

**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](#_Toc136110256)

[1.1. Tema 3](#_Toc136110257)

[1.2. Objetivos a serem alcançados 3](#_Toc136110258)

[1.3. Escopo principal 3](#_Toc136110259)

[2. Modelo de processo de desenvolvimento 5](#_Toc136110260)

[2.1. Modelo de processo de desenvolvimento 5](#_Toc136110261)

[2.2. Embasamento histórico/teórico 5](#_Toc136110262)

[2.3. Justificativa da escolha do modelo 5](#_Toc136110263)

[3. Engenharia de requisitos 6](#_Toc136110264)

[3.1. Relatório de viabilidade do sistema computacional 6](#_Toc136110265)

[3.2. Documentação do levantamento dos requisitos 8](#_Toc136110266)

[3.3. Análise e negociação de requisitos 10](#_Toc136110267)

[3.4. Especificação dos requisitos do sistema computacional 11](#_Toc136110268)

[3.5. Documentação a validação dos requisitos 15](#_Toc136110269)

[4. Arquitetura de software 16](#_Toc136110270)

[4.1. Tipo de arquitetura de software 16](#_Toc136110271)

[4.2. Embasamento histórico/teórico 16](#_Toc136110272)

[4.3. Justificativa da escolha da arquitetura 17](#_Toc136110273)

[5. Framework ágil 18](#_Toc136110274)

[5.1. Framework 18](#_Toc136110275)

[5.2. Documentação dos entregáveis/sprints 18](#_Toc136110276)

[6. Considerações finais 21](#_Toc136110277)

[7. Referências 22](#_Toc136110278)

# 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.**

**Projeto 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.

**Projeto 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.

# Modelo de processo de desenvolvimento

## Modelo de processo de desenvolvimento

Modelo de processo: **Espiral.**

## Embasamento histórico/teórico

Modelo espiral criado por Barry Boehm, com a ideia de melhorar o modelo incremental. Esse modelo consiste em trabalhar o tempo todo com riscos, dividindo o projeto em outros menores. Devido ao movimento em espiral que o projeto irá tomar rumo, é visado a eliminação de todos os problemas que surgirão no projeto. Ao longo de cada loop na espiral, é desenvolvido uma versão de software, onde a mesma é testada e atualizada a cada novo loop.

Principais melhorias no modelo espiral:

* Planejamento
* Análise de riscos
* Engenharia
* Atualização feita pelo cliente

O modelo espiral exige uma consideração direta dos riscos técnicos em todas as etapas do projeto e, se adequadamente aplicado, deve reduzir os riscos antes que eles se tornem problemáticos (PRESSMAN, 2006).

## Justificativa da escolha do modelo

Modelo de processo escolhido pois: foi o mais bem visto pela equipe do projeto e devido a sua grande flexibilidade com possibilidade de revisão de erros e correção dos mesmos em um estado breve do desenvolvimento do software.

# Engenharia de requisitos

## Relatório de viabilidade do sistema computacional

**Sobre o sistema**

O sistema irá armazenar os projetos desenvolvidos para a ODS de Saúde e Bem-Estar, para que sejam mais fáceis de serem catalogados para serem apresentados aos investidores que podem vir a patrocinar tais projetos.

Os projetos que forem desenvolvidos poderão ser implementados facilmente, pois não será necessário a implementação ou criação de novas tecnologias. E como os projetos são voltados para a área da saúde, é necessário a aprovação dos órgãos públicos para aprovação dos mesmos, liberando assim uma verba para concretizá-los.

O sistema poderá e deverá ser integrado aos sistemas já existentes, pois já existem sistemas voltados para área de saúde e caso não seja possível integrar todos eles com este que está sendo desenvolvido, não serão possíveis atingir o objetivo principal desse projeto.

