

UNIVERSIDADE PAULISTA- EAD
PROJETO INTEGRADO MULTIDISCIPLINAR IV
CURSOS SUPERIORES DE TECNOLOGIA

**SISTEMA EM C PARA CADASTRAR PACIENTES DIAGNOSTICADOS
COM COVID-19**

POLO AMPARO
2021

UNIVERSIDADE PAULISTA- EAD
PROJETO INTEGRADO MULTIDISCIPLINAR IV
CURSOS SUPERIORES DE TECNOLOGIA

**SISTEMA EM C PARA CADASTRAR PACIENTES DIAGNOSTICADOS
COM COVID-19**

Nome: Giovanna dos Santos Ferreira
RA: 2157868

Orientador: Júlio César Severino

Curso: Análise e Desenvolvimento de Sistemas

Semestre: 2º

POLO AMPARO
2021

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me proporcionado chegar até aqui hoje.

Aos meus pais Abner e Marta por todo o apoio ao decorrer de todo esse Projeto Integrado Multidisciplinar IV, por todo o amor que eles me dão e encorajamento em tudo o que faço.

Ao meu irmão caçula Artur por ter sido crítico em relação a este trabalho e também me apoiar.

Ao Professor e Orientador Júlio César Severino por sua dedicação em suas orientações prestadas na elaboração deste trabalho.

Eu não teria conseguido sem vocês, muito obrigada a todos!

Resumo

O objetivo desse Projeto Integrado Multidisciplinar IV da UNIP, é o desenvolvimento de um Sistema de Monitoramento da Covid-19, que cadastrará pacientes diagnosticados. Salvará em formato de texto que serão repassados para a Secretaria Municipal de Saúde em caso de risco, ou seja, se for maior de 65 anos ou se o paciente tiver alguma comorbidade, tais como hipertensão, diabetes e outras, agregando e integrando todos os conhecimentos adquiridos nas disciplinas de Engenharia de Software e Linguagem e técnicas de programação.

Foi utilizada como ambiente de desenvolvimento a IDE Codeblocks e como Linguagem de Programação, a linguagem C.

O Projeto tem embasamento na íntegra nas regras da ABNT, respeitando as normas e os direitos autorais.

Palavras-chaves: Monitoramento. Software. Comorbidade.

Abstract

The objective of this UNIP Multidisciplinary Integrated Project IV is the development of a Covid-19 Monitoring System, which will register diagnosed patients. It will save in text format that will be transferred to the Municipal Health Department in case of risk, that is, if you are over 65 or if the patient has any comorbidity, such as hypertension, diabetes and others, adding and integrating all knowledge acquired in the disciplines of Software Engineering and Language and Programming Techniques.

The IDE Codeblocks was used as a development environment and the C language as a Programming Language.

The Project is fully based on the ABNT rules, respecting the rules and copyrights.

Key-words: Monitoring. Software. Comorbidity.

LISTA DE FIGURAS

Figura 1- Camadas de Engenharia de Software.....	9
Figura 2- Modelo Cascata.....	14
Figura 3- IDE CodeBlocks 20.03	17
Figura 4- Pasta Documentos	18
Figura 5- Acesso ao Sistema.....	18
Figura 6- Seleção do Arquivo do Programa.....	19
Figura 7- Acesso ao Código-Fonte	19
Figura 8- Tela Inicial do CodeBlocks	20
Figura 9- Compilação Efetuada	20
Figura 10- Rodando o Programa	21
Figura 11- Tela Inicial do SYSCOVID	21
Figura 12- Cadastro de Login e Senha	22
Figura 13- Efetuando o Login	22
Figura 14- SYSCOVID logado	23
Figura 15- CPF do Paciente	23
Figura 16- Data de Nascimento, E-mail e Telefones	23
Figura 17- Endereço	24
Figura 18- Comorbidades	24
Figura 19- Dados Cadastrados.....	24
Figura 20- Retorno ao Menu Inicial.....	25
Figura 21- Opções de Menu	25
Figura 22- Sistema Encerrado	25

Sumário

1. Introdução	7
2. Desenvolvimento	8
2.1 Engenharia de Software	8
2.1.1 Camadas da Engenharia de Software.....	9
2.1.2 Classificações de aplicações de software	9
2.1.3 Campos de Aplicação de Software	11
2.1.4 Qualidade de software	12
2.1.5 Processo de software	13
2.1.6 Ciclo de Vida.....	13
2.1.7 Métodos Ágeis.....	15
2.2 Linguagem e técnicas de programação	15
2.3 Manual de instalação do SYSCOVID	17
2.4 Compilando o programa.....	19
2.5 Manual do Usuário	21
3. Conclusão	25
4. Referências	26

1. INTRODUÇÃO

Esse Projeto Integrado Multidisciplinar tem como objetivo, desenvolver um Software para o monitoramento dos casos de Covid-19 em um hospital da rede privada, utilizando a disciplina de Linguagem e Técnicas de Programação.

