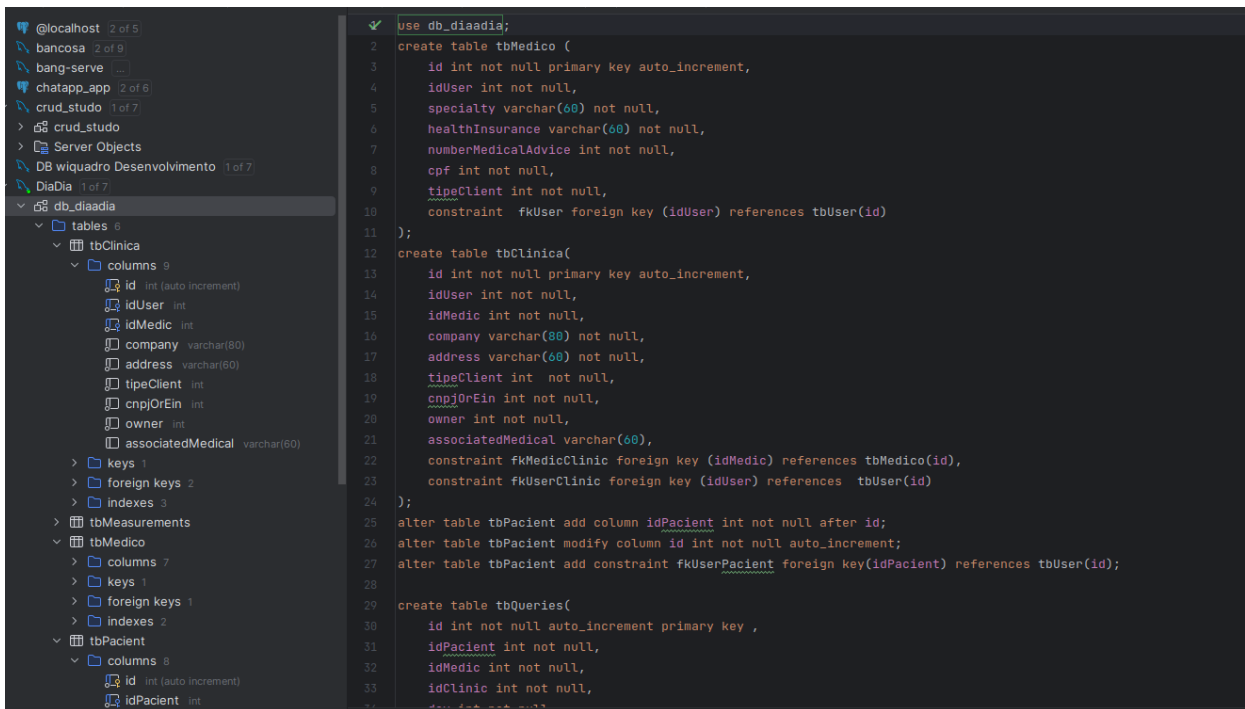
8.3.1 DIAGRAMA DE DESENVOLVIMENTO DE BANCO DE DADOS.

Um diagrama relacional em desenvolvimento, gerado pelo Datagrip, está sendo utilizado para visualizar a estrutura e relacionamentos de tabelas em um banco de dados MySQL.



8.3.2 SCRIPT DO BANCO DE DADOS COM MYSQL.

Trecho de script mysql:



The screenshot shows a MySQL database management tool interface. On the left, a tree view displays the database structure for 'db_diaadia'. It includes tables like 'tbClinica', 'tbMedico', 'tbPaciente', and 'tbQueries'. The 'tbClinica' table is expanded, showing columns: 'id' (int, auto increment), 'idUser' (int), 'idMedic' (int), 'company' (varchar(80)), 'address' (varchar(60)), 'tipeClient' (int), 'cnpjOrEin' (int), 'owner' (int), and 'associatedMedical' (varchar(80)).

On the right, a SQL script is displayed, starting with 'use db_diaadia;'. The script includes the following SQL statements:

```

2 create table tbMedico (
3   id int not null primary key auto_increment,
4   idUser int not null,
5   specialty varchar(60) not null,
6   healthInsurance varchar(60) not null,
7   numberMedicalAdvice int not null,
8   cpf int not null,
9   tipeClient int not null,
10  constraint fkUser foreign key (idUser) references tbUser(id)
11 );
12
13 create table tbClinica(
14   id int not null primary key auto_increment,
15   idUser int not null,
16   idMedic int not null,
17   company varchar(80) not null,
18   address varchar(60) not null,
19   tipeClient int not null,
20   cnpjOrEin int not null,
21   owner int not null,
22   associatedMedical varchar(60),
23   constraint fkMedicClinic foreign key (idMedic) references tbMedico(id),
24   constraint fkUserClinic foreign key (idUser) references tbUser(id)
25 );
26
27 alter table tbPaciente add column idPaciente int not null after id;
28 alter table tbPaciente modify column id int not null auto_increment;
29 alter table tbPaciente add constraint fkUserPaciente foreign key(idPaciente) references tbUser(id);
30
31 create table tbQueries(
32   id int not null auto_increment primary key ,
33   idPaciente int not null,
34   idMedic int not null,
35   idClinic int not null,
36   ...

