UNIVERSIDADE PAULISTA - UNIP EAD

Projeto Integrado Multidisciplinar

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Bruna da Silva Barretto (RA 0617559)

Tiago Costa Santos (RA 0623314)

Ivan Silva Dias Moura (RA 0622330)

Luis Cardoso Queiroz (RA 0620320)

Fabricio Alves da Silva (RA 0616633)

Rafael Lopes Martins (RA 0621808)

Sistema em C para cadastro de pacientes diagnosticados com covid-19

ARARAQUARA-SP

2022

UNIVERSIDADE PAULISTA - UNIP EAD

Projeto Integrado Multidisciplinar

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Sistema em C para cadastro de pacientes diagnosticados com covid-19

Projeto multidisciplinar para o

desenvolvimento e aplicação dos conteúdos das disciplinas cursadas no primeiro bimestre vigente no 1° Semestre do curso de Análise e desenvolvimento de sistema na instituição de ensino Universidade Paulista-UNIP EAD.

ARARAQUARA-SP

2022

SUMÁRIO

1 INTRODUÇÃO....................................................................................................06

2 ENGENHARIA DE SOFTWARE .........................................................................07

2.1 Os princípios da engenharia de software.....................................................07

2.1.1 Metodologia.................................................................................................07

2.1.2 Metodologia cascata (Waterfall)..................................................................08

2.1.3 Requisitos....................................................................................................10

2.1.4 Requisitos funcionais ..................................................................................12

2.1.5 Requisitos não funcionais............................................................................12

3 DESENVOLVIMENTO...........................................................................................16

3 .1 Linguagem C..................................................................................................16

3.2 Modo de funcionamento do software..............................................................18

3.3 Arquivos...........................................................................................................22

3.4 Execução do sistema.......................................................................................24

3.5 Fluxograma......................................................................................................25

4 CONCLUSÃO.........................................................................................................26

REFERÊNCIAS..........................................................................................................27

**RESUMO**

O projeto multidisciplinar proposto pela UNIP tem como finalidade aplicar os conhecimentos adquiridos nas aulas ao longo do bimestre para o desenvolvimento de um sistema em linguagem C utilizando o Code::Blocks (Ambiente de desenvolvimento integrado de código aberto e multiplataforma) que será utilizado em hospitais para cadastrar pacientes que forem diagnosticados com Covid-19 e que precisem de acompanhamento médico e monitoramento. Ao receber o diagnóstico positivo, os profissionais da saúde devem realizar o login no sistema informando o usuário e uma senha válida. Em seguida efetuar o cadastro do paciente com os dados pessoais do próprio, o sistema deverá salvar em um arquivo de texto o CEP (código de endereçamento postal) e idade dos pacientes que fazem parte do grupo de risco (maiores de 65 anos) e que possuem alguma comorbidade para que a informação possa ser consultada sempre que necessário e enviada para a central da Secretaria da Saúde da cidade.

Palavras-chaves: Sistema, Linguagem C, Cadastro, Pacientes, Covid-19, Monitoramento

. **ABSTRACT**

The multidisciplinary project proposed by UNIP aims to apply the knowledge acquired in classes over the two months for the development of a C-language system using Code::Blocks (Integrated open source and cross-platform development environment) that will be used in hospitals to register patients who are diagnosed with Covid-19 and who need medical follow-up and monitoring.Upon receiving the positive diagnosis, health professionals must log into the system informing the user and a valid password.Then register the patient with the patient's own personal data, the system must save in a text file the ZIP Code (postal address code) and age of patients who are part of the risk group (over 65 years) and who have some comorbidity so that the information can be consulted whenever necessary and sent to the central Health Department of the city.

Keywords: System, Language C, Registration, Patients, Covid-19, Monitoring.

**1 INTRODUÇÃO**

Os primeiros casos diagnosticados pela COVID-19 apareceram em dezembro de 2019, na cidade de Wuhan, China. Devido sua rápida propagação a nível mundial surgiu um destaque maior aos idosos, especialmente aqueles com comorbidades em razão do grande potencial de risco dessa parcela da população, orientações de distanciamento social foram recomendadas para esse grupo.

Com ênfase no momento pandêmico surge a necessidade do desenvolvimento de um sistema operacional que tem como funcionamento cadastrar pacientes diagnosticados com a covid-19 que demanda monitoramento e acompanhamento médico. Os profissionais da saúde que realizaram o cadastro do paciente informam todos os dados pessoais que o sistema solicitará, como nome, CPF, telefone, endereço, data de nascimento e e-mail, se o mesmo possua comorbidades e for acima de 65 anos será inserido ao grupo de risco e as informações serão enviadas para a central da Secretaria da Saúde da cidade.

Ao longo do desenvolvimento do projeto aplicou uso dos conceitos apresentados nas disciplinas estudadas: Linguagem e Técnicas de Programação para aplicação da regras da Linguagem, estrutura de dados , entrada e saída de informações, condicionais, laços de repetição entre outros conteúdos e na Engenharia de Software I os fundamentos da engenharia de software, o processo de software, práticas de comunicação, planejamento, modelagem, construção e implantação para a elaboração do sistema operacional fez-se a utilização da Linguagem C na IDE (ambiente de desenvolvimento integrado) Code::Blocks que foi concluído pelo grupo.