O Sistema será desenvolvido na Linguagem de Programação em C, por ser uma linguagem estruturada e mais indicada no desenvolvimento de banco de dados para hospitais, que trabalham com um enorme volume de dados.

O Hospital fará o cadastro do paciente no sistema informando o nome, o CPF, o telefone, o endereço, data de nascimento, o e-mail, a data do diagnóstico e se o paciente tem ou não alguma comorbidade.

Se o paciente tiver alguma comorbidade, o próprio sistema salvará automaticamente a idade e o CEP deste paciente que será informado imediatamente à Secretaria Municipal de Saúde da cidade.

O Sistema também deverá calcular a idade do paciente e se o paciente tiver mais de 65 anos, o procedimento será o mesmo, ou seja, será salvo em um arquivo que posteriormente será informado à Secretaria Municipal de Saúde.

Também foi desenvolvido um Manual de Usuário do Software, para que os funcionários do Hospital possam aprender como utilizar esse Sistema da melhor maneira possível e com o máximo de rendimento.

Com o auxílio da disciplina Engenharia de Software I, o sistema foi desenvolvido de forma que as camadas da tecnologia ficassem coesas possibilitando que o mesmo desenvolva suas funções de maneira racional e dentro do prazo.

2. DESENVOLVIMENTO

2.1 Engenharia de Software

O Software está presente em todo lugar nos dias atuais; na aviação civil e militar, no entretenimento, na música, nas grandes empresas nacionais e multinacionais, no governo, no sistema financeiro como Bancos e Bolsa de valores.

No mundo atual, com o advento do Youtube e de cursos online, qualquer pessoa pode programar um computador, entretanto, programar para você mesmo é uma coisa, e programar para vários usuários é outra coisa.

Um programa pode conter um erro (bug) ou mais, e pode ou não ser simples de ser resolvido. Quando se trata de um Sistema Profissional, este programa deve ter uma estrutura baseada na Engenharia de Software, que visa não só a criação deste sistema, mas também sua manutenção, ampliação e implementação futura.

A Engenharia de Software se preocupa com o planejamento, qualidade com os prazos que serão acordados com o cliente, com os custos deste projeto e até com sua manutenção futura.

O Software nada mais é do que um conjunto de instruções (algoritmos) em uma Linguagem de Programação.

O Software não é visível e por este motivo o cliente exige que o projeto seja concluído o mais rápido possível. Mas, o mais oneroso de um projeto na verdade é a mão-de-obra que pode ser calculada por hora ou um preço fechado por Projeto. Existe também o fator tempo, que é estimado e não exato pois o cliente precisa do software para daqui a três meses e no entanto pode surgir um problema e demorar cinco meses. Deve-se levar em conta no Projeto todas estas variáveis para que o cliente fique satisfeito.

Um software pode também, a pedido do cliente ser modificado, alterado e o Desenvolvedor deve prever isto em seu projeto.

Outra questão importantíssima, é que o desenvolvimento Mobile vem ganhando espaço cada dia mais, e, o Desenvolvedor deve criar uma Aplicação que seja responsivo, ou seja, que se adapte em qualquer tipo de tela, smartphone, tablet, notebook, etc...

2.1.1 Camadas da Engenharia de Software

A Engenharia de software é uma tecnologia de camadas e essas quatro camadas são: Foco na qualidade, Processo, Métodos e Ferramentas.

Qualquer tipo de abordagem feita na engenharia tem é importante que esteja fundamentada no foco de qualidade.

A camada de processo é a base da engenharia de software, é nessa camada onde é determinada a metodologia, o controle do gerenciamento de projetos e estabelece o contexto no qual são aplicados métodos técnicos.

Os métodos fornecem informações técnicas para desenvolver software, incluem uma ampla variedade de tarefas com técnicas para cada fase do desenvolvimento do software.

Por fim as ferramentas fornecem suporte automatizado ou semi- automatizado para o processo e para os métodos. Quando as informações criadas pelas ferramentas são utilizadas por outras, elas são integradas e estabelecidas em um sistema que dá suporte ao desenvolvimento de software, a engenharia de software com o auxílio do computador.

Na Figura 1 é possível ver cada das camadas da engenharia de software:

Figura 1- Camadas de Engenharia de Software



Fonte: Livro Texto UNIP, 2021.

2.1.2 Classificações de aplicações de software

Existem Seis tipos básicos de aplicações para as classificações de softwares e são elas:

- **Sistemas de Processamento de Transações (SPT)**

São tipos de sistemas de informações ou de aplicações de software transacional. São utilizados nas operações das áreas de negócios. Alguns exemplos são: financeira, imobiliária, custo de transação, de banco de dados, processamento de transações etc.

