**Histórico de Versões**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** | **Revisor** | **Aprovado por** |
| 00/00/0000 | 1.0 | Versão inicial | Natan Almeida de Lima e Maria Jakeline Freitas da Silva | Raphael de Carvalho Muniz |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Responsáveis**

**Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte**

**Idealizadores**

Rafael Ferreira da Silva

Raphael de Carvalho Muniz

Jeferson Queiroga Pereira

**Professores responsáveis**

Ciro Daniel Gurgel de Moura

Raphael de Carvalho Muniz

Jeferson Queiroga Pereira

**EPSR Corporation**

**Presidente**

Maria Jakeline Freitas da Silva

**CEO**

Natan Almeida de Lima

**Desenvolvedores**

Natan Almeida de Lima

Maria Jakeline Freitas da Silva

**Design**

Maria Jakeline Freitas da Silva

**Documento de visão do sistema**

**1 - Objetivo**

O objetivo deste documento, é coletar, analisar e definir as principais necessidades e características que o sistema web *BestRoom* deve conter, para que os envolvidos tenham um ponto de referência sobre o que será desenvolvido.

O documento abstrai informações detalhadas e tem como foco elucidar as principais funcionalidades do sistema.

O documento deve ser atualizado no final do projeto.

**2 - Descrição do Produto**

O projeto consiste em um duas aplicações webs, sendo elas, uma API REST e uma Aplicação Web de interface que consome o serviço da API. O Webservice é responsável pelo gerenciamento dos dados e regras do negócio, funcionando de forma modular, e o website servindo como interface para o uso do sistema.

**3 - Envolvimento**

**3.1 Abrangência**

O sistema abrange as relações entre usuários (clientes) e estabelecimentos de hospedagens, facilitando o acesso a reservas de quartos.

**3.2 Papel dos Atores**

**3.2.1 Usuário**

|  |  |
| --- | --- |
| **Descrição** | **Usuário do sistema** |
| Papel | Se cadastrar no sistema, reservar quartos, visualizar e pagar reservas. |
| Insumos ao sistema | Cadastrado como Usuário |
| Representante | Qualquer pessoa que desejar usar o sistema. |

**4 - Funcionalidades**

|  |  |  |
| --- | --- | --- |
| **Necessidade 1** | **Cadastrar Usuários no Sistema** | **Tipo**  Crítica |
| **ID Funcionalidade** | **Descrição das Funcionalidades** | |
| **F 1.1** | Se cadastrar no sistema | |
| **Usuário** | |

|  |  |  |
| --- | --- | --- |
| **Necessidade 2** | **Reservar Quartos** | **Tipo**  Crítica |
| **ID Funcionalidade** | **Descrição das Funcionalidades** | |
| **F 2.1** | Escolher período de estadia e solicitar a reserva. | |
| **Usuário** | |

|  |  |  |
| --- | --- | --- |
| **Necessidade 3** | **Visualizar Reservas** | **Tipo**  Importante |
| **ID Funcionalidade** | **Descrição das funcionalidades** | |
| **F 1.1** | Visualizar informações de suas reservas já cadastradas. | |
| **Usuários** | |

|  |  |  |
| --- | --- | --- |
| **Necessidade 4** | **Pagar Reservas** | **Tipo**  Importante |
| **ID Funcionalidade** | **Descrição das funcionalidades** | |
| **F 1.1** | Pagar reservas cadastradas usando serviço externo. | |
| **Usuários** | |

**5 - Proposta de solução tecnológica**

O sistema do website será feito com HTML, CSS, JavaScript ( + Jquery) e PHP com o framework Laravel. Já o servidor será inteiramente em PHP (também com Laravel) e com persistência de dados em banco de dados Postgresql. Para hospedagem do banco de dados, será utilizado o serviço de hospedagem disponível no site <https://www.heroku.com>.