**2.1 OS PRINCÍPIOS DA ENGENHARIA DE SOFTWARE**

Os princípios da engenharia de software formam a base de métodos, técnicas, métodos e ferramentas que são aplicados na prática e orientam a prática de desenvolvimento de soluções de software. Os princípios se aplicam a produtos e processos de software que se tornam práticas de desenvolvimento de software por meio da adoção de métodos e técnicas. Métodos e técnicas geralmente descrevem a metodologia suportada pelo uso de ferramentas.

**2.1.1 METODOLOGIA**

Um método de desenvolvimento é um conjunto de práticas projetadas para coordenar o trabalho de criação de software. Como indicado anteriormente, o projeto correspondente envolve muitas etapas e muitas pessoas. Então não pode ser conduzido de modo aleatório tem que ser coordenado quem, quando, como e onde. E não basta apenas definir tudo isso em uma reunião, manter a palavra até que o projeto seja concluído.

À medida que a fase avança, os requisitos mudam. Tudo é muito dinâmico. Nesse contexto, os métodos de desenvolvimento de software entram em cena para sequenciar como o trabalho é feito e acompanhado. Claro, tudo deve ser planejado com antecedência. Ou seja, a metodologia utilizada, quem é responsável por ela, prazos, etc.

Para este projeto, aplicamos o método Waterfall, uma cascata de etapas utilizada desde a Segunda Guerra Mundial e popularizada na década de 70 que foi desenvolvida para criação de software.

Há algum tempo, Winston W. Royce publicou um artigo explicando como funciona o método cascata. Ele ilustra e questiona sua funcionalidade, argumentando que não era uma das melhores maneiras de usá-los.

Embora seu artigo tenha criticado o modelo, Royce foi novamente responsável por nomear e publicar o modelo em cascata.

**2.1.2 METODOLOGIA CASCATA**

O método cascata, como o nome sugere, é uma metodologia de desenvolvimento fluida em que cada passo leva ao seguinte, como uma cascata. Em seu exemplo, W.W. Royce chama essa metodologia como estática fluida. Isso ocorre porque não há alterações no processo. Portanto, o modelo em cascata é menos sujeito a erros e alterações, pois há apenas uma direção e não há como reiniciar as etapas.

Por isso, a metodologia cascata é um modelo em que um plano de etapas é essencial para garantir que cada fase seja executada corretamente, destravando as próximas etapas no desenvolvimento de software para aumentar a qualidade. O projeto só pode prosseguir quando as fases anteriores forem concluídas e aprovadas pelos stakeholders.

Outro aspecto importante do modelo em cascata é que é o cliente quem específica suas necessidades e prioridades de software, portanto, você deve trabalhar com ele para dar esse passo. Ou seja, a equipe de desenvolvimento não atua de forma consultiva, mas simplesmente identifica melhorias no projeto e as implementa conforme sugestão do cliente.

Esse detalhe é uma das maiores desvantagens desta metodologia. Torna-se difícil saber antecipadamente o que o cliente quer e precisa. Essas informações são coletadas com mais eficiência à medida que o projeto avança.

Além do desenvolvimento fluido do método cascata, a fase de testes também se diferencia de outros métodos mais modernos e inovadores. Esta etapa é realizada ao final do desenvolvimento onde são verificadas a funcionalidade e a qualidade do software. Erros durante a fase de teste podem levar ao reinício do projeto. Isso porque no método cascata não há como passar por apenas uma fase.

### O Modelo Cascata funciona da seguinte maneira:

### • Concepção

### • Iniciação

### • Análise

### • Design

### • Codificação

### • Teste

### • Implementação

### • Manutenção

### Nosso grupo é formado por 6 integrantes, e logo após o levantamento da análise de requisitos, o projeto teve que ser subdividido em duas partes e executado para que todos os envolvidos trabalhassem no desenvolvimento do programa e aplicassem as disciplinas estudadas no projeto multidisciplinar de acordo com o proposto.

* PARTE I

Foi realizada a implementação do registro do usuário, senha, login, validação correspondente e do menu. Em seguida, a elaboração da inscrição de pacientes e a devida validação.

* PARTE II

Os participantes do grupo ficaram responsáveis pela implementação de arquivos .txt da parte do paciente em questão e o design do programa.

### Todos os 6 alunos estiveram envolvidos no desenvolvimento do programa, permitindo contribuições e aprendizado prático.

### **2.1.3 Requisitos**

Requisitos são as condições, recursos, objetivos e utilidades que o sistema precisa satisfazer de as demandas da empresa ou usuário. Ao corresponder tais expectativas, segurando que o sistema se torne funcional e confiável, bem como também apresente um excelente desempenho.

Uma especificação de software é uma descrição dos recursos e funcionalidades de um sistema. A especificação sinaliza as expectativas dos usuários do produto de software. Os pré-requisitos podem ser óbvios ou ocultos, conhecidos ou desconhecidos, esperados ou inesperados do ponto de vista do cliente.

### Levantamento e documentação de requisitos:

A especificação de requisitos de software normalmente acontece com a realização de um documento pelo analista de sistemas. O profissional verifica junto ao cliente e demais interessados quais são as funções necessárias no software. Esta etapa é chamada de levantamento de requisitos pode ser feita através de:

* Reuniões;
* Entrevistas;
* Workshops;
* Relatórios;
* Prototipagem;
* Questionários;
* Brainstorming.