- **Sistemas de Informações de Gestão (SIG)**

São sistemas que providenciam informações necessárias para fazer o gerenciamento das organizações. São sistemas que envolvem três recursos principais: a tecnologia, as próprias informações e as pessoas. Alguns exemplos são: a gestão mais eficiente do abastecimento que integra a conexão de uma cadeia de suprimentos, o que inclui fornecedores, fabricantes, atacadistas, varejistas e os consumidores finais; gestão de projetos, de contabilidade, de produção, de distribuição, de marketing, de vendas, de serviços, de documentos, de registros contábeis, dados sem registro e procedimentos, práticas e habilidades.

- **Sistemas de Apoio à Decisão (SAD)**

Diz respeito a um modelo genérico de tomada de decisão que analisa um número grande de variáveis para que seja possível o posicionamento correto diante de uma determinada questão. Alguns exemplos: pesquisas de bancos de dados, inteligência artificial métodos de simulação, engenharia de software e telecomunicações.

- **Sistemas de Informação Executiva (SIE)**

É um tipo de sistema de informações que gerenciam a informação e a tomada de decisão dos altos executivos. É uma forma especializada de SAD, um exemplo é que apresenta os resultados em telas gráficas e fáceis de usar.

- **Sistemas Especialistas (SE)**

Já os Sistemas Especialistas são programas de computadores que buscam emular a capacidade de decisão de um especialista humano. Muitos estudiosos afirmam que são programas de computador derivados de um ramo da pesquisa em computação chamado Inteligência Artificial (IA).

- **Sistemas de Automação de Escritório (SAE)**

É um Sistema responsável por automatizar as operações que são realizadas nos escritórios, aumentar a produtividade, reduzir custos e oferecer melhores resultados. Logo após essa automação, as atividades realizadas em escritórios, a organização das informações se tornou mais confiável e mais rápida.

2.1.3 Campos de Aplicação de Software

Existem Sete categorias de campos de aplicações de Softwares que são:

- **Software de Sistema**

É um conjunto de programas feito para atender a outros programas. Exemplos: compiladores, editores, utilitários para gerenciamento de arquivos, componentes de sistema operacional, drivers, software de rede e processadores de telecomunicações.

- **Software de Aplicação**

São programas totalmente independentes que buscam solucionar uma necessidade específica de negócios.

- **Software de engenharia/científico**

É uma variedade ampla de programas de cálculo em massa. Esses tipos de programas abrangem muitas áreas tais como astronomia, vulcanologia, análise de estresse automotivo, dinâmica orbital, projeto auxiliado por computador, biologia molecular, análise genética e meteorologia, entre outras.

- **Software embarcado**

É um sistema utilizado para implementar e controlar as características e as funções para os usuários e para o próprio sistema. Exemplos: controle do painel de um micro-ondas e controla as funções digitais de automóveis.

- **Software para linha de produtos**

É um sistema projetado para ter a capacidade de utilização de muitos clientes diferentes. Exemplo: produtos de controle de inventário.

- **Aplicações Web/aplicativos móveis**

Essas aplicações contemplam aplicativos voltados para redes e navegadores de dispositivos móveis.

- **Software de Inteligência artificial**

Esse tipo de software utiliza algoritmos não numéricos para solucionar problemas complexos. Exemplos: robótica, sistemas especialistas, reconhecimento de padrões (imagem e voz), redes neurais artificiais, prova de teoremas e jogos.

2.1.4 Qualidade de software

É o melhor caminho lógico a ser seguido para o desenvolvimento dos processos e do produto final.

Um assunto importante a ser tratado é que a condição do processo afeta no gerenciamento da qualidade, a qualidade é o sentimento de satisfação de uma pessoa obter um trabalho bem realizado.

Como a qualidade tem muitas dimensões, é muitas vezes difícil defini-la em um produto.

Existem duas maneiras principais de se enxergar a qualidade de um produto: do ponto de vista do produtor, onde ela está associada à fabricação de um produto que vá de encontro das necessidades de um cliente e do ponto de vista do próprio cliente que busca melhor preço, utilidade do produto, durabilidade, design, dimensão, cor, qual função desempenha, entre outras características, tornando-se assim um conceito multidimensional.

Para uma empresa, o conceito de qualidade deve ser definido de forma clara e objetiva.

Alguns aspectos para se obter a qualidade de um produto de acordo com a norma ISO/IEC 9126 são: funcionalidade, confiabilidade, usabilidade, eficiência, e portabilidade.

2.1.5 Processo de software

Em se tratando de processo de software, tanto a indústria, comércio e governos em geral, já entenderam e aceitaram que é de extrema importância uma melhoria significativa para o aumento de produtividade e de qualidade, o que todos estes entes não entendem, é a alta complexidade de isto ser realizado. Tanto por questões geográficas, quanto por questões financeiras ou técnicas.