**Análise de mercado**

Como é possível ver, atualmente o sistema de saúde brasileiro está muito desorganizado (ao menos o sistema de saúde público), com problemas na área de agendamento, atendimento, entre outras áreas. Com o sistema que estamos desenvolvendo, poderão ser armazenados inúmeros projetos para solucionar esses problemas e também futuros problemas que podem vir a ser criados, fazendo com que até mesmo projetos que não foram usados atualmente possam vir a ser usados no futuro.

No entanto, criar todos esses projetos leva tempo e dinheiro para serem desenvolvidos, por causa disso temos a necessidade de nos aliar a iniciativas públicas, pois o governo irá conseguir sustentar o desenvolvimento de tais projetos e com a realização dos mesmo o sistema de saúde será implementado.

**Desenvolvimento do Sistema**

Para desenvolver um sistema de armazenamento de projetos de ODS de Saúde e Bem-Estar, é necessário ter uma equipe de desenvolvimento de software com habilidades em programação. O sistema será programado na linguagem JAVA e será acessível por meio de uma interface de usuário simples e intuitiva.

A criação do sistema terá um investimento significativo. A equipe de desenvolvimento terá que trabalhar em tempo integral por alguns meses para construir o sistema, e haverá custos contínuos para a manutenção e atualização do sistema.

No entanto, os benefícios potenciais para as instituições médicas são notáveis. Um sistema de armazenamento de ODSs voltadas para área da saúde pode vir a solucionar problemas que já existem a muitos anos, melhorando desde o atendimento ao paciente até pesquisas de campo voltadas à saúde e ao bem-estar.

**Conclusão**

Com base na análise de mercado, técnica e financeira, concluímos que o desenvolvimento desse sistema é viável. Embora haja alguns desafios a serem enfrentados, os benefícios potenciais para as instituições de saúde são significativos. A implementação de um sistema desse tipo pode melhorar significativamente a área da saúde no país.

## Documentação do levantamento dos requisitos

|  |  |  |
| --- | --- | --- |
| **Dia** | **Requisitos funcionais** | **Descrição** |
| **29.03.2023** | RF01 | Os profissionais de saúde poderão: fazer o gerenciamento geral dos pacientes (cadastro, remoção, alteração e consulta); |
| RF02 | O sistema deverá possuir a opção de consulta, inclusão, alteração e exclusão de dados do paciente; |
| RF03 | O sistema deverá possuir a consulta das seguintes funcionalidades para os perfis de pacientes: Nome, Idade, Endereço, CPF, Histórico de doenças. Onde estas funcionalidades podem ser alteradas pelos usuários; |
| **02.04.2023** | RF04 | O usuário administrador deve ser capaz de agendar consultas para o paciente, sendo necessário os dados do paciente através do próprio sistema; |
| RF05 | O usuário paciente deve ser capaz de visualizar um histórico de consultas realizadas anteriormente, e visualizar o resultado das consultas; |
| **04.04.2023** | RF06 | O sistema deve armazenar pedidos de exames médicos de instituições públicas e privadas; |
| **06.04.2023** | RF07 | Os usuários com acesso de administrador poderão fazer o gerenciamento de novos projetos (cadastro, alteração, consulta e exclusão dos mesmos;  Para o cadastro de novos projetos referentes a ODS, será necessário preencher os seguintes campos: código do projeto, descrição, nome da ODS, nome do responsável, telefone do responsável, data da criação do projeto e status do mesmo; |
| RF08 | A cada alteração feita no registro médico do paciente, o médico responsável pela consulta, deverá: incluir sua senha e nome para constar no log de alterações. |
|  |  |  |
| **Dia** | **Requisitos não-funcionais** | **Descrição** |
| **29.03.2023** | RNF01 | O sistema deve ter uma política de privacidade rigorosa; |
| RNF02 | O software deve atender aos padrões de idioma local; |
| RNF03 | A plataforma de desenvolvimento da empresa é Java; |
| RNF04 | O banco de dados deve ser desenvolvido com MySQL; |
| RNF05 | O sistema deverá conter criptografia de ponta para a segurança do sistema como um todo. |
| **02.04.2023** | RNF06 | Conexão com o banco de dados do governo: o sistema requer conexão com os bancos de dados do governo para o seu funcionamento. Se não houver conexão, não será possível fazer as operações de consulta, atualização e cadastro de doenças e pacientes; |
| RNF07 | Histórico de consultas: o sistema deve armazenar as informações de cada consulta realizada pelo paciente, para que posteriormente seja possível consultar o histórico do paciente. |
| **04.04.2023** | RNF08 | O sistema deverá rodar no mínimo em um sistema operacional Windows 7; |
| RNF09 | O sistema deverá fazer backups diariamente para assegurar as informações. |
| **06.04.2023** | RNF10 | O sistema deverá se adaptar à medida que projetos forem implementados, como por exemplo, atualização de interface com conexão do banco de dados se o projeto necessitar do mesmo; |
| RNF11 | O sistema deverá possuir verificação de acesso para os usuários decorrente do tipo de perfil do mesmo; |
| RNF12 | O sistema deve cumprir todas as leis e regulamentos aplicáveis; |
| RNF13 | O sistema deve ser fácil de manter e atualizar; |
| RNF14 | Com a inclusão de novos exames, de forma automática, será gerado a data e hora de inclusão no sistema do mesmo; |
| RNF15 | A cada 15 dias o sistema irá enviar uma confirmação de segurança de sua senha, para a checagem se a mesma não foi alterada por terceiros. |
| RNF16 | Interface de fácil uso; |
| RNF17 | Compatibilidade com outros sistemas de saúde; |
| RNF18 | Acesso de forma rápida e eficiente. |