Essa tarefa é realizada usando um método conhecido como VORD, sigla em inglês para de Definição de Requisitos Orientada a Ponto de Vista (Viewpoint-Oriented Requirements Definition).

Com essas informações, os analistas podem documentar esses requisitos em linguagem técnica em documentação de requisitos. Então é enviado esse documento para que a equipe de desenvolvimento do software dê continuidade no processo.

Durante o processo de construção, programadores, arquitetos e analistas podem sugerir sugestões, soluções e melhorias.

Seguindo o protocolo é de fácil entendimento as expectativas e objetivos de acordo com o cliente para o desenvolvimento do sistema e além disso otimizar o tempo da empresa e da equipe.

Isso em função que a especificação dos requisitos de software deixa ambas as partes com mais especificação e detalhes sobre o software e torna o processo de desenvolvimento mais assertivo.

## Classificação dos requisitos de software:

Para documentar adequadamente uma especificação de requisitos de software, deve-se prestar atenção à sua taxonomia. Isso é importante para não perder o foco durante o processo.

A prioridade de cada uma dessas categorias varia de acordo com a empresa, plataforma e tipo de dispositivo no qual o software é usado. Por exemplo, aplicativos para smartphone e sistemas web têm prioridades diferentes. Os requisitos podem ser divididos em requisitos funcionais e não funcionais.

### **2.1.4 Requisitos funcionais**

Agora que sabemos quais são os requisitos, vamos dividi-los explicando cada um deles, começando pelos requisitos funcionais. A engenharia da empresa de software pode enfatizar os requisitos funcionais. Trata-se da realização de uma necessidade ou requisito por software. No entanto, vários requisitos funcionais podem ser atendidos dentro da mesma função. São várias as funcionalidades e serviços que o sistema pode disponibilizar aos seus clientes. Abaixo estão algumas das muitas funções que o software pode executar:

* Incluir/Excluir/Alterar nome em uma tela de manutenção de funcionário
* Geração de relatório de determinado período de vendas
* Efetuar pagamentos de compra através de crédito ou débito
* Consulta e alterações de dados pessoais de clientes
* Emissão de relatórios de clientes ou vendas
* Consulta de saldo ou estoque

Os requisitos funcionais são de extrema importância no desenvolvimentode aplicativos, pois, sem eles não há funcionalidades nos sistemas. Seus modelos devem ser construídos em um nível de entendimento claro e objetivo, além de um código fonte totalmente aplicável. Conclusão, para se obter requisitos funcionais de qualidade a fábrica de software deve estar atenta a síntese e a semântica dos mesmos.

**2.1.5 Requisitos não funcionais**

Os requisitos funcionais definem o que o sistema faz, então a engenharia de software diz que os requisitos não funcionais definem como o sistema faz, mas essa definição não é clara. Os requisitos não funcionais não estão diretamente relacionados à funcionalidade do sistema.

Também conhecido como atributo de qualidade, ainda é muito importante no desenvolvimento de sistemas. Geralmente tratados como pressupostos do projeto e restrições técnicas, os requisitos não funcionais são praticamente qualquer necessidade que não pode ser satisfeita pela funcionalidade.

Requisitos não funcionais geralmente mensuráveis definem as características e limites do sistema. B. Se você pode determinar o método de desenvolvimento, tempo, espaço, sistema operacional, etc. e suas medições, é importante vincular essa medição ou referência a cada requisito não funcional, requisitos funcionais. Para maior clareza, aqui está um exemplo de algumas propriedades e suas métricas:

* O tamanho pode ser medido em kbytes e número de Chip de RAM.
* A velocidade está ligada ao tempo de utilização da tela, ou transações processadas por segundos.
* A métrica da portabilidade é o número de sistema-alvo.
* A facilidade de uso pode ser medida pelo número de janelas ou o tempo de treino
* A confiabilidade tem ligação com o tempo médio que o sistema pode vir a falhar, a disponibilidade ou até mesmo a taxa de ocorrência de falhas.

Esses são apenas alguns dos exemplos onde as propriedades e métricas são associadas a cada requisito não funcional. Além do mais, esses requisitos não funcionais são divididos em três tipos principais. Requisitos de Produto Final, Requisitos Organizacionais, Requisitos Externos que se dividem em diversos outros tipos. Listamos abaixo alguns exemplos básicos de requisitos não funcionais:

* Utilização do módulo de Informações Cadastrais em modo off-line
* O sistema deve ser implementado na linguagem Java
* O sistema deverá se comunicar com o banco SQL Server
* Um relatório de supervisão deverá ser fornecido toda sexta-feira
* O sistema deve ser executável em qualquer plataforma

Muitos programas de software falham e até falham. Isso ocorre porque não pré-definimos esses atributos de qualidade ao priorizá-los ao definir nossa arquitetura. Se esses requisitos não funcionais causam problemas tão críticos em sistemas de software que ignorá-los leva a uma falta de sucesso ou falha no desenvolvimento futuro de software, por que eles são tão negligenciados?

1. Apenas as grandes empresas, onde existem uma equipe de TI com maior disposição, onde acompanham de perto a elaboração dos softwares é que dão credibilidade aos Requisitos não Funcionais.
2. Os Requisitos não funcionais são de difícil estimativa tanto para prazo, trabalho ou custo.
3. É grande o número de fornecedores que julgam “gastar mais” no desenvolvimento do aplicativo, por exemplo, ao lidarem com os Requisitos não Funcionais e só não os ignoram quando são solicitados explicitamente pelo cliente.

