

**Documentação de um**

**Produto de Software**

**Bem-Star System**

**Nome dos Alunos:**

**Anderson Vitor da Silva Barros**

**Antonio Tonon Gregorio Franco**

**Bruno Vitor**

**Gustavo Gino Terezo**

**Lucas Souza**

**Matheus Afonso Batista**

**2023**

**ÍNDICE DETALHADO**

Sumário

[1. Introdução 3](#_Toc136109514)

[1.1. Tema 3](#_Toc136109515)

[1.2. Objetivos a serem alcançados 3](#_Toc136109516)

[1.3. Escopo principal 3](#_Toc136109517)

[2. Integração continua (IC) 4](#_Toc136109518)

[2.1. Plano das integrações 4](#_Toc136109519)

[2.2. Documentação das integrações 4](#_Toc136109520)

[3. Qualidade aplicada 5](#_Toc136109521)

[3.1. Técnicas de qualidade aplicadas ao sistema computacional 5](#_Toc136109522)

[3.2. Documentação da aplicação da ferramenta 5](#_Toc136109523)

[4. Testes 7](#_Toc136109524)

[4.1. Estratégias e roteiro de testes 7](#_Toc136109525)

[4.2. Documentação dos testes unitários e integrados 7](#_Toc136109526)

[4.3. Documentação dos testes manuais e automatizados 7](#_Toc136109527)

[5. Estimativa de software 8](#_Toc136109528)

[5.1. Documentação da análise por pontos de função 8](#_Toc136109529)

[5.2. Valorização do sistema computacional 9](#_Toc136109530)

[6. Considerações finais 10](#_Toc136109531)

[7. Referências 11](#_Toc136109532)

# Introdução

## Tema

ODS: 3 – Saúde e Bem-Estar

Tratando-se de um dos muitos temas importantes que a ONU definiu como um objetivo mundial, escolhemos a Saúde e Bem-Estar pelo motivo de a mesma, muitas vezes ser desorganizada e de difícil acesso para as pessoas mais necessitadas.

## Objetivos a serem alcançados

O objetivo será criar um sistema onde haverá a organização das pessoas a partir de cadastros, para que seja feito um levantamento das pessoas, e será feito no mesmo sistema a compilação do histórico de saúde dos cadastrados dos mesmos. Feito isso, será mais fácil identificar padrões de doenças e irá corroborar com os tratamentos por áreas urbanas. Devido a implementação do sistema, para os médicos que avaliarem os pacientes através de consultas, será mais fácil checar todo o histórico de saúde do paciente. Com todo o sistema em funcionamento, será mais fácil gerir o histórico de saúde das pessoas e unificá-lo no país, tornando-o padrão para as instituições de saúde e gerando um sistema único de verificação de todo o quadro do paciente.

## Escopo principal

O sistema, terá incluso dois projetos como desenvolvimento inicial: **SB\_001 e SB\_002.**

**SB\_001**

Identificação e cadastro das pessoas no sistema.

Nesse projeto, será feito o levantamento das pessoas que se enquadram na definição da ODS e cadastro das mesmas.

Com a ajuda do governo, será feito mutirões de funcionários que estão contribuindo para o projeto, onde os mesmos irão fazer visitas em residências para coletar informações das pessoas que residem no local e a partir disso, será feito o cadastro da pessoa no sistema.

**SB\_002**

Cadastro das doenças existentes e que afetam as pessoas.

Nesse projeto, será feito o levantamento das doenças que existem no planeta e será feita o cadastro das mesmas no sistema e com isso, será mais prático identificar o tipo de doença que o paciente avaliado possui. OBS: esse projeto está linkado com o projeto SB\_001, pois para a inclusão da doença no prontuário do paciente, o mesmo precisará estar cadastro no sistema.

Com a ajuda do banco de dados do governo sobre doenças e sintomas, será feito o cadastro das mesmas no sistema, para que seja possível gerenciar atualização do prontuário do paciente. A partir das doenças cadastradas no sistema, o médico conseguirá fazer a alteração no histórico de saúde do paciente, incluindo o resultado dos exames.

# Integração continua (IC)

## Plano das integrações

As entregas serão feitas semanalmente com o uso de versionamento de projeto, com a ferramenta Git e GitHub.

