UNIVERSIDADE PAULISTA – UNIP EaD

Projeto Integrado Multidisciplinar

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

PROJETO CADASTRO DE PACIENTES DA COVID-19 EM C

SÃO PAULO – UNIP

2022

BEATRIS ANTUNES SILVA - 2273384

BIANCA ANTUNES SILVA - 2292834

MÁRCIO SANTANA PORTELA - 2292870

PROJETO CADASTRO DE PACIENTES DA COVID-19 EM C

Projeto Integrado Multidisciplinar em Análise e Desenvolvimento de Sistemas

Projeto Integrado Multidisciplinar para obtenção do título de tecnólogo em Análise e Desenvolvimento de Sistemas, apresentado à Universidade Paulista – UNIP EaD.

Orientador (a): Profa. Dra. Vanessa Lessa

SÃO PAULO – UNIP

2022

**RESUMO**

Esta pesquisa tem o intuito de desenvolver um sistema de cadastro de casos da COVID-19 em linguagem C, bem como a utilização dos fundamentos de engenharia de software para administrar o progresso do projeto.

O sistema será utilizado pelos hospitais para registro e acompanhamento de casos positivos de COVID-19, onde os pacientes cadastrados serão verificados pelo software e se está no grupo de risco, arquivados externamente, permitindo assim consultas.

O projeto foi dividido em três etapas:

* Na primeira etapa houve uma discussão sobre a ponderação e ranqueamento e sendo discutidos os benefícios e planeado um diagrama do caso de uso para documentar.
* Na segunda etapa, estabelecida a metodologia XP como utilizada na gestão do progresso do software, tendo eficiência e praticidade especialmente em cenários onde a vida útil do sistema é relativa.
* Na terceira e última etapa, o resultado foi obtido por codificação, o software desenvolvido de maneira simplificada para autenticação e controle do paciente em seu escopo geral, além deste, contendo os procedimentos de tempo de execução e testes neste documento.

**Palavras-Chave**: COVID-19, Software.

**ABSTRACT**

This research has the intention to develop a COVID-19 case registration system in C language, as well as to use software engineering knowledge to guide the progress of the project.

The system will be used by hospitals to register and monitor positive cases of COVID-19, where registered patients will be verified by the software and as if the patient belongs to the risk group, filed externally, allowing consultations.

The project was divided into three stages:

* In the first step there was a discussion about weighting, ranking and discussing benefits being counted, as well as planning a use case diagram to document.
* In the second stage, the XP methodology was established as used in the management of software progress, with efficiency and practicality, especially in scenarios where the useful life of the system is relative.
* In the third and last step, the result was obtained by coding, the software developed in a simplified way for patient authentication and control in its general scope, in addition to this, containing the execution time procedures and tests in this document.

**Keywords**: COVID-19, software.

Sumário

[INTRODUÇÃO 6](#_Toc120288756)

[REQUISITOS DA FERRAMENTA DE MONITORAMENTO PARA CASOS DE COVID-19 7](#_Toc120288757)

[REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS 8](#_Toc120288758)

[REQUISITOS FUNCIONAIS 8](#_Toc120288759)

[REQUISITOS NÃO FUNCIONAIS 8](#_Toc120288760)

[REQUISITOS DO PROJETO 9](#_Toc120288761)

[DIAGRAMA DE CASO DE USO DO SOFTWARE 10](#_Toc120288762)

[METODOLOGIAS ÁGEIS 10](#_Toc120288763)

[EXTREME PROGRAMMING 11](#_Toc120288764)

[DESENVOLVIMENTO E RESULTADO DO SOFTWARE 13](#_Toc120288765)

[MANUAL DE COMO UTILIZAR 13](#_Toc120288766)

[CONCLUSÃO 17](#_Toc120288767)

[REFERÊNCIAS 18](#_Toc120288768)

# INTRODUÇÃO

Devido a nova ameaça sanitária do coronavírus várias entidades públicas e privadas estão em busca de inovação tecnológica relacionada à pesquisa em benefício da saúde e ajudar a monitorar os casos para assim controlar a disseminação.