Todas as atividades relacionadas aos requisitos são, até certo ponto, parte do processo de engenharia de requisitos. A engenharia de requisitos, uma das principais áreas do desenvolvimento de software, possui várias definições de acordo com a literatura técnica de engenharia:

* Termo que descreve as atividades relacionadas à investigação e definição de escopo de um sistema de software.
* Método sistemático de desenvolvimento de requisitos através de um processo cooperativo de análise onde os resultados das observações são codificados em uma variedade de formatos onde a precisão das observações são verificadas frequentemente.
* Curso de descoberta, análise, documentação e verificação das funções e restrições do sistema.

Características e exemplos de requisitos não funcionais:

Os requisitos não funcionais estão relacionados ao uso do aplicativo em termos de desempenho, usabilidade, confiabilidade, disponibilidade, segurança e tecnologias relacionadas. Os requisitos não funcionais geralmente levam a restrições nos requisitos funcionais. Tratados como restrições, os requisitos não funcionais geralmente determinam todas as necessidades que não podem ser atendidas pelos requisitos funcionais. Estes normalmente definem as características e restrições do aplicativo, como método de desenvolvimento, tempo, espaço, etc.

Os requisitos não funcionais são divididos em três tipos principais: requisitos do produto final, requisitos organizacionais e requisitos externos. Confira abaixo alguns requisitos básicos não funcionais.

* Módulo de informações cadastrais offline
* Linguagem específica: Java, Objective-C, Swift, #C, [Visual Basic](https://docs.microsoft.com/pt-br/visualstudio/get-started/visual-basic/tutorial-console?view=vs-2019) etc
* Executável em qualquer plataforma
* Emissão de relatório por usuário

Com uma compreensão profunda dos tipos de requisitos do seu aplicativo, você pode determinar o futuro do seu aplicativo. Também vale a pena considerar a flexibilidade no desenvolvimento de aplicativos, combinando projetos de alta qualidade sem altos custos iniciais. Em suma, este tipo de estratégia não só permite uma visão clara dos serviços implementados, como também facilita e agiliza a manutenção e resolução de problemas. Não apenas do lado do cliente, mas também do lado da empresa de desenvolvimento.

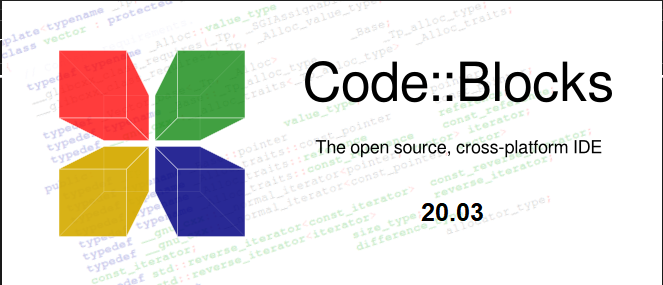
**3 Desenvolvimento**

IDE CODEBLOCKS será usado em todo o processo de codificação e execução do sistema, usando a linguagem de codificação C.

Code::Blocks (ou C:: B) é um ambiente de desenvolvimento integrado de código aberto e multiplataforma. Ele foi desenvolvido em C ++, usando wxwidgets. Sua arquitetura é orientada a plugin, portanto, sua funcionalidade é definida pelos plugins que fornece a ele.

Code::Blocks é voltado para o desenvolvimento em C/C++ e Fortran, podendo também ser usado para a criação de ARM, AVR, D(linguagem de programação), DirectX, FLTK, GLFW, GLUT, GTH+, Irrlicht, Lightfeather, MATLAB, OGRE, OpenGL, Qt,SDL, SFML, STL, SmartWin e programas ou aplicativos com wx, embora, em certos casos, a instalação de SDKs ou frameworks seja necessária.

IMAGEM 01 - IDE CODE::BLOCKS

 Fonte: Autoria própria

**3.1 Linguagem C**

C é uma linguagem estruturada, imperativa, procedural e compilada padronizada pela International Organization for Standardization (ISO) e criada em 1972 por Dennis Ritchie do AT&T Bell Labs para o desenvolvimento do sistema operacional Unix.

C é uma das linguagens de programação mais populares e quase nenhuma arquitetura está sem um compilador C. C influenciou muitas outras linguagens de programação, como a linguagem Java. Em particular, afeta C ++, que originalmente começou como uma extensão de C.

A linguagem C encontra-se na versão/padrão internacional C17 (ISO/IEC 9899:2018) lançada em junho de 2018, substituindo a versão C11 (ISO/IEC 9899:2011), disponível em ISO e IEC e com suporte para GCC8 e Clang LLVM6.