## Documentação das integrações

| Data | Tarefa |
| --- | --- |
| 13/03/2023 | Definição do tema do projeto |
| 16/03/2023 | Validação do tema do projeto |
| 20/03/2023 | Documentação do tema do projeto |
| 22/03/2023 | Separação das tarefas dos projetos |
| 28/03/2023 | Separação dos tópicos de entrega do projeto |
| 31/03/2023 | Entrega de parte do projeto |
| 01/04/2023 | Planejamento dos requisitos e separação de tarefas |
| 25/04/2023 | Elaboração da análise por pontos de função |
| 04/05/2023 | Separação das funcionalidades do sistema para análise |
| 14/05/2023 | Entrega da análise do sistema por pontos de função |
| 25/05/2023 | Separação das tarefas das entregas finais do projeto |
| 26/05/2023 | Entrega de parte da documentação do projeto |
| 27/05/2023 | Finalização dos templates de apresentação do projeto |
| 29/05/2023 | Finalização do projeto |

# Qualidade aplicada

## Técnicas de qualidade aplicadas ao sistema computacional

Usaremos a norma ISO/IEC 9126 e Scrum.

## Documentação da aplicação da ferramenta

Dentro do Scrum, teremos iterações que onde a cada iteração entregamos software, ou incrementos de software, potencialmente usável e de acordo com a necessidade do cliente. E a cada nova iteração, temos “feedback” do que foi entregue e que utilizamos para melhorar o produto (sempre de acordo com a prioridade do cliente).

O “feedback” do cliente haveria de qualquer forma, seja apresentando o produto ao final de uma iteração, seja ao final de todo o ciclo de desenvolvimento, o que normalmente acarreta em alterações no código. Entretanto, se estas alterações forem feitas no final do projeto, isto também pode acarretar efeitos colaterais indesejados, ou seja, podemos antecipá-las, o que impediria este tipo de problema.

Já com a ISO/IEC 9126 poderemos ser capazes de medir as características da qualidade de produtos de um software, nela podemos medir a funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e a portabilidade do software. Essa norma propõe atributos de qualidade, e são distribuídos em seis características e cada características é dividida em sub características, são elas:

**Funcionalidade -** Verifica a capacidade de o software prover as funcionalidades que foram definidas e requisitadas. As sub características são: Adequação, onde mede o quanto as funcionalidades são adequadas às necessidades do usuário. Precisão, é a capacidade de o software fornecer os resultados precisos de acordo com o que foi solicitado. Interoperabilidade, como o software interage com outro sistema. Segurança, a capacidade de o software proteger os dados do sistema.

**Confiabilidade** – Verifica se o produto continua no nível de desempenho das condições que foram estabelecidas. As sub-características são: Maturidade, onde se evita que as falhas do software sejam decorrentes de outro software. Tolerância a falhas, o software se mantém funcionando independente dos defeitos que ocorrem nele ou externamente. Recuperabilidade, o software se recupera depois de uma falha e restabelece seus níveis de desempenho.

**Usabilidade** – O software é compreendido, o funcionamento aprendido e é atraente para o usuário. Possui quatro sub-características, inteligibilidade onde é avaliado se o usuário compreende as funcionalidades do sistema e avalia se satisfaz as suas necessidades, apreensibilidade identifica a facilidade de aprendizado do sistema para os usuários, operacionalidade verifica a maneira que o produto facilita a operação por parte do usuário, atratividade são as características que podem atrair um potencial usuário.

**Eficiência** – Se o tempo de execução e os recursos são compatíveis com o nível de desempenho do sistema. Comportamento em relação ao tempo, avalia se o tempo de resposta estão de acordo com as requisições e Utilização de recursos, identifica tanto os recursos consumidos quanto a capacidade quem o sistema possui para utilizar os recursos. São as sub-características.

**Manutenibilidade** – A capacidade que o software tem em ser modificado, melhorado e sua capacidade de inclusão de extensões. As sub-características: Analisibilidade verifica a facilidade de diagnosticar os problemas e as suas causas. Modificabilidade, a capacidade do software em ser alterado. Estabilidade, identifica a capacidade de o sistema evitar efeitos colaterais devidos as modificações.