## Análise e negociação de requisitos

Segundo o levantamento dos requisitos e a análise feita pela equipe e pelos stakeholders referente aos mesmos, segue resultado da análise:

|  |  |  |
| --- | --- | --- |
| **Dia** | **Requisitos funcionais** | |
| **29.03.2023** | RF01 | |
| RF02 | |
| RF03 | |
| **02.04.2023** | RF04 | |
| RF05 | |
| **04.04.2023** | RF06 | |
| **06.04.2023** | RF07 | |
| RF08 | |
|  | |  |  | | --- | --- | | Legenda |  | | Requisitos aprovado |  | | Requisito negado |  | |  |
| **Dia** | **Requisitos não-funcionais** | |
| **29.03.2023** | RNF01 | |
| RNF02 | |
| RNF03 | |
| RNF04 | |
| RNF05 | |
| **02.04.2023** | RNF06 | |
| RNF07 | |
| **04.04.2023** | RNF08 | |
| RNF09 | |
| **06.04.2023** | RNF10 | |
| RNF11 | |
| RNF12 | |
| RNF13 | |
| RNF14 | |
| RNF15 | |
| RNF16 | |
| RNF17 | |
| RNF18 | |

## Especificação dos requisitos do sistema computacional

**Requisitos funcionais**

|  |  |
| --- | --- |
| **RF** | **01** |
| **Nome** | Gerenciamento dos pacientes |
| **Descrição** | Os profissionais de saúde poderão: fazer o gerenciamento geral dos pacientes (cadastro, remoção, alteração e consulta); |
| **Atores** | Médicos e pacientes |
| **Prioridade** | Essencial |
| **Entradas** | Nome, Idade, Endereço, CPF, Histórico de doenças. |
| **RNF associados** | RNF04, RNF06 e RNF07 |
| **Saídas** | Paciente cadastrado / removido / atualizado / pesquisado com sucesso. |
|  |  |
| **RF** | **02** |
| **Nome** | Opções do sistema |
| **Descrição** | O sistema deverá possuir a opção de consulta, inclusão, alteração e exclusão de dados do paciente; |
| **Atores** | Pacientes |
| **Prioridade** | Essencial |
| **Entradas** | Nome, Idade, Endereço, CPF, Histórico de doenças. |
| **RNF associados** | RNF04, RNF06, RNF07 e RNF14 |
| **Saídas** | Paciente cadastrado / removido / atualizado / pesquisado com sucesso. |
|  |  |
| **RF** | **03** |
| **Nome** | Consulta de paciente |
| **Descrição** | O sistema deverá possuir a consulta das seguintes funcionalidades para os perfis de pacientes: Nome, Idade, Endereço, CPF, Histórico de doenças. Onde estas funcionalidades podem ser alteradas pelos usuários médicos; |
| **Atores** | Médicos e pacientes |
| **Prioridade** | Essencial |
| **Entradas** | Nome, Idade, Endereço, CPF, Histórico de doenças. |
| **RNF associados** | RNF04, RNF06 e RNF07 |
| **Saídas** | Paciente consultado com sucesso |
|  |  |
| **RF** | **05** |
| **Nome** | Consulta de histórico de doenças |
| **Descrição** | O usuário paciente deve ser capaz de visualizar um histórico de consultas realizadas anteriormente, e visualizar o resultado das consultas; |
| **Atores** | Paciente |
| **Prioridade** | Essencial |
| **Entradas** | CPF |
| **RNF associados** | RNF04, RNF06 e RNF07 |
| **Saídas** | Consulta de histórico de doenças realizada com sucesso. |