2.1.6 Ciclo de Vida

O Processo como sendo a base do software possui vários ciclos de vida de um produto, que são exatamente as camadas do software; os processos, métodos/técnicas, as ferramentas e a qualidade. O que será acrescentado agora será o gerenciamento.

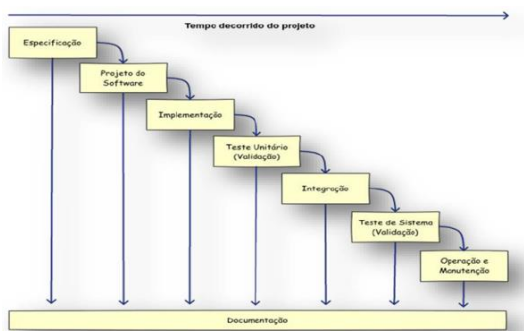
O ciclo de vida de um software consiste em Criação, evolução e decadência do mesmo.

Os principais modelos de ciclo de vida são:

- Codifica-remenda;
- Cascata;
- Incremental;
- RAD;
- Evolucionários;
- Especializados;
- UP;
- RUP;
- Processo Práxis;
- Processo cleanroom;

O modelo Cascata, que é o que foi escolhido, que é o mais tradicional, entretanto, uma etapa depende da outra, como mostrado na Figura 2.

Figura 2- Modelo Cascata



Fonte: <https://medium.com/contexto-delimitado/o-modelo-em-cascata-f2418addaf36>, 2021.

Na especificação, são colhidos os requisitos do sistema, ou seja, tudo aquilo que o cliente precisa.

No projeto de software, é desenvolvido a arquitetura do mesmo e dividido em unidades do programa.

Na etapa implementação, cada unidade de programa é codificada por um ou mais desenvolvedores, concomitantemente.

Ao final da implementação, deve ser feito um teste por unidades, chamado de Teste Unitário, onde se verifica se cada unidade funciona corretamente ou não. Este processo quando está concluído chamamos de Validação.

Ao concluir a validação e já com a documentação pronta, entregamos o produto de software ao cliente, que poderá detectar algum erro ou que descobriu alguma necessidade durante o uso do sistema. Esta etapa tem o nome de Operação e Manutenção.

2.1.7 Métodos Ágeis

Na década de 90 os clientes contratavam uma empresa para desenvolver um software e se deparava com diversos problemas. Os principais eram a má qualidade do software, funções excessivas, alto custo e a demora na entrega, muitas vezes um software demorava até três anos para ser entregue, o que hoje é um absurdo.

Pensando nisto, um grupo de desenvolvedores se reuniu para discutir sobre o assunto e chegaram à conclusão de que era preciso criar uma padronização nos processos de desenvolvimento de software, desde os requisitos do cliente até a entrega final do produto. Chamaram esta padronização de Métodos Ágeis, visto que os outros eram chamados de Métodos Pesados e foi então que surgiu o Manifesto Ágil, que é um conjunto de premissas e valores a serem seguidos por desenvolvedores aumentando assim a produtividade em até 800%.

Foram criados vários métodos ágeis e os principais são o SCRUM, ICONIX, XP, FDD, ASD, DSDM, CRYSTAL, TDD e MA. O mais utilizado destes métodos é o SCRUM.

As principais premissas dos Métodos Ágeis são:

- A interação entre pessoas;
- O funcionamento do software;
- Colaboração com o cliente;
- Responder às mudanças.

O gerenciamento se preocupa com o aumento da eficiência no desenvolvimento deste software com uma redução de custos.

2.2 Linguagem e técnicas de programação

Por volta de 1960 foi criada uma linguagem para rodar no sistema UNIX, esta linguagem se chamava ALGOL 60, também chamada de Linguagem A. Em 1967 Martin Richards criou a BCPL-Basic CPL, em 1970 Ken Thompson aprimorou a BCPL e apelidou-a de linguagem B.

O que aconteceu foi que tanto a linguagem BCPL, quanto a linguagem B eram muito limitadas, então Dennis Ritchie, do Bell Labs sentiu a necessidade de criar uma nova Linguagem que atendesse a demanda do mercado e mantivesse contato com o computador real. Esta linguagem foi batizada de Linguagem C que perdura até os dias de hoje.

A linguagem C diferentemente das outras linguagens, possui portabilidade, que nada mais é do que a facilidade de ser transportada para qualquer computador sem custos elevados, o que atraiu muitos programadores.

As Linguagens de programação são divididas em Alto e Baixo nível. A Linguagem de Baixo nível se assemelha a Linguagem de máquina, ou seja, códigos binários, já uma Linguagem de Alto nível, se assemelha a Linguagem humana.