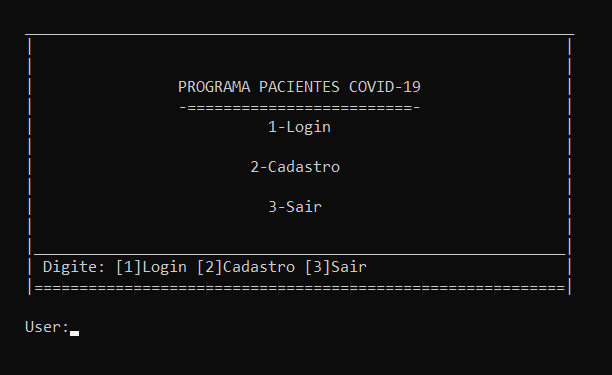
As principais características da linguagem C que as definem são: portabilidade, modularidade, recursos de baixo nível e simplicidade. Além disso, é uma linguagem imperativa de uso geral. Entenda o que significam:

* Portabilidade: pode ser compilada em diferentes arquiteturas, seja de hardware ou de software. É possível usar C no Mac ou PC, com Linux ou Windows. Isso pode parecer algo simples, mas na verdade nem todas as linguagens funcionam assim. Java, C++ e Python, por exemplo, também são linguagens altamente portáteis. C#, nem tanto. Como foi desenvolvida pela Microsoft, programadores têm de fazer vários rodeios para conseguir compilar um programa escrito em C# no Linux, por exemplo.
* Modularidade: um programa em C é dividido em vários blocos de programação distintos, ou seja, funções que não estão interligadas. Assim, no momento em que a sua função é fechada, aquilo que você escreveu não vai afetar os blocos seguintes - a não ser que seja chamada, é claro.
* Recursos de baixo nível: C é uma ótima linguagem para sistemas operacionais porque, além de eficiente, tem muitos recursos para controlar a memória da sua máquina. Além disso, ela permite acesso ao microprocessador, permitindo a chamada de rotinas em Assembly.
* Simplicidade: a sintaxe de C é relativamente fácil de aprender e, desde que você siga as regras, dificilmente vai cometer erros que possam comprometer o seu programa. Além disso, o compilador lhe avisará da maior parte dos erros de sintaxe que você cometer e dará dicas para você corrigi-la.
* Linguagem Imperativa: em C você usa funções que modificam o estado do programa. Elas são divididas em blocos e você as chama diretamente.
* Uso geral: embora não seja, obviamente, a melhor escolha para todas as aplicações, você pode fazer quase qualquer coisa com C.

**3.2 Modo de funcionamento do software**

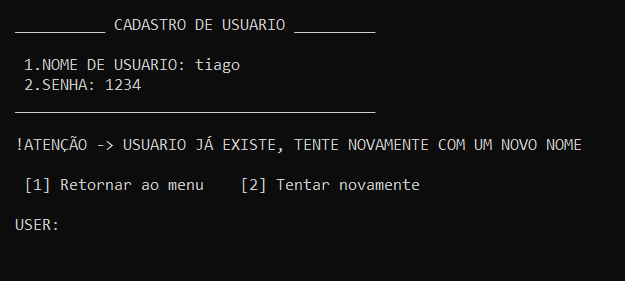
O sistema conta com diversas funcionalidades, sessões, telas e opções onde o usuário deverá digitar o número da opção que deseja acessar, uma vez digitada a opção correta o usuário deverá pressionar enter, para acessar, caso o mesmo digite e tente acessar uma opção que não existe será apresentado um erro

IMAGEM 02 – Tela inicial



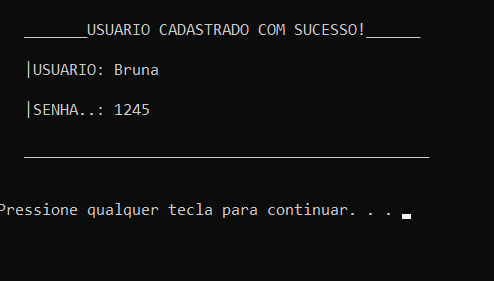
Fonte: Autoria própria.

* Antes de poder utilizar para registrar pacientes o usuário deverá criar um acesso, este acesso pode ser criado acessando na primeira tela a opção cadastro de funcionários, onde o mesmo deverá inserir um novo nome de usuário e senha para o acesso às demais opções.

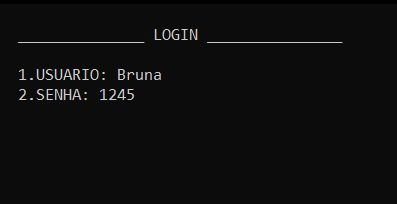
IMAGEM 03 – Login e senha já existentes.

Fonte: Autoria Própria

* Após criar este acesso o usuário está apto a acessar as demais áreas do sistema, caso queira poderá alterar seus dados de acesso. É possível continuar a inscrever o paciente usando o sistema.

IMAGEM 04 - Cadastro de usuário e senha

* Fonte: Autoria própria
* Optando pela opção 1 da tela inicial o funcionário deve inserir um nome de usuário e uma senha.

IMAGEM 05 - Tela de login

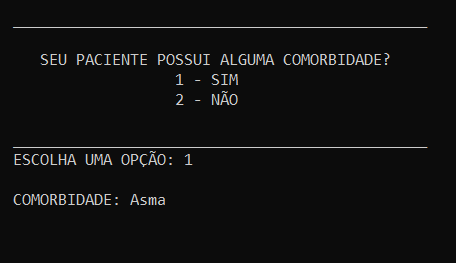
Fonte: Autoria Própria

* Se o login estiver correto, abre uma tela com a opção para cadastrar os dados do paciente. Em hipótese de erro, uma mensagem de erro aparecerá na tela indicando tentar novamente pois as credenciais estão incorretas.

IMAGEM 06 - Tela para Cadastro de Paciente

Fonte: Autoria própria

* A próxima tela pergunta se o paciente que está sendo cadastrado tem alguma comorbidade. Se houver comorbidade, o funcionário deverá indicar a comorbidade.