|  |  |
| **RF** | **07** |
| **Nome** | Funções de administrador |
| **Descrição** | Os usuários com acesso de administrador poderão fazer o gerenciamento de novos projetos (cadastro, alteração, consulta e exclusão dos mesmos; para o cadastro de novos projetos referentes a ODS, será necessário preencher os seguintes campos: código do projeto, descrição, nome da ODS, nome do responsável, telefone do responsável, data da criação do projeto e status do mesmo; |
| **Atores** | Administradores |
| **Prioridade** | Essencial |
| **Entradas** | Código do projeto, descrição, nome da ODS, nome do responsável, telefone do responsável, data da criação do projeto e status do mesmo. |
| **RNF associados** | RNF10 |
| **Saídas** | Projeto cadastrado com sucesso |

**Requisitos não-funcionais**

**RNF01** – O sistema deverá possuir uma política de privacidade rigorosa pois é necessário seguir as normas regidas pelo governo e éticas morais estabelecidas pela sociedade.

**RNF02** – O software deverá possuir o padrão de idioma local pelo motivo de ser mais acessível a todos os usuários com a língua nativa do país. Lembrando que o sistema irá prover acesso a todas as classes de pessoas, ou seja, do mais carente ao maior retentor de capital, e com isso, para abranger todas as camadas, será usado a língua nativa para o sistema.

**RNF03** – A plataforma será desenvolvida em Java, pois irá abranger ao paradigma de orientação a objetos.

**RNF04** – O banco de dados para o armazenamento das informações do sistema, será desenvolvido com a plataforma MySQL, pois o mesmo possui uma alta relevância no mercado e é de fácil uso para todos os desenvolvedores.

**RNF05** – O sistema deverá conter criptografia de ponta para a segurança de todos os cadastrados no sistema, mantendo assim a sua privacidade intacta.

**RNF06** – O sistema deverá possuir conexão contínua para que se possa acessar os dados do banco de dados, pois sem o mesmo, não será possível fazer as solicitações ao banco de dados para o funcionamento correto do sistema.

**RNF07** – O sistema deverá possuir o método de pesquisa das consultas realizadas pelos pacientes, para que os mesmos consigam acompanhar a situação atual da sua saúde e buscar o melhor método para tratar os problemas se o tiverem.

**RNF09** – Uma política para a segurança de dados do sistema, será o backup diário dos dados do sistema, para que o trabalho feito não seja perdido e as informações fiquem intactas caso haja algum problema envolvendo os servidores.

**RNF10** – O sistema irá se adaptar conforme os novos projetos da ODS forem surgindo, ou seja, a cada implementação ou elaboração de projetos novos no sistema, se necessário, será implementado novas funções no sistema, referindo-se aquele novo projeto implantado.