Por conta da escala global de disseminação, a indústria de software desempenha um papel importante para ajudar os profissionais de saúde e autoridades a prevenir o avanço.

Objetivando desenvolver um sistema de monitoramento em linguagem C, tendo a principal função ser utilizado pelos hospitais para registrar e monitorar casos positivos de COVID-19.

Visando calcular e analisar se o paciente possui comorbidades e/ou a idade, nesses casos os dados ficam hospedado externamente e serão encaminhados ao departamento de saúde local para cadastro e acompanhamento.

Existem dois tipos de requisitos:

* Requisitos Funcionais: Estabelece a função do sistema e um panorama geral do software. Sendo documentações de serviços que o sistema deve fornecer, além do comportamento de entradas e situações específicas.
* Requisitos não funcionais: Sendo definido como propriedades ou proibições de tempo e padrões do progresso de desenvolvimento, sendo possível obter mais ferramentas no controle do vírus.

# REQUISITOS DA FERRAMENTA DE MONITORAMENTO PARA CASOS DE COVID-19

Definido por requerimentos de clientes no projeto a ser desenvolvido, um requisito é a particularidade que um software apresenta para problemas reais, sendo a circunstância essencial para satisfazer o objeto.

Em se tratando de software requisitados, por exemplo, uma forma que o sistema propõe ou uma proibição no desenvolvimento, nas duas ações, sendo necessário chegar a um acordo para elucidar o problema do cliente.

A manutenção é essencial para o sucesso do software, fornecendo a alicerce para estimar, modelar, projetar, testar e manter. Portanto, estando presentes durante toda a vida útil do software.

Ao início, os requisitos devem ser compreendidos e documentados. Além de conduzir práticas de monitoramento de qualidade para verificar, validar e garantir sua qualidade.

Sendo importante a gestão do progresso dos requisitos, sem garantia de estabilidade e podendo sofrer alterações. Dessa forma, mantendo equilíbrio.

Podendo ser caracterizados por:

* Erro: Termo usado para se referir ao comportamento inesperado do software do sistema.
* Personalização: Isso agrega ao que foi solicitado no projeto original. Ex: incorporação com outros softwares.
* Solicitações: Aprimoramento recursos que podem adicionar coisas ao sistema, como uma coluna extra em um botão extra etc.

## REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

### REQUISITOS FUNCIONAIS

Podemos colocar no requisito funcional a concretização ou uma requisição formulada por um software.

O sistema possui diversas funcionalidades:

* Adicionar/remover/editar nome na tela de manutenção;
* Relatórios de vendas e clientes;
* Pagamentos por cartão;
* Análise e modificação de dados;
* Consulta de receita ou estoque.

### REQUISITOS NÃO FUNCIONAIS

Esses tipos de requisitos são chamados de propriedades de qualidade, sendo essenciais na criação de sistemas. Geralmente tratados como suposições e restrições técnicas não funcionais, sendo todos como essencialidades que não substitui recursos.

Sem limitações ao sistema de método, tempo, espaço ou desenvolvimento operacional, esse requisito possui algumas propriedades de exemplo e métricas:

* Medição em kytes e chips de RAM;
* Tempo por segundo;
* Medição de portabilidade;
* Praticidade em uso por janelas ou período de treinamento;
* Confiabilidade em análise de falhas.

## REQUISITOS DO PROJETO

Os requisitos foram descritos em uma escala definem o valor/importância bem como a dificuldade de implementação, logo após cada requisito:

Tabela – Requisitos do Software

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

## DIAGRAMA DE CASO DE USO DO SOFTWARE

Demonstração do funcionamento do software utilizando diagramação, simplificando as interações entre o usuário e o sistema.