```

9. RESULTADOS:

A respeito da integração com os componentes curriculares Testes de Software ministrada pelo docente Marcelo Valera e com Desenvolvimento Mobile ministrada pelo docente Bruno Emerson Queiroz. O projeto Dia-a-Dia na segunda estabelece relação com disciplinas acadêmicas como Testes e Desenvolvimento Mobile. No âmbito de Testes, a aplicação passará por testes de unidade e carga, a tecnologia primariamente idealizada será o unittest.mock em python para testes unitários o k6 para teste de carga e desempenho. utilizando a biblioteca unittest.mock , a fim de garantir a qualidade do código e a correta funcionalidade do sistema. Serão aplicadas técnicas de testes automatizados e manuais para identificar e corrigir possíveis erros e bugs. Já em relação à disciplina de Desenvolvimento Mobile, o projeto utilizará frameworks como Ionic para a criação de uma interface amigável e

responsiva, considerando as melhores práticas de design para dispositivos móveis. A integração dessas disciplinas ao projeto proporcionará uma abordagem prática e aplicada, fortalecendo o conhecimento teórico adquirido durante os estudos e permitindo uma melhor compreensão das técnicas e metodologias envolvidas no desenvolvimento de software.

Durante o desenvolvimento do projeto, algumas limitações relacionadas ao tempo e dificuldades técnicas. O tempo disponível para o desenvolvimento foi um desafio. Assim como dificuldades relacionadas à integração das tecnologias no projeto. Exemplo como uso do pandas e do matplotlib e sua integração com os demais recursos.

No entanto, apesar dessas limitações, permanecemos otimistas e focados em alcançar resultados positivos. Superamos os obstáculos com determinação e buscamos soluções alternativas quando necessário. Mesmo diante das dificuldades, temos certeza que conseguiremos desenvolver uma aplicação funcional e intuitiva, que proporcionará aos pacientes diabéticos uma ferramenta útil para o monitoramento e gerenciamento de sua saúde.

Em resumo, apesar das limitações encontradas, estamos otimistas com os resultados alcançados até o momento e confiantes de que o projeto Dia-a-Dia trará benefícios significativos para os pacientes diabéticos, auxiliando-os no monitoramento e controle da doença.

CONCLUSÃO:

No relatório abordamos diversos aspectos relevantes para o projeto Dia-a-Dia. Iniciamos com a apresentação do contexto, justificativa, objetivos e público-alvo do projeto. Em seguida, descrevemos a metodologia utilizada no desenvolvimento, incluindo as etapas, técnicas e ferramentas adotadas, como o uso do Ionic e Flutter para a criação da aplicação multiplataforma, a integração com o MySQL para o gerenciamento de dados, e a utilização do Matplotlib para a visualização dos dados glicêmicos. Também apresentamos os resultados obtidos, destacando os aspectos positivos e eventuais limitações. Por fim, concluímos ressaltando a importância e os benefícios do projeto, que visa melhorar o monitoramento e gerenciamento da diabetes, promovendo uma melhor qualidade de vida para os pacientes.

REFERÊNCIA:

Análise de Requisitos. Disponível em:

<<https://analisederequisitos.com.br/requisitos-funcionais-e-nao-funcionais>>. Acesso em: 10 jan. 2023

BARRETO, Elis. Diabetes aumentou 16% na população mundial nos últimos dois anos. CNN Brasil, [notícia online]. Disponível em:

<<https://www.cnnbrasil.com.br/saude/diabetes-aumentou-16-na-populacao-mundial-nos-ultimos-dois-anos/#:~:text=Dados%20da%20Federa%C3%A7%C3%A3o%20Internacional%20de,cerc a%20de%207%25%20da%20popula%C3%A7%C3%A3o.>>>. Acesso em: 05 jul. 2023.

BUTTERFIELD, Eric. Flutter in Action. Shelter Island, NY: Manning Publications, 2020.

DART. Dart Documentation. Disponível em: <<https://dart.dev/guides>>. Acesso em: 03 jul 2023.

FASTAPI. FastAPI Documentation. Disponível em: <<https://fastapi.tiangolo.com/>>. Acesso em: 05 jul 2023.

<<https://ionicframework.com/docs>>. Acesso em: 05 jul 2023.

FLUTTER. Flutter Documentation. Disponível em: <<https://flutter.dev/docs>>. Acesso em: 06 jul 2023..

K6.IO. k6 Documentation. Disponível em: <<https://k6.io/docs>>. Acesso em: 10 jul 2023.

MUNROE, Josh. Building Mobile Apps with Ionic: Cross-Platform Apps Built with Web Technologies. Sebastopol, CA: O'Reilly Media, 2017.

Pressman, R. S. Engenharia de Software: uma abordagem profissional. 8. ed. Porto Alegre: AMGH, 2016.

ORACLE CORPORATION. MySQL Documentation. Disponível em:

<<https://dev.mysql.com/doc/>>. Acesso em: [data de acesso].

PYTHON SOFTWARE FOUNDATION. unittest.mock - mock object library. Disponível em:

<<https://docs.python.org/3/library/unittest.mock.html>>. Acesso em: [data de acesso].

SAVAGE, Sebastián. FastAPI: Building High-Performance Python Web Applications.

Birmingham, Reino Unido: Packt Publishing, 2020.

De acordo com as normas da ABNT (Associação Brasileira de Normas Técnicas), a referência bibliográfica para um livro sobre MySql seria estruturada da seguinte forma:

SILVA, Ricardo. Banco de Dados com MySql. São Paulo: Novatec Editora, 2019.

SMITH, Chris. Dart in Action. Shelter Island, NY: Manning Publications, 2013.
SOMMERVILLE, Ian. **Engenharia de SOFTWARE**. 9. ed. São Paulo: Pearson, 2011.