Especificações de requisito de Software

para

Marca-passo de coração

**Versão 1.0 aprovada**

**Preparado por José Paulo Corrêa Ramos**

**UDESC**

**06/09/2019**

**Table of Contents**

**Tabela de conteúdos [ii](#_gjdgxs)**

**Revisão [ii](#_30j0zll)**

**1. Introdução [1](#_1fob9te)**

1.1 Propósito.…………………………………………………………………………………………..[1](#_3znysh7)

1.2 Document Conventions…..………………………………………………………………………...[1](#_2et92p0)

1.3 Intended Audience and Reading Suggestions……………………………………………………...[1](#_tyjcwt)

1.4 Product Scope……………………………………………………………………………………...[1](#_3dy6vkm)

1.5 References………………………………………………………………………………………….[1](#_1t3h5sf)

**2. Overall Description [2](#_4d34og8)**

2.1 Product Perspective……………………………………………………...………………………....[2](#_2s8eyo1)

2.2 Product Functions……………………………………………………...…………………………..[2](#_17dp8vu)

2.3 User Classes and Characteristics ……………………………...…………………………………...[2](#_3rdcrjn)

2.4 Operating Environment…………………………………………………………………………….[2](#_26in1rg)

2.5 Design and Implementation Constraints……………………………...…………………………...[2](#_lnxbz9)

2.6 User Documentation……………………………………………………………………………….[2](#_35nkun2)

2.7 Assumptions and Dependencies……………………………...…………………………………….[3](#_1ksv4uv)

**3. External Interface Requirements [3](#_44sinio)**

3.1 User Interfaces……………………………...……………………………………………………...3

3.2 Hardware Interfaces……………………………...………………………………………………...[3](#_z337ya)

3.3 Software Interfaces……………………………...………………………………………………….[3](#_3j2qqm3)

3.4 Communications Interfaces……………….……………...………………………………………...3

**4. System Features [4](#_4i7ojhp)**

4.1 System Feature 1……………………………...…………………………………………………....[4](#_2xcytpi)

4.2 System Feature 2…………….……………………………...……………………………………...[4](#_1ci93xb)

**5. Other Nonfunctional Requirements [4](#_3whwml4)**

5.1 Performance Requirements……………………………...………………………………………....[4](#_2bn6wsx)

5.2 Safety Requirements……………………………...………………………………………………..[5](#_qsh70q)

5.3 Security Requirements……………………………...……………………………………………...[5](#_3as4poj)

5.4 Software Quality Attributes……………………………...………………………………………...[5](#_1pxezwc)

5.5 Business Rules……………………………...……………………………………………………...[5](#_49x2ik5)

**6. Other Requirements [5](#_2p2csry)**

**Appendix A: Glossary [5](#_147n2zr)**

**Appendix B: Analysis Models [5](#_3o7alnk)**

**Appendix C: To Be Determined List [6](#_23ckvvd)**

**Histórico de revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introdução

## Propósito

Esse documento tem a intenção de apresentar os requisitos necessário que um sistema de marcapasso necessita para seu funcionamento adequado.

## Convenção de documentos

Para esse documento, foi utilizado um estudo de caso de um marcapasso, o qual informou quais os principais requisitos que o sistema necessita.

## Público alvo e sugestões de leitura

*Esse documento é direcionado para pessoas que irão montar um marca passo, assim como precisam saber de suas especificações em um nível mais técnico, já que ele irá descrever detalhes de como é o funcionamento dele. Também é recomendado para médicos e pacientes que utilizarão ele.*

*<Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>*

## Escopo do produto

Um marca-passos é um pequeno sistema eletrônico destinado a ajudar o coração a manter um ritmo regular. Será utilizado principalmente por pessoas que tem problemas com o ritmo do batimento, como a bradicardia e taquicardia. O marcapasso conseguirá manter o ritmo através da emissão de sinais elétricos, os quais farão o ritmo voltar ao normal.

## Referências

Para esse SRS foi utilizado um estudo de caso do professor Willian Castaneda, o qual pode ser encontrado no Moodle próprio da matéria dele, na plataforma em que o professor disponibiliza seus arquivos. O arquivo foi disponibilizado em Agosto de 2019.

# Descrição geral

## Perspectiva do produto

O marcapasso é dispositivo único, que tem a finalidade de fazer o coração funcionar adequadamente. Ele será sempre implantado próximo ao coração de uma pessoa, internamente, portanto ele sempre pertencerá a esse ambiente enquanto estiver em funcionamento.

Um diagrama dessa interação entre o marcapasso e o ambiente está no link a seguir.

https://drive.google.com/file/d/11GKa9Dq2awZ\_G4lNv\_wnXXGBN1NIomW3/view?usp=sharing

## Funcionalidades do produto

Tratar bradicardia, tratar taquicardia, ter um vários modos operacionais, para controlar situações de estresse em tempo real e estresse de causalidade

## Tipos de usuários e características.

A classe de usuários mais importante é que o marcapasso deve satisfazer é a das pessoas que terão o dispositivo implantado, ou seja, as pessoas que tem problemas no batimento do coração. Após eles, vem as pessoas que terão que implantar o dispositivo, pois precisa ser simples o suficiente para eles conseguirem realizar o processo.

## Ambiente operacional

Os elementos básicos do marcapasso são: Ligações, gerador de pulso, monitor do controlador do dispositivo e o acelerômetro.

## Restrições de projeto e implementação

O marcapasso não deverá ser muito grande, pois deve ser implantado internamente. O gerador de pulso terá uma capacidade máxima de quanta carga poderá guardar, assim como um regulador da intensidade dos pulsos.

## Manual do usuário

Estudo de caso do professor Willian Castaneda, o qual pode ser encontrado no Moodle próprio da matéria dele, na plataforma em que o professor disponibiliza seus arquivos. O arquivo foi disponibilizado em Agosto de 2019.

## Hipóteses e dependências

Para o marcapasso entrar em funcionalidade, ele precisa ser implantado cirurgicamente no paciente, e ter todas as suas peças em perfeito estado.

# External Interface Requirements

## User Interfaces

*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>*

## Hardware Interfaces

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

## Software Interfaces

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

## Communications Interfaces

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

# System Features

*<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>*

## System Feature 1

*<Don’t really say “System Feature 1.” State the feature name in just a few words.>*

4.1.1 Description and Priority

*<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>*

4.1.2 Stimulus/Response Sequences

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

4.1.3 Functional Requirements

*<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>*

*<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>*

REQ-1:

REQ-2:

## System Feature 2 (and so on)

# Other Nonfunctional Requirements

## Performance Requirements

*<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>*

## Safety Requirements

*<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>*

## Security Requirements

*<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>*

## Software Quality Attributes

*<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>*

## Business Rules

*<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>*

# Other Requirements

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

**Appendix A: Glossary**

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

**Appendix B: Analysis Models**

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

**Appendix C: To Be Determined List**

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*