**Portabilidade** – Verifica a capacidade do sistema se transferido de um ambiente para outro. Adaptabilidade, capacidade para ser instalado, coexistência e capacidade para substituir são as sub-características

# Testes

## Estratégias e roteiro de testes

A listagem abaixo identifica os itens de software, hardware e elementos de suporte do produto que foram identificados como objetivos dos testes. Esta lista representa os itens que serão testados.

* Sistema de cadastro de histórico de doenças dos pacientes
* Formulário de cadastro de pacientes
* Registro de histórico de doenças
* Funcionalidade de consulta do histórico de doenças
* Permissões de acesso para pacientes, médicos e administradores

## Documentação dos testes unitários e integrados

O teste de volume sujeitará o sistema de cadastro de histórico de doenças dos pacientes a grandes volumes de dados, a fim de determinar se serão atingidos limites que farão com que o software deixe de funcionar. Esse teste também identifica o volume ou carga máxima contínua que o sistema poderá suportar durante um determinado período de tempo. Por exemplo, será realizado um teste de volume para verificar como o sistema se comporta ao processar um grande número de registros de pacientes e histórico de doenças.

## Documentação dos testes manuais e automatizados

Os testes automatizados serão executados utilizando as seguintes ferramentas:

* Selenium WebDriver para automação de testes de interface de usuário.
* JUnit como framework de testes unitários para testes automatizados de componentes e classes.

# Estimativa de software

## Documentação da análise por pontos de função

A partir do levantamento do diagrama de classes do projeto, foi possível realizar a análise por Ponto de Função para cada funcionalidade do sistema. A análise por Ponto de Função é uma técnica de medição que se baseia em medidas estabelecidas pela metodologia utilizada, a fim de determinar o valor de uma funcionalidade ou do sistema como um todo.

Nesse contexto, a análise por Ponto de Função é uma abordagem objetiva e padronizada para medir o tamanho funcional de um software, levando em consideração as funcionalidades fornecidas ao usuário final. Essa técnica se concentra nas características que agregam valor para o usuário, como entrada de dados, saída de informações, consultas, processamentos internos e interações com o usuário.

Ao utilizar o diagrama de classes como base, é possível identificar as principais funcionalidades do sistema e classificá-las de acordo com as categorias definidas pela metodologia de análise por Ponto de Função. Cada funcionalidade é avaliada levando-se em conta fatores como complexidade, número de transações, número de dados de entrada e saída, entre outros.

A partir dessas medidas, é possível atribuir um valor em pontos de função para cada funcionalidade, representando sua contribuição para o tamanho total do sistema. Esses pontos de função podem ser utilizados para estimar esforço de desenvolvimento, tempo de implementação, custo e outros aspectos relacionados ao projeto de software.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Método** | **Função** | **PF** | **Total para o Método** | | |
| gerenciarProjetos() | Criar | 10 | 43 | | |
| Pesquisar | 13 |
| Atualizar | 10 |
| Deletar | 10 |
| gerenciarDoenças() | Criar | 10 | 43 | | |
| Pesquisar | 13 |
| Atualizar | 10 |
| Deletar | 10 |
| gerenciarUsuários() | Criar | 10 | 43 | | |
| Pesquisar | 13 |
| Atualizar | 10 |
| Deletar | 10 |
| gerenciarPaciente() | Criar | 10 | 43 | | |
| Pesquisar | 13 |
| Atualizar | 10 |
| Deletar | 10 |
| gerenciarFuncionário() | Criar | 10 | 43 | | |
| Pesquisar | 13 |
| Atualizar | 10 |
| Deletar | 10 |
| gerenciarMédico() | Criar | 10 | 43 | | |
| Pesquisar | 13 |
| Atualizar | 10 |
| Deletar | 10 |
| gerenciarAdministrador() | Criar | 10 | 43 | | |
| Pesquisar | 13 |
| Atualizar | 10 |
| Deletar | 10 |
| gerenciarHistóricoDoenças() | Criar | 10 | 43 | | |
| Pesquisar | 13 |
| Atualizar | 10 |
| Deletar | 10 |
| visualizarProntuário() | Pesquisar | 13 | 13 | | |
|  | |
|  | |
|  | |
| login() | Pesquisar | 10 | 10 | | |  | |
|  | |
|  | |
|  | |
| **Total da APF** | | **357** |  |  |  | |  |

## Valorização do sistema computacional

Utilizando-se como base os valores cobrados por ponto de função no mercado atual, onde os mesmos podem varia de R$ 250 a R$ 1000, foi estipulado para o desenvolvimento do sistema, o valor de R$ 200 por ponto de função, podendo-se é claro, ser negociado.