As principais características desta Linguagem são a portabilidade, simplicidade, compilação e recursos de baixo nível.

A linguagem C é vastamente aplicada em jogos, projetos de automação, sistemas de cadastro para hospitais, editores de imagem e vídeo, mas principalmente em sistemas operacionais tais como Windows, Linux e Mac OS, por sua velocidade, exatidão e baixo índice de bugs.

Os desenvolvedores utilizam vários Ambientes de Desenvolvimento para programar em C, chamados de IDEs. Os principais são o Codeblocks, o Dev C++, Eclipse e Netbeans. Neste projeto utilizamos o Codeblocks conforme solicitado. Sua interface é excelente e muito intuitiva como mostra a figura xxx.

Os requisitos do sistema são classificados em Funcionais e Não-funcionais.

Os requisitos Funcionais são aqueles que o Software deverá realizar, já os Não-funcionais são aqueles ligados a tecnologia utilizada, desempenho, em qual plataforma deverá rodar, etc.

Este sistema foi batizado de SYSCOVID-Sistema de Monitoramento da Covid-19.

Os Requisitos Funcionais do SYSCOVID são:

- a) Cadastrar login e senha;
- b) Fazer login e senha;

- c) Cadastrar pacientes;
- d) Cadastrar CPF;
- e) Salvar os registros dos pacientes em arquivos.

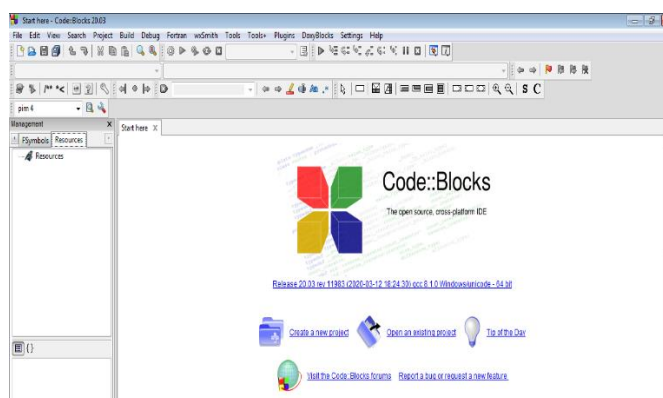
Os Requisitos Não-Funcionais do SYSCOVID são:

- a) Tecnologia –Linguagem de programação em C;
- b) Rodar em qualquer plataforma.

2.3 Manual de instalação do SYSCOVID

O SYSCOVID foi desenvolvido na IDE CodeBlocks 20.03, que é muito intuitiva conforme a Figura 3.

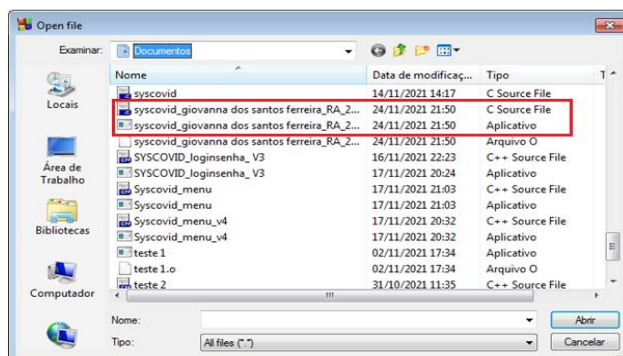
Figura 3- IDE CodeBlocks 20.03



Fonte: Autora, 2021.

A instalação do SYSCOVID será feita através de um pen-drive na pasta “documentos” do desktop. Existem dois arquivos conforme Figura 4.

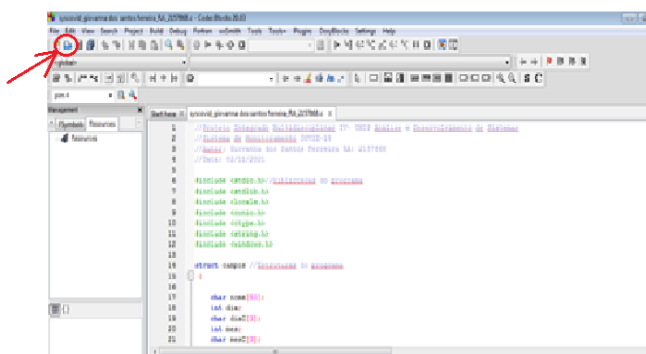
Figura 4- Pasta Documentos



Fonte: Autora, 2021.

Para entrar no sistema clique no ícone open da barra de ferramentas do Codeblocks conforme Figura 5.