Figura – Digrama de caso de uso

Interface gráfica do usuário

Descrição gerada automaticamente

# METODOLOGIAS ÁGEIS

A engenharia de software é uma técnica onde os conceitos são desenvolvidos por um especialista em uma determinada área. O software em sua estrutura de dados permite manipulá-lo para controlá-lo e desenvolvê-lo na resolução de problemas, a partir desse desenvolvimento, os métodos passaram a organizar e gerenciar sua estrutura com eficiência, sendo conhecidos como metodologias de desenvolvimento de software.

Ao longo do tempo, na busca de melhoria, de custos e de aproveitamento de cada um, as metodologias foram aprimoradas com a criação do Manifesto Ágil, possibilitando as tecnologias de informação tornar-se mais adaptáveis, voltadas para as reais necessidades dos clientes.

No método ágil, o desenvolvimento adquiriu aquele em que o software é criado em colaboração com equipes multidisciplinares que possuem uma boa autonomia no trabalho.

Tabela - Metodologias tradicionais X Metodologias ágeis

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Tendo diferenças fundamentais entre a metodologia tradicional e o método ágil como questões burocráticas, a metodologia em cascata tradicional tornou-se bastante pesada, para metodologias emergentes que foram referidas como metodologias "em cascata" leves.

Figura – Modelo em cascata do progresso do software

Uma imagem contendo Linha do tempo

Descrição gerada automaticamente

## EXTREME PROGRAMMING

Inerente ao manifesto ágil, diversos frameworks surgiram utilizando seus pilares ágeis, sendo um software simplificado especialmente em projetos rápidos ou relativamente pequenos.

Segundo Beck, o criador da metodologia XP, o sucesso estrutural da metodologia vem da busca pela satisfação do cliente.

*Figura 3 – Ciclo do Extreme Programming*

Diagrama

Descrição gerada automaticamente

Seus objetivos eram:

* Satisfação do cliente;
* Atender as exigências do cliente;
* Ser altamente orientado para o trabalho;
* Manter todos focados na criação de qualidade.
* Usa padrões por meio da programação extrema.

É preeminente na figura que todos os passos realizados passam por validação e um retorno direto, sem essencialmente exigir uma figura comercial intermediária entre desenvolvimento e os stakeholders como no Scrum, por exemplo.

Desta forma, é um recurso identificável com requisitos que mudam constantemente devido à demanda do cliente, e em detrimento dele, muito menos escrutínio inicial no estudo de onde a arquitetura geralmente visa não a escalabilidade, mas a resolução de problemas momentâneos.

O XP foi desenvolvido para ser aplicado em equipes de dois a dez programadores, sendo possível criar sistemas de maior qualidade, que são produzidos em menos tempo e econômico.

É possível atingir esses objetivos por meio de princípios e práticas, que diferem significativamente da forma tradicional de desenvolvimento de software.

Resumo das principais características:

* Fase curta do software;
* Proeminência mínima na escalabilidade do aplicativo;
* Resolve problemas diretos e momentâneos;
* Burocracia mínima e ênfase em documentação;
* Mudanças frequentes devido à demanda do cliente.

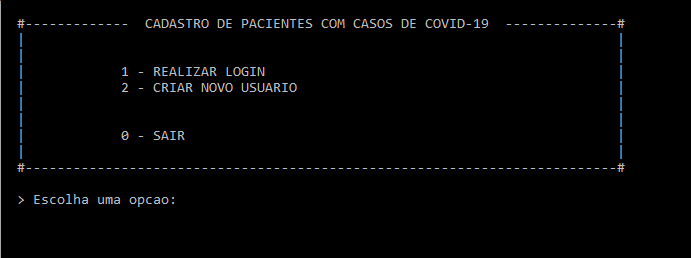
# DESENVOLVIMENTO E RESULTADO DO SOFTWARE

## MANUAL DE COMPILAÇÃO

1. Instale o CodeBlocks no seu sistema. Você pode acessar o site oficial da IDE em: http://www.codeblocks.org/;
2. Para clonar o repositório, abra o terminal do git e digite: “git clone https://github.com/biancaantunes98/PIM-IV.git”;
3. No seu gerenciador de arquivos, navegue até o diretório PIM-IV e abra o arquivo 'pim.cbp' utilizando o CodeBlocks. O projeto será aberto.
4. Para executar, com o projeto aberto no CodeBlocks, vá em 'Build > Build and Run' e o programa será executado.