O tempo que será necessário para o desenvolvimento por ponto de função, foi estimado em 1 hora, ou seja, o total para o desenvolvimento dos pontos de função seria igual a 357 horas, o equivalente a 45 dias, sendo que seriam usados 8 horas diárias para o desenvolvimento.

# Considerações finais

O presente trabalho abordou a temática das entregas de engenharia de software e gestão de software, com foco no desenvolvimento de um software relacionado aos Objetivos de Desenvolvimento Sustentável (ODS) da ONU, principalmente na área da saúde. Ao longo da pesquisa, foi possível explorar os desafios e as melhores práticas envolvidas nesse contexto, bem como os benefícios e impactos positivos que podem ser alcançados por meio da aplicação da engenharia de software e da gestão adequada nesse tipo de projeto.

Um dos principais pontos destacados neste trabalho é a importância de alinhar as entregas de engenharia de software e a gestão do projeto. Os ODS fornecem uma estrutura globalmente reconhecida e amplamente aceita para direcionar esforços em direção ao desenvolvimento sustentável, e a utilização de software como ferramenta nessa jornada pode contribuir significativamente para alcançar essas metas.

A engenharia de software desempenha um papel fundamental na criação de soluções tecnológicas inovadoras que podem melhorar a saúde e o bem-estar das pessoas. Ao aplicar as práticas adequadas de engenharia de software, como o desenvolvimento ágil, a prototipagem rápida e a colaboração multidisciplinar, são possíveis criar softwares mais eficientes, confiáveis e adaptáveis às necessidades dos usuários e das comunidades envolvidas.

Além disso, a gestão de software desempenha um papel crucial no sucesso do projeto. A gestão adequada envolve a definição clara de requisitos, o planejamento estratégico, o acompanhamento do progresso, a alocação eficiente de recursos e a garantia da qualidade. A adoção de boas práticas de gestão de software contribui para o cumprimento dos prazos, a redução de custos e a entrega de um software de alta qualidade, capaz de atender às necessidades dos usuários e promover um impacto positivo na área da saúde.

No contexto dos ODS da ONU, é fundamental considerar a sustentabilidade do software desenvolvido. Isso implica em garantir que o software seja projetado para ser escalável, modular e fácil de manter, permitindo que seja adaptado e atualizado de acordo com as necessidades futuras. Além disso, a sustentabilidade também envolve a consideração dos impactos ambientais, sociais e econômicos do software ao longo de seu ciclo de vida, desde o desenvolvimento até o descarte adequado.

Por fim, este trabalho ressalta a importância de uma abordagem holística para as entregas de engenharia de software e gestão de software relacionadas aos ODS da ONU na área da saúde. Isso implica em envolver as partes interessadas relevantes, como profissionais da saúde, usuários finais e comunidades afetadas, desde as fases iniciais do projeto até a sua conclusão. A colaboração e o engajamento de todas as partes interessadas são essenciais para garantir que o software desenvolvido atenda às necessidades reais, seja efetivo e promova resultados sustentáveis na área da saúde.

Em suma, a combinação da engenharia de software e da gestão adequada é uma poderosa ferramenta para impulsionar o alcance dos Objetivos de Desenvolvimento Sustentável da ONU na área da saúde. Ao adotar as melhores práticas e considerar a sustentabilidade em todas as etapas do processo de desenvolvimento de software, é possível criar soluções eficazes e impactantes, contribuindo para o avanço da saúde global e o bem-estar das comunidades.

# Referências

1. Valderramas, E.L.B. (2023). Gestão e qualidade de software. Universidade São Judas Tadeu – Mooca. Material da aula disponibilizado por PDF.

2. Nações Unidas. (2023). Objetivos de Desenvolvimento Sustentável: Saúde e Bem-Estar. Recuperado de https://brasil.un.org/pt-br/sdgs/3.

3. OpenAI. (2021). ChatGPT [Modelo de linguagem]. OpenAI API. https://openai.com.