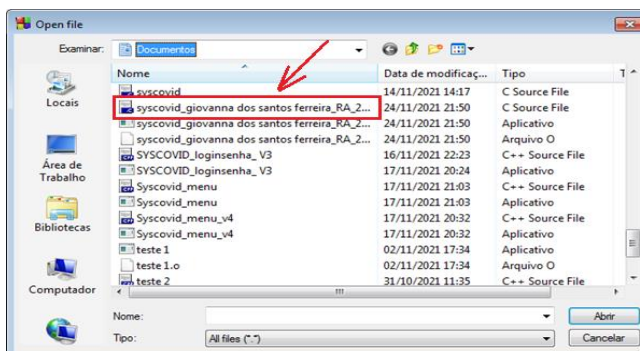
Figura 5- Acesso ao Sistema



Fonte: Autora, 2021.

Clicar no arquivo SYSCOVID indicado na Figura 6.

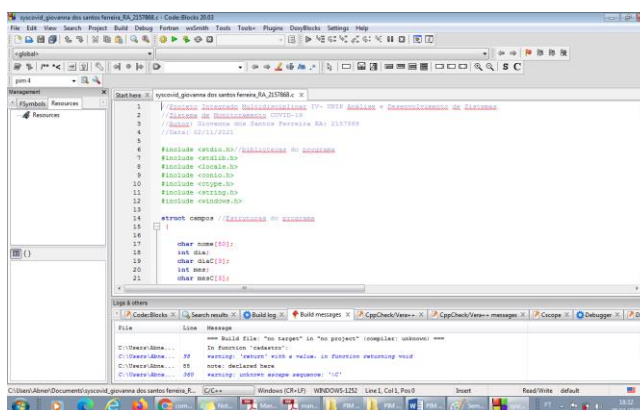
Figura 6- Seleção do Arquivo do Programa



Fonte: Autora, 2021.

A tela que deverá aparecer é a da Figura 7.

Figura 7- Acesso ao Código-Fonte

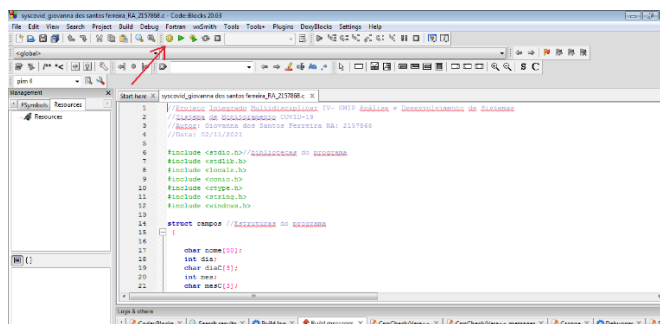


Fonte: Autora, 2021.

2.4 Compilando o programa

Para compilar o programa, ou seja, transformar tudo em linguagem de máquina, clique no ícone indicado na Figura 8.

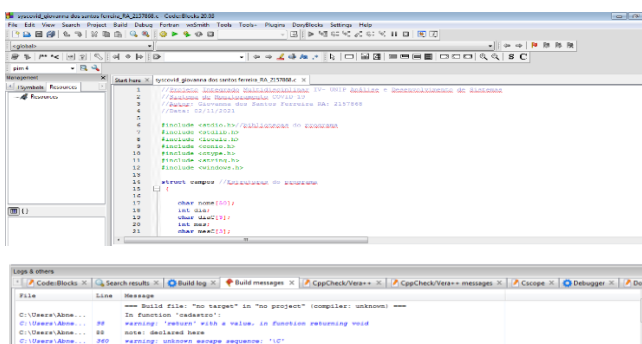
Figura 8- Tela Inicial do CodeBlocks



Fonte: Autora, 2021.

Aparecerá a tela da Figura 9, o programa foi compilado com sucesso.

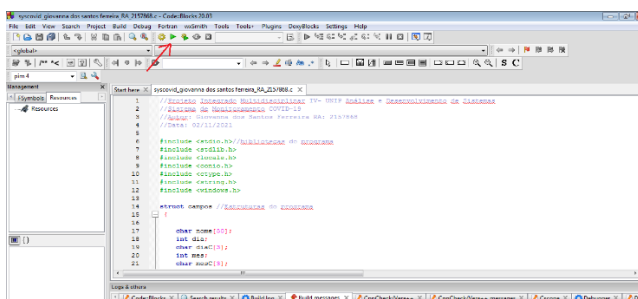
Figura 9- Compilação Efetuada



Fonte: Autora, 2021.

Clicar agora no ícone abaixo, para rodar o programa conforme a Figura 10.

Figura 10- Rodando o Programa

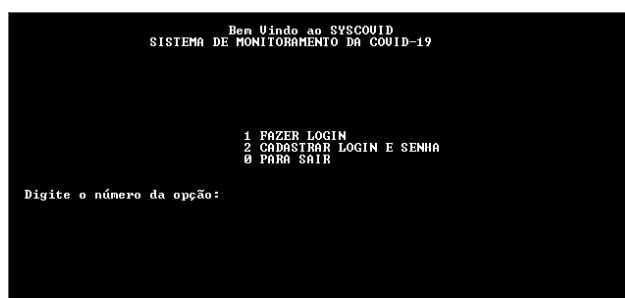


Fonte: Autora, 2021.