**RNF11** – O sistema deverá possuir nível de acesso decorrente do tipo de usuário, ou seja, para administradores, eles possuirão acesso geral a toda funcionalidade do sistema; já para os de nível de acesso para o seu referente projeto, o mesmo terá acesso somente as funções para a sua parte, exemplo: no projeto de saúde, o paciente poderá somente consultar seus resultados de exame; o usuário médico poderá fazer todo o gerenciamento de paciente, incluindo novos exames, alterando dados do paciente e buscando os mesmos.

**RNF12** – O sistema deverá cumprir as leis e regulamentos aplicáveis pela nação e por normas de qualidade especificados pelas instituições internacionais para o melhor desenvolvimento do software e garantir uma excelente qualidade do software na produção do mesmo.

**RNF13** – O sistema deverá ser de fácil manutenção, pois como o sistema se tornará unificado, o mesmo deverá ser de fácil entendimento pelos diversos técnicos que atuarão na manutenção do mesmo.

**RNF14** – O sistema atualizará automaticamente a hora e data de alguma inclusão de exame em históricos de doenças dos pacientes para a melhor verificação do estado atual do paciente decorrente das datas em cada exame.

**RNF16** – A interface deverá ser de fácil utilização, pois será utilizado por muitos usuários que não são da área técnica de tecnologia, ou seja, será utilizado por todo tipo de pessoa com todo tipo de conhecimento sobre tecnologia.

**RNF17** – Como o sistema será unificado em todo o país, independente do lugar em que se faça exames e seja gerado resultados, os mesmos será implementado no sistema único do país, tornando assim o sistema de utilização e consulta geral, ou seja, possuirá compatibilidade com outros sistemas de saúde.

**RNF18** – O sistema deverá possuir acesso de forma rápida e eficiente, pois a atualização do prontuário dos pacientes é de extrema importância, fazendo com que os mesmos fiquem satisfeitos com o serviço prestado e possam acompanhar a sua situação de saúde.

## Documentação a validação dos requisitos

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Item para verificação** | **SIM** | **NÃO** |
| **1** | Cada requisito está descrito com clareza, concisão e sem ambiguidade? | X |  |
| **2** | Existem requisitos conflitantes? |  | X |
| **3** | Existem requisitos implícitos? |  | X |
| **4** | Os requisitos exibem a distinção clara entre funções, dados e restrições? | X |  |
| **5** | As restrições e dependências foram claramente descritivas? | X |  |
| **6** | Existem requisitos que contêm algum nível desnecessário de detalhe do projeto? |  | X |
| **7** | Os requisitos definem todas as informações a serem apresentadas aos usuários? | X |  |
| **8** | Os requisitos descrevem as respostas do sistema ao usuário devido às  condições de erro? |  | X |
| **9** | Existem situações não tratadas pelos requisitos que precisam ser consideradas? | X |  |
| **10** | O documento contém realmente toda a informação prometida em sua introdução? | X |  |

|  |
| --- |
| Assinatura dos responsáveis do projeto |
| Anderson Vitor da Silva Barros  Antonio Tonon Gregorio Franco  Bruno Victor  Gustavo Gino Terezo  Lucas Souza  Matheus Afonso Batista |
|  |
|  |
| Assinatura do requisitante do projeto |
| Universidade São Judas Tadeu |
|  |

# Arquitetura de software

## Tipo de arquitetura de software

Os tipos de arquiteturas que mais se adequam ao projeto desenvolvido são:

* Data Abstraction and Object-Oriented Organization (Abstração de Dados e Organização Orientada a Objetos)
* Event-based, Implicit Invocation (Invocação Implícita, Baseada em Evento)
* Layered Systems (Sistemas em Camadas)

## Embasamento histórico/teórico

**Abstração de Dados e Organização Orientada a Objetos**

Muito material e muitos estudos sobre orientação a objetos surgiram após a concepção do paradigma nos anos 70, entre eles a concepção da UML, que abraçou os conceitos para trazer uma nova linguagem que fosse capaz de ajudar na modelagem de software, em especial, trazendo aspectos do paradigma de objetos para essa linguagem.