IMAGEM 07 - Tela de cadastro paciente com comorbidade

Fonte: Autoria própria

* O registro está completo após todos os dados do paciente terem sido inseridos corretamente.

IMAGEM 08- Cadastro do paciente concluído.

Fonte: Autoria própria.

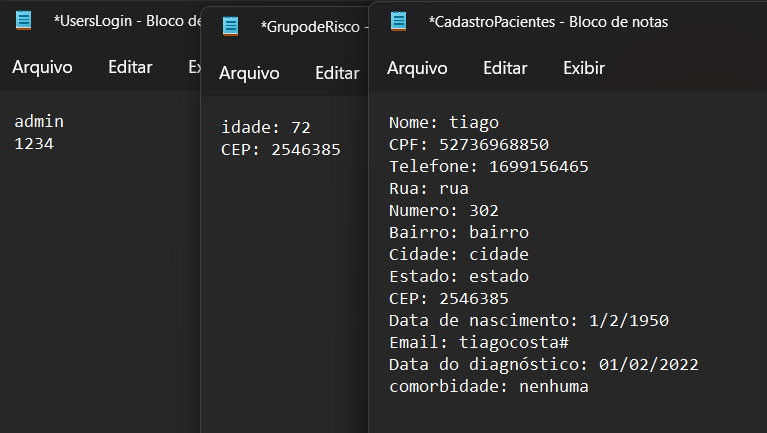
* As opções aparecem na tela para que possa ser decidido a próxima etapa do programa. Pressione [1] para retornar ao registro do novo paciente. [2] é usado para finalizar.

**3.3 Arquivos**

Todos os dados inseridos no sistema são armazenados em arquivos e o funciona como uma espécie de banco de dados. Todos os dados do paciente podem ser visualizados lá.

Em seguida da primeira execução do sistema, arquivos do tipo 'txt' que são criados o armazenamento contendo o .txt para ser compilado e executado.

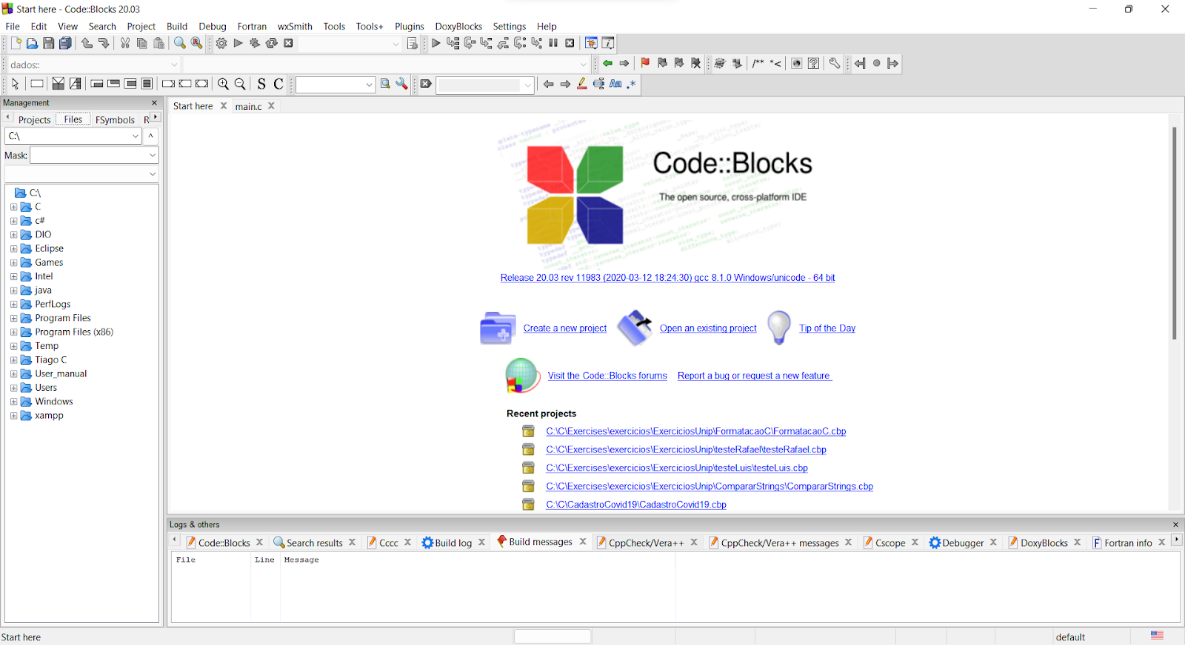
* O arquivo CadastroCovid19.txt armazena informações de todos os pacientes registrados no sistema.
* Os dados dos pacientes cadastrados no sistema são salvos no arquivoCadastroPacientes.txt
* O arquivo UsersLogin.txt armazena o usuário e a senha para entrar no sistema. A primeira linha corresponde ao login e a segunda linha equivale a senha, para fins de teste, optamos pela palavra "admin" como usuário e “1234” para senha. No entanto, os profissionais da saúde podem cadastrar qualquer usuário e senha por meio de letras, números ou símbolos.
* O arquivo GrupodeRisco.txt armazena informações de pacientes que fazem parte de um grupo de risco.

IMAGEM 09- Arquivos .txtFontes:Autoria própria.