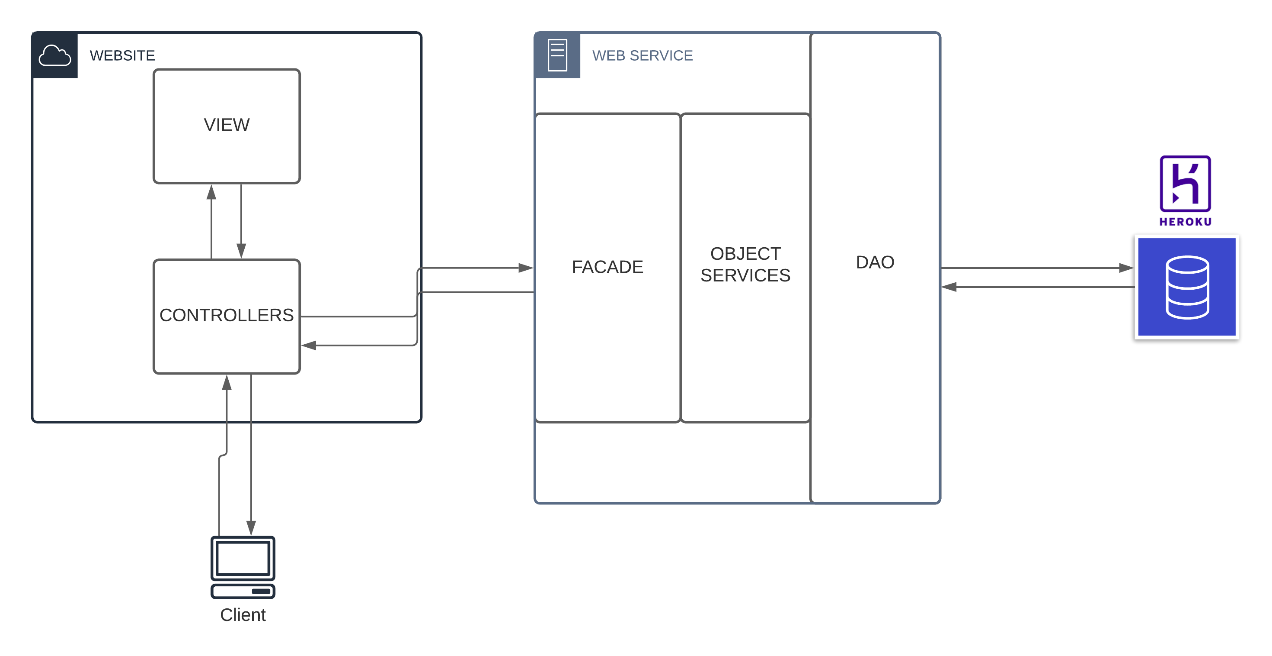
**6 – Arquitetura de software**

As arquiteturas definidas para o sistema são as seguintes:

Para o website é utilizada a arquitetura MVC (Model – View – Controller), sendo que a parte referente ao *Model* é abstraída e é utilizado o serviço de hospedagem como camada de dados.

Já o sistema do serviço de hospedagem é dividido nas seguintes camadas: *Facade*; *Object Services;* e *DAO (Data Access Object)* como mostra a figura a seguir*.*

Figura 1 - Representação arquitetural do sistema



A arquitetura utilizada para o desenvolvimento da aplicação web simplifica e organiza a distribuição das funções do sistema, utilizando uma camada para a parte visual gráfica (View – em português, Visão), uma para o gerenciamento dos dados (Model – em português, Modelo) e uma para o gerenciamento da interligação dos dados com a *apresentação*, responsável pelas chamadas de acesso aos dados, e seu envio para a View, essa chamada de *Controller* (Controlador).

* O Serviço de hospedagens por sua vez, organiza-se em 3 partes ou camadas sequenciais:
* A Fachada (Facade), responsável pela verificação dos dados chegados, acesso e chamada aos serviços e retornos dos dados;
* DAO (*Data Access Object)* – Responsável pelo acesso irrestrito ao banco de dados (ler, escrever, editar e excluir).
* A chamada de Serviços (*Object Services)*, responsável pelas *regras do negócio*, suas restrições e chamadas à camada DAO.

Com a divisão dos sistemas em camadas, as quais tenham suas responsabilidades bem definidas, o processo de desenvolvimento se torna mais organizado, facilitando a identificação de erros e expandindo a escalabilidade do sistema.