## MANUAL DE COMO UTILIZAR

1 - Ao rodar o projeto, a seguinte tela vai aparecer:



*2 - Ao selecionar a opção 1, será direcionado para a tela de login:*

Texto

Descrição gerada automaticamente

- Ao selecionar 2, será direcionado para a tela de cadastro de novo usuário:

Texto

Descrição gerada automaticamente

- Após inserir um novo login e senha e apertar enter, um novo usuário será criado:

Texto

Descrição gerada automaticamente

- Ao realizar login, o usuário será direcionado à tela principal:

Texto

Descrição gerada automaticamente

- Caso selecione a opção 1, será direcionado à tela de consulta de pacientes cadastrados no sistema:

Texto

Descrição gerada automaticamente

- Caso a opção 2 seja selecionada, será direcionado à tela de consulta de pacientes no grupo de risco:

Texto

Descrição gerada automaticamente

- Já, ao selecionar a opção 3, a tela para qual será direcionado é a tela de cadastro de novos pacientes:

Texto

Descrição gerada automaticamente

- Ao preencher todas as informações e pressionar enter o programa irá perguntar se deseja salvar:

Texto

Descrição gerada automaticamente

# CONCLUSÃO

Com o auxílio dessa pesquisa, foi possível alcançar a aplicação prática de conhecimentos básicos sobre engenharia de software em prol da saúde pública da população, habilidade de autenticação de usuários para permitir que tenha registros de casos de pacientes de COVID-19, realizando segregação e notificação de casos de risco para auxiliar no acompanhamento. Como primeiro passo, os requisitos devidamente pré-estipulados o uso de arquivos TXT para armazenar dados externamente, em geral foi considerado como uma decisão arquitetônica, a visualização facilita o controle desses dados na exportação, sendo facilmente legíveis por meio de editores de texto ou de outros sistemas externos que suportam a serialização de dados nesse formato.

Foram postos em pauta os benefícios do uso de metodologias ágeis em geral, superando o trivial de gerenciar projetos que eram usados no passado, bem como a escolha de metodologia sob o nome de Extreme Programming como ideal para o projeto.

O código foi versionado usando as ferramentas Git e Github disponível em: https://github.com/biancaantunes98/Pim-IV.

Por fim, os resultados e o procedimento de teste foram apresentados por meio de um manual neste documento.

# REFERÊNCIAS

**Codeblocks**. Disponível em: <http://www.codeblocks.org/downloads/>. Acesso em: 18 de novembro de 2022.

**Covid-19**. Disponível em: <https://www.saopaulo.sp.gov.br/coronavirus/>. Acesso em: 21 de novembro de 2022.

**Documentação da Linguagem C**. Disponível em: <https://learn.microsoft.com/pt-br/cpp/c-language>. Acesso em 18 de novembro de 2022.

E-ARCHITECTS INC. **Manifesto for Agile Software Development**. Agile Manifesto. 2001. Disponível em: https://agilemanifesto.org/. Acesso em: 21 novembro 2022.

PRESSMAN, Roger S. **Engenharia de Software** - 7.ed. AMGH Editora, f. 1006, 2008. 2011 p.

SOMMERVILLE, Ian. **Engenharia de software** (8a. ed.)., f. 286. 2007. 572 p.

**PIM IV**. Disponível em: <https://github.com/Solimar13/PIM-IV>. Acesso em: 16 de novembro de 2022.