**3.4 Execução do sistema**

Para executar o sistema, o usuário deve ter algum tipo de IDE (ambiente de desenvolvimento integrado) instalado no computador onde o sistema é utilizado. Após a instalação do programa, o usuário deve abrir o arquivo em sua IDE compilá-lo (GNU GCC Compiler) e executá-lo.

IMAGEM 08-Tela inicial do Code::Blocks.

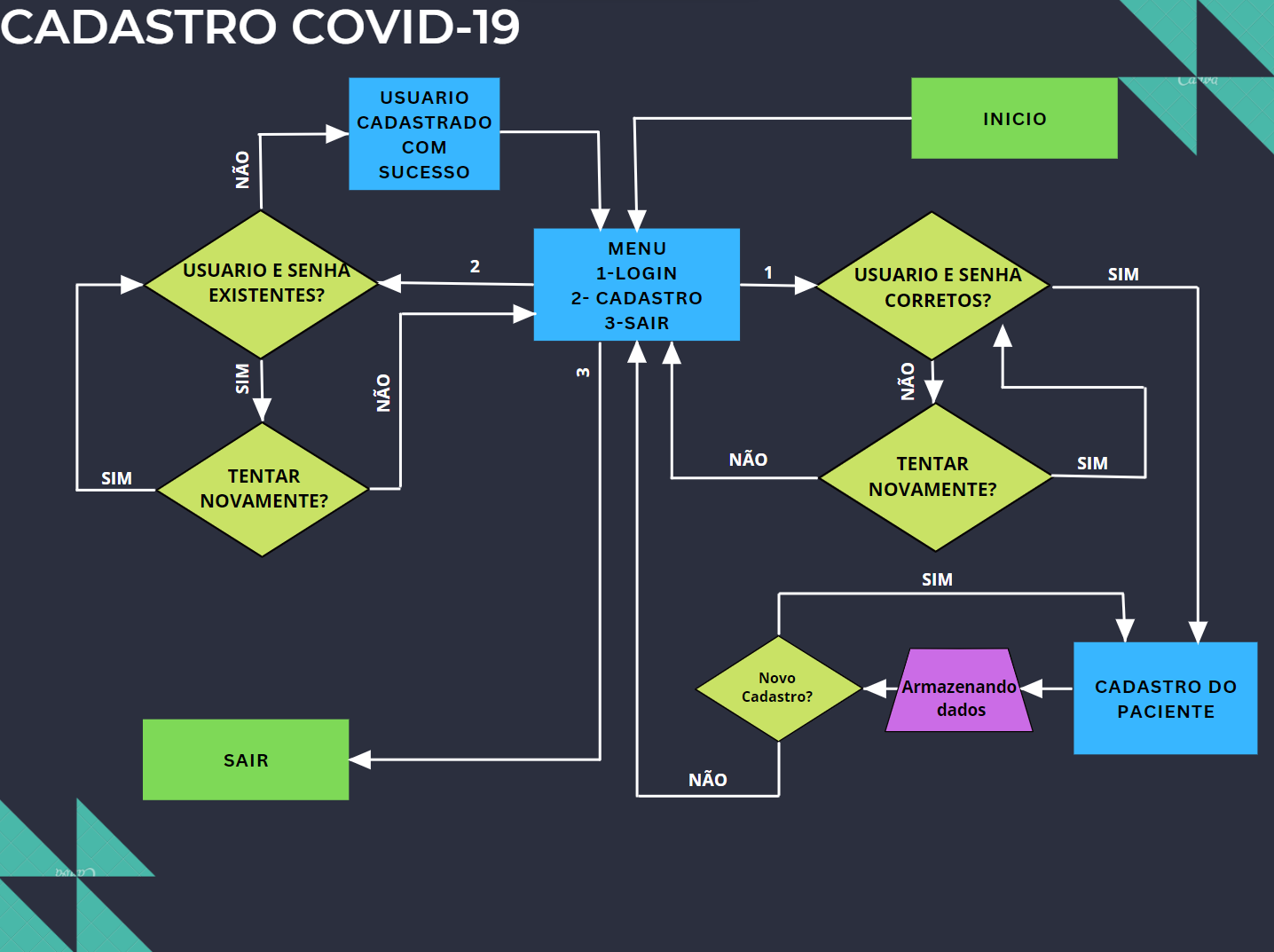


Fonte: Autoria própria.

**3.5 Fluxograma**

Um fluxograma de processo é um conjunto de símbolos usados para representar graficamente as etapas de uma operação, criando uma sequência de etapas para a execução de uma operação.

IMAGEM 09- Fluxograma

 Fonte: Autoria próprio

**4 Conclusão**

Devido ao surto global de covid-19 passou a ser detectadas as características dos sintomas do vírus, que em consequência da falta de informações e meios de se realizar o diagnóstico e monitorar, com aumento dos casos as equipes de cientistas e médicos analisaram os sinais para a identificação da doença por práticas de testes rápido de antígenos o resultado precisa ser confirmado por um exame de RT-PCR com fim de comprovar a presença do vírus e monitorar o paciente.

Por meio da tecnologia foi desenvolvido nesse projeto um sistema operacional capaz de cadastrar pacientes que possuem comorbidades (diabetes, obesidade, hipertensão, tuberculose, insuficiência cardíaca, etc.) e que são considerados acima de 65 anos para o controle e fácil acesso às informações, auxiliando também no registro de novos pacientes de forma rápida e eficaz.