2.5 Manual do Usuário

Agora o programa foi aberto, aparece a tela inicial do SYSCOVID que possui um menu bem simples que consta de opções para fazer login, cadastrar login e senha e sair, conforme Figura 11.

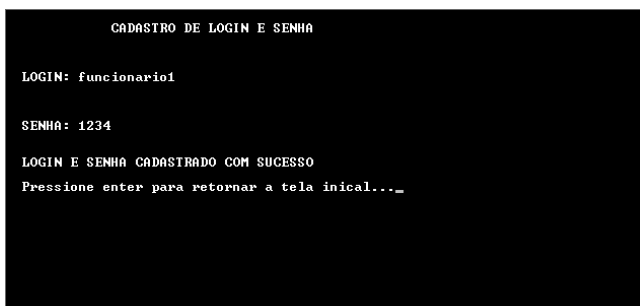
Figura 11- Tela Inicial do SYSCOVID



Fonte: Autora, 2021.

Quando o funcionário vai ter o primeiro contato com o programa, deverá cadastrar seu login e sua senha. Aqui o Login será funcionario1 (sem acentos) e a senha 1234, conforme Figura 12.

Figura 12- Cadastro de Login e Senha



Fonte: Autora, 2021.

Realizado o cadastro, o funcionário deverá selecionar a opção 1 e efetuar seu login para entrar no SYSCOVID, conforme Figura 13.

Figura 13- Efetuando o Login



Fonte: Autora, 2021.

Aparecerá na tela “Você está logado no SYSCOVID”.Logo em seguida, o funcionário digita a opção 1 para cadastrar o 1º paciente, conforme Figura 14.

Figura 14- SYSCOVID logado

```

LOGIN

LOGIN: funcionario1

SENHA: ****

Você está logado no SYSCOUID

- 1 CADASTRAR PACIENTES -
- 2 VOLTAR AO MENU INICIAL -

Digite o número da opção:

```

Fonte: Autora, 2021.

Será solicitado o CPF do paciente com 11 números e sem o traço, conforme Figura 15.

Figura 15- CPF do Paciente

```

*****DADOS DO PACIENTE*****
-----
Digite os 11 números do cpf sem ponto e traço
CPF: 17902937674_

```

Fonte: Autora, 2021.

Após isto, será solicitado o nome do paciente, a data de nascimento, o e-mail e dois telefones para contato, conforme Figura 16.

Figura 16- Data de Nascimento, E-mail e Telefones

```

*****DADOS DO PACIENTE*****
-----
Digite os 11 números do cpf sem ponto e traço
CPF: 17902937674
*NOOME: Marcus Azevedo
-----
*DATA DE NASCIMENTO
->DIA dd : 23
->MÊS mm : 12
->ANO aaaa : 1950
-----
EMAIL: marcus@outlook.com
-----
*NUMERO DE TELEFONE
***Cadastre dois números de telefone
Digite os números todos juntos XXXX...
Telefone residencial: 1138003456
-----
Telefone celular: 11994055789

```

Fonte: Autora, 2021.

Logo em seguida, será solicitado o endereço do paciente com CEP, nome da rua, número, complemento, cidade, bairro e Estado, conforme Figura 17.

Figura 17- Endereço


```

=====
Dados de Endereço
CEP: 07140000
=====
RUA: Das Rosas
=====
NÚMERO: 07
=====
COMPLEMENTO: apto
=====
CIDADE: Sumaré
BAIRRO: Jd. das Flores
ESTADO: SP

```

Fonte: Autora, 2021.

No próximo campo, será solicitado se o paciente possui ou não alguma comorbidade, em caso de comorbidade positiva, informe a comorbidade, conforme Figura 18.

Figura 18- Comorbidades

```

-0 paciente possui algum tipo de comorbidade? -
-1 SIM -
-2 NÃO -
Escolha a opção desejada: 1
COMORBIDADE DO PACIENTE: diabetes e hipertensão_

```

Fonte: Autora, 2021.

Assim que der um Enter, aparecerá na tela “dados cadastrados com sucesso!”, e solicitará um novo cadastro. Se for digitado 1, abrirá um novo cadastro, conforme Figura 19.