O conceito de OO surgiu da ideia da simulação de eventos e da troca de mensagens na criação de uma nova linguagem de programação que fornecesse esses suportes, mais tarde outras denominações e fundamentos foram definidos como encapsulamento, herança e classes.

**Invocação Implícita, Baseada em Evento**

Surgiu na década de 1970, quando os sistemas de tempo compartilhado começaram a se tornar populares. Naquele momento, os programadores estavam procurando uma maneira de lidar com a complexidade crescente de seus programas. Eles descobriram que poderiam melhorar a legibilidade e a manutenibilidade de seus códigos se o dividissem em partes menores que pudessem ser ativadas em resposta a eventos específicos. Essa ideia foi levada adiante com o surgimento da programação orientada a objetos na década de 1980, onde a invocação implícita, baseada em evento, tornou-se uma das principais abordagens para lidar com a complexidade em sistemas baseados em objetos.

**Sistemas em Camadas**

Arquitetura em camadas é um estilo arquitetural que dentre vários objetivos, destaca-se o de **organizar as responsabilidades** de partes de um software, normalmente criando um isolamento e dando um propósito bem definido a cada camada de forma que a mesma possa ser **reutilizável** por um nível mais alto ou até substituível.

Mas na prática, principalmente em aplicações corporativas, essas características se perdem em meio a códigos desorganizados e entendimento confuso sobre tais conceitos. Desta forma o tema resgata inicialmente um resumo da história e evolução do uso de camadas.

O uso do conceito de camadas é algo antigo, amplamente utilizado antes mesmo de se tornar popular na engenharia de software. Dentre exemplos conhecidos, pode-se citar:

* Modelo OSI, criado ainda na década de 70 e que na década de 80 virou o modelo de rede padrão para protocolos de comunicação.
* Sistemas operacionais: O uso de camadas por sistemas operacionais permitiu a padronização e evolução com relação a construção e uso de CPU, memória, dispositivos, kernel e aplicações.

## Justificativa da escolha da arquitetura

Conforme as ferramentas que serão utilizadas para a criação do sistema e o conhecimento sobre programação, foram escolhidas essas arquiteturas, pois as mesmas se encaixam perfeitamente no modelo de projeto que iremos desenvolver, e também, vale ressaltar, que haverá no desenvolvimento do projeto, algumas partes que se caracterizam com outros tipos de arquitetura, como por exemplo, será usado um banco de dados central para o armazenamento dos dados do sistema e projetos que serão criados no mesmo.

# Framework ágil

## Framework

O framework utilizado para o desenvolvimento do projeto/software é uma mesclagem entre o framework Extreme Programming (XP) e SCRUM. Foi utilizado a mesclagem dessas metodologias devido ao desenvolvimento do projeto, em que o mesmo deve ser desenvolvido utilizando o paradigma de orientação a objetos, sendo que o projeto pode ser ajustado conforme o desenvolvimento do software (ideia principal do framework XP); é usado uma abordagem iterativa e incremental para aperfeiçoar o controle de riscos do projeto, onde cada entrega é considerada uma sprint, em que cada sprint consiste na definição de tarefas que deverão ser cumpridas e entregues em um certo período, para o cumprimento do prazo do projeto, e satisfação dos resultados com os clientes.