O objetivo deste trabalho foi nos permitir através de conhecimentos adquiridos, desenvolver a prática de pesquisa sobre os conteúdos, métodos, técnicas e conhecimentos abordados nesta bimestre como finalidade de nos permitir desenvolver um trabalho técnico, documentado e estruturado, feito com um tema atual proposto para apresentarmos uma solução que facilite o monitoramento do grupo considerado de risco.

O Programa tem como objetivo o uso para cadastro, verificação e para atualização de registros de pacientes, este sistema é destinado para uso hospitalar, pode ser usado na recepção com objetivo de otimizar o tempo de atendimento e proporcionar fácil acesso às informações pelos agentes de saúde que serão compartilhadas com secretaria de saúde da cidade, pode-se cadastrar também novos usuários e gerenciar os existentes, no documento foi adicionado como visto prints da tela do programa e informações de como usar e instalar o mesmo, assim como as instruções para cadastro de novos usuários. Em função das regras e normas aprendidas ao longo da disciplina de Engenharia de software I, aplicamos para desenvolvimento de nosso programa o método de cascata que visa a utilização de controles e processos rigorosos, onde uma ação é feita somente após a conclusão de outra.

**REFERÊNCIAS**

WIKIPÉDIA. **Engenharia de *software***. Disponível em: <https://pt.wikipedia.org/wiki/Engenharia_de_software> .Acesso em: 20. nov. 2022.

FIA. **Desenvolvimento de Softwares: O que é, Como Funciona e Dicas.** Disponível em: [https://www.google.com/url?q=https://fia.com.br/blog/desenvolvimento-de-softwares/%23:~:text%3DAs%2520metodologias%2520de%2520desenvolvimento%2520s%25C3%25A3o%2520conjuntos%2520de%2520pr%25C3%25A1ticas,quem%2520faz%2520o%2520qu%25C3%25AA%252C%2520quando%252C%2520como%2520e%2520onde&sa=D&source=docs&ust=1668919344126331&usg=AOvVaw3maY5ok2\_c0ukwu7okaRHu](https://fia.com.br/blog/desenvolvimento-de-softwares/#:~:text=As%20metodologias%20de%20desenvolvimento%20s%C3%A3o%20conjuntos%20de%20pr%C3%A1ticas,quem%20faz%20o%20qu%C3%AA%2C%20quando%2C%20como%20e%20onde) **.**Acesso em: 20. nov. 2022.

OBJETIVE. **Modelo Ágil X Modelo Cascata: as principais diferenças nas metodologias.** Disponível em: [https://www.objective.com.br/insights/modelo-agil-modelo-cascata/](https://www.objective.com.br/insights/modelo-agil-modelo-cascata/%3e%3e%3e%3e...) .Acesso em:22. nov. 2022.

MONITORA. **Entenda a especificação de requisitos de software em projetos.** Disponível em: <https://www.monitoratec.com.br/blog/especificacao-de-requisitos-de-software/#:~:text=A%20especifica%C3%A7%C3%A3o%20de%20requisitos%20de%20software%20%C3%A9%20a,possui%20quatro%20fases%3A%20Estudo%20de%20viabilidade%20do%20software%3B>.Acesso em:23. nov.2022.

CODIFICAR.**O que são Requisitos Funcionais e Requisitos Não Funcionais?**

Disponível em:<https://codificar.com.br/requisitos-funcionais-nao-funcionais/> .Acesso em:22. nov.2022.

TUTORIALSPOINT. **Requisitos de Software.** Disponíveis em: <https://www.tutorialspoint.com/pg/software_engineering/software_requirements.htm#:~:text=Os%20requisitos%20de%20software%20s%C3%A3o%20descri%C3%A7%C3%A3o%20dos%20recursos,ou%20inesperado%20do%20cliente%20do%20ponto%20de%20vista.> .Acesso em:23.nov.2022.

WIKIPÉDIA. **Code::Blocks.** Disponível em:<https://pt.wikipedia.org/wiki/C_%28linguagem_de_programa%C3%A7%C3%A3o%29> .Acesso em: 22. nov. 2022.

WIKIPÉDIA. C (linguagem de programação). Disponível em:<https://pt.wikipedia.org/wiki/C_%28linguagem_de_programa%C3%A7%C3%A3o%29#:~:text=Para%20este%20fim%2C%20a%20linguagem%20C%20possui%20as,v%C3%A1rias%20opera%C3%A7%C3%B5es%20que%20n%C3%A3o%20fazem%20sentido%20Mais%20itens> .Acesso em:22.nov.2022.

# CANALTECH**. C: a linguagem de programação que está em tudo o que você usa.** Disponível em: <https://canaltech.com.br/software/c-a-linguagem-de-programacao-que-esta-em-tudo-o-que-voce-usa-19512/> .Acesso em:22.nov.2022.

TUTORIALSPOINT. **Requisitos de Software.** Disponível em: <https://www.tutorialspoint.com/pg/software_engineering/software_requirements.htm#:~:text=Os%20requisitos%20de%20software%20s%C3%A3o%20descri%C3%A7%C3%A3o%20dos%20recursos,ou%20inesperado%20do%20cliente%20do%20ponto%20de%20vista.> .Acesso em:23. nov. 2022.

MOKI.**Fluxograma de processo: o que é, importância e como montar.** Disponível em:<https://site.moki.com.br/fluxograma-de-processo/> .Acesso em:24. nov .2022.