Figura 19- Dados Cadastrados

```

-0 paciente possui algum tipo de comorbidade? -
-1 SIM -
-2 NÃO -
Escolha a opção desejada: 1
COMORBIDADE DO PACIENTE: diabetes e hipertensão
DADOS CADASTRADOS COM SUCESSO!

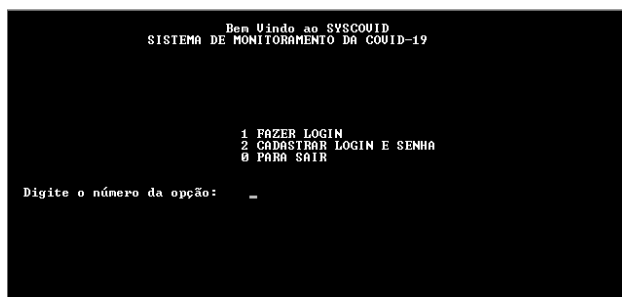
-Deseja realizar um novo cadastro? -
- 1 SIM -
- 2 NÃO -
Digite o número da opção:

```

Fonte: Autora, 2021.

Se digitar a opção 2, voltará ao Menu inicial, conforme Figura 20.

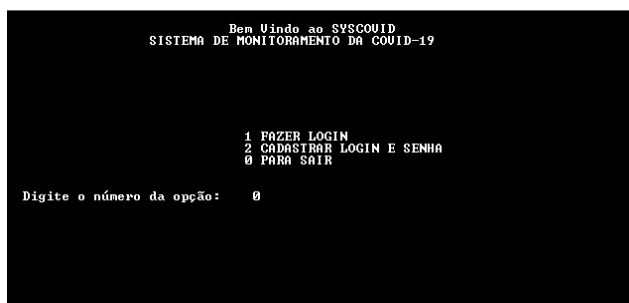
Figura 20- Retorno ao Menu Inicial



Fonte: Autora, 2021.

Se o funcionário quiser sair do SYSCOVID deverá digitar a opção 0, conforme Figura 21.

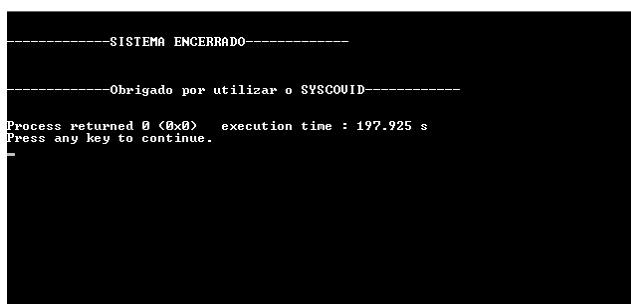
Figura 21- Opções de Menu



Fonte: Autora, 2021.

Aparecerá na tela “Sistema encerrado” e “Obrigado por utilizar o SYSCOVID” , conforme Figura 22.

Figura 22- Sistema Encerrado



Fonte: Autora, 2021.

3. CONCLUSÃO

Neste Projeto Multidisciplinar IV foram realizadas diversas atividades relacionadas com as disciplinas de Engenharia de Software e Linguagem e Técnicas de Programação.

A Aplicação de Software que foi utilizada no SYSCOVID foi o SAD (Sistemas de Apoio à Decisão), pois é o sistema que é utilizado para tomar decisões e analisar um grande número de variáveis.

Já o modelo de software escolhido foi o modelo em Cascata que é um modelo padrão e de muita confiabilidade, apesar de uma etapa depender exclusivamente da outra.

Percebemos no desenvolvimento deste projeto, que a Engenharia de Software e a Linguagem de Programação devem caminhar juntas para que o produto de software seja entregue dentro do prazo, com o máximo de qualidade e um custo de tempo e financeiro mínimos ou mitigados.

Para que seja possível esta entrega rápida, faz-se necessário um estudo aprofundado dos Métodos Ágeis em especificamente do SCRUM para que os resultados sejam os esperados pelo cliente.

4. REFERÊNCIAS

SOMMERVILLE, Ian. **Engenharia de Software**. 10. ed. São Paulo: Pearson, 2018. Disponível em: <https://plataforma.bvirtual.com.br>
Acesso dia: 18/05/2021

FILHO, Wilson de Pádua. **Engenharia de Software: Fundamentos, métodos e padrões**. 4. ed. LTC, 2019.

Acesso dia: 19/11/2021

DEITEL, Paul; DEITEL, Harvey. **C: Como programar**. 6. ed. São Paulo: Pearson, 2011.

Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/186034>.

Acesso dia: 20/11/2021

PEZZÉ, M.; YOUNG, M. **Teste e análise de software: processos, princípios e técnicas**. 1. Ed. Porto Alegre: Bookman, 2008.

Acesso dia: 20/11/2021

PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de Software**. 8. Ed. AMGH, 2016.

Disponível em: <https://integrada.minhabiblioteca.com.br>

Acesso dia: 23/11/2021

Disponível em: <https://www.youtube.com/watch?v=j5mCirZD-0s&t=597s>

Acesso dia: 24/11/2021

Disponível em: <https://medium.com>

Acesso dia: 25/11/2021