## Documentação dos entregáveis/sprints

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprints - Modelos, Métodos e Técnicas de Engenharia de Software** | | | | | | | | | | | |
| Nº da Entrega | | | | | | | | **01** | | | |
| Data | | | | Inicio | | | | 16.03.2023 | | | |
| Término | | | | 24.03.2023 | | | |
| Conteúdo da Sprint | | Definição do grupo de trabalho | | | | | | | | | |
| Definição da ODS que será utilizado como tema do sistema | | | | | | | | | |
| Definição dos projetos que serão utilizados para atender a ODS | | | | | | | | | |
| Modelo de processo de desenvolvimento do software | | | | | | | | | |
|  |  | |  | |  |  |  | |  |  |
| Nº da Entrega | | | | | | | | **02** | | | |
| Data | | | | Inicio | | | | 30.03.2023 | | | |
| Término | | | | 07.04.2023 | | | |
| Conteúdo da Sprint | | Criar o relatório de viabilidade do sistema computacional | | | | | | | | | |
| Documentar o levantamento dos requisitos | | | | | | | | | |
| Realizar a análise e negociação de requisitos | | | | | | | | | |
|  |  | |  | |  |  |  | |  |  |
| Nº da Entrega | | | | | | | | **03** | | | |
| Data | | | | Inicio | | | | 30.03.2023 | | | |
| Término | | | | 14.04.2023 | | | |
| Conteúdo da Sprint | | Especificar os requisitos do sistema computacional | | | | | | | | | |
| Documentar a validação dos requisitos | | | | | | | | | |
|  |  | |  | |  |  |  | |  |  |
| Nº da Entrega | | | | | | | | **04** | | | |
| Data | | | | Inicio | | | | 06.04.2023 | | | |
| Término | | | | 28.04.2023 | | | |
| Conteúdo da Sprint | | Escolher os tipos de arquiteturas de software | | | | | | | | | |
| Realizar embasamento histórico sobre as arquiteturas | | | | | | | | | |
| Justificar a escolha das arquiteturas | | | | | | | | | |
|  |  | |  | |  |  |  | |  |  |
| Nº da Entrega | | | | | | | | **05** | | | |
| Data | | | | Inicio | | | | 18.05.2023 | | | |
| Término | | | | 26.05.2023 | | | |
| Conteúdo da Sprint | | Escolha do framework utilizado no software | | | | | | | | | |
| Documentação dos entregáveis/sprints | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprints - Gestão e Qualidade de Software** | | | | | | | | | | |
| Nº da Entrega | | | | | | | | **01** | | |
| Data | | | | Inicio | | | | 13.03.2023 | | |
| Término | | | | 21.03.2023 | | |
| Conteúdo da Sprint | | Definição do grupo de trabalho | | | | | | | | |
| Definição da ODS que será utilizado como tema do sistema | | | | | | | | |
| Definição dos projetos que serão utilizados para atender a ODS | | | | | | | | |
|  |  | |  | |  |  |  | |  |  |
| Nº da Entrega | | | | | | | | **02** | | |
| Data | | | | Inicio | | | | 20.03.2023 | | |
| Término | | | | 28.03.2023 | | |
| Conteúdo da Sprint | | Criação da EAP - Estrutura Analítica do Projeto proposto | | | | | | | | |
| Definir prazos do projeto proposto | | | | | | | | |
| Definir custos do projeto proposto | | | | | | | | |
|  |  | |  | |  |  |  | |  |  |
| Nº da Entrega | | | | | | | | **03** | | |
| Data | | | | Inicio | | | | 27.03.2023 | | |
| Término | | | | 04.04.2023 | | |
| Conteúdo da Sprint | | Criar o Tempo de abertura do projeto proposto (TAP) | | | | | | | | |
| Planificar a Integração Contínua | | | | | | | | |
| Escolher as técnicas de Qualidade que serão aplicadas | | | | | | | | |
|  |  | |  | |  |  |  | |  |  |
| Nº da Entrega | | | | | | | | **04** | | |
| Data | | | | Inicio | | | | 24.03.2023 | | |
| Término | | | | 09.04.2023 | | |
| Conteúdo da Sprint | | Realizar a análise por pontos de função do software | | | | | | | | |
| Valorizar o sistema computacional | | | | | | | | |

# 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, é possível 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). Modelos, métodos e técnicas da engenharia 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**](https://brasil.un.org/pt-br/sdgs/3).

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