



INSTITUTO FEDERAL

Rio Grande do Norte
Campus EaD

Instituto Federal de Educação, Ciência e Tecnologia Rio Grande do Norte
Curso Técnico em Informática para Internet

Desenvolvimento de um sistema web para controle de veículos e acesso aos estacionamentos dos campi do IFRN

José Victor Gomes de França

Natal-RN, Novembro de 2019

José Victor Gomes de França

Desenvolvimento de um sistema web para controle de veículos e acesso aos estacionamentos dos campi do IFRN

Trabalho de Conclusão de Curso Monografia apresentada ao Instituto Federal de Educação, Ciência e Tecnologia Rio Grande do Norte (IFRN), como pré-requisito para a obtenção do título de Técnico em Informática para Internet.

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

Orientador: Prof. Rafael Fernandes de Queiroz

Natal-RN

Novembro de 2019

José Victor Gomes de França

Desenvolvimento de um sistema web para controle de veículos e acesso aos estacionamentos dos campi do IFRN

Trabalho de Conclusão de Curso Monografia apresentada ao Instituto Federal de Educação, Ciência e Tecnologia Rio Grande do Norte (IFRN), como pré-requisito para a obtenção do título de Técnico em Informática para Internet.

Trabalho aprovado. Natal-RN, 01 de janeiro de 2014:

Prof. Rafael Fernandes de Queiroz
Orientador

Professor
Convidado 1

Professor
Convidado 2

Professor
Convidado 3

Natal-RN
Novembro de 2019

Dedico esse trabalho aos meus queridos pais, minha triunfante e amada Mãe, Paula Frassinete Gomes França, e meu batalhador pai, José Vital de França. A eles dedico todo meu reconhecimento, gratidão e amor. Igualmente dedico esse trabalho as minhas igualmente amadas irmãs , Maria José Gomes de França, Ana Maria Gomes de França e Maria das Dores Gomes de França por todo apoio oferecido ao longo de minha vida. Desejo poder ter sido merecedor do esforço dedicado por vocês em todos os aspectos, especialmente quanto à minha formação. Também dedico este trabalho ao Leonardo Maximino Bernardo pelo companheirismo e felicidade proporcionados.

Agradecimentos

Ao longo destes dois anos de estudo que resultaram neste trabalho de conclusão de curso, pessoas e instituições me ajudaram, ensinando e apoiando. Agora que alcanço meus objetivos, não poderia deixar de reconhecê-las.

Começo por agradecer a minha adorável e digníssima mãe, pilar seguro, colo de amor incondicional, Paula Frassinete Gomes França e ao meu amado e dedicado pai, José Vital de França. Desejo que esta minha conquista sirva de algum modo, se possível, como exemplo para minhas irmãs e irmãos.

Agradeço ao meu orientador, Prof. Rafael Fernandes de Queiroz, por ser meu guia nesta árdua tarefa de desenvolver esse sistema.

Meu reconhecimento aos professores, colegas e funcionários do curso de informática para internet EaD do IFRN.

“O que não provoca minha morte faz com que eu fique mais forte.”
(Friedrich Nietzsche)

Resumo

(Em construção:) Tendo em vista que a quantidade de vagas no estacionamento do IFRN é limitado e a quantidade de veículos ocupantes está aumentando gradativamente, é necessário controlar os veículos que realmente devem ter acesso ao estacionamento do IFRN, tais como aqueles pertencentes a servidores do instituto. Além da quantidade, o processo para cadastro de um veículo é bastante burocrático, exigindo a comunicação entre várias pessoas (Diretor, Departamento de Segurança, etc.) para autorizar o cadastro do veículo, dificultando a aquisição do adesivo que identifica um veículo autorizado. O sistema deve ser desenvolvido utilizando o framework Ruby On Rails e banco de dados PostgreSQL, utilizando conhecimentos de diversas disciplinas do curso (APOO, PDSI, Banco de Dados, e outras).

Palavras-chaves: latex. abntex. editoração de texto.

Abstract

Given that the number of IFRN parking spaces is limited and the number of occupant vehicles is gradually increasing, it is necessary to control the vehicles that really should have access to the IFRN parking, such as those belonging to IFRN servants. In addition to the amount, the process for registering a vehicle is quite bureaucratic, requiring communication between several people (Director, Security Department, etc.) to authorize the registration of the vehicle, making it difficult to purchase the sticker that identifies an authorized vehicle. The system should be developed using the Ruby On Rails framework and PostgreSQL database, using knowledge of several course subjects (APOO, PDSI, Database, and others).

Keywords: latex. abntex. text editoration.

Lista de ilustrações

Figura 1 – Estrutura MVC do RoR.	9
Figura 2 – Diagrama de Caso de Uso	12
Figura 3 – Diagrama de Classes	14
Figura 4 – Tela de Cadastramento do Usuário.	15
Figura 5 – Tela de Confirmação de Cadastramento do Usuário.	15
Figura 6 – Tela de Confirmação de Cadastramento de Funcionário.	15
Figura 7 – Tela de Confirmação de Cadastramento de Estacionamento.	16
Figura 8 – Tela de Confirmação de Cadastramento de Vaga.	16
Figura 9 – Tela de Confirmação de Cadastramento de Veículo.	16

Lista de tabelas

Lista de abreviaturas e siglas

ABNT	Associação Brasileira de Normas Técnicas
abnTeX	ABsurdas Normas para TeX

Lista de símbolos

Γ	Letra grega Gama
Λ	Lambda
ζ	Letra grega minúscula zeta
\in	Pertence

Sumário

I	PREPARAÇÃO DA PESQUISA	1
1	INTRODUÇÃO	3
1.1	Tema	4
1.2	Objetivo Geral	4
1.3	Objetivos Específicos	4
1.4	Delimitação do Problema	4
1.5	Motivações e Justificativas	4
1.6	Método de Trabalho	5
1.7	Organização do Trabalho	5
2	ESTADO DA ARTE	7
2.1	Desenvolvimento web	7
2.2	Descrição da arquitetura Model-View-Controller	8
2.3	Banco de Dados	10
2.4	Requisitos do Sistema	10
2.4.1	Atores	10
2.4.2	Requisitos Funcionais	10
2.4.3	Requisitos Não-Funcionais	11
2.5	Diagrama de Casos de Uso	11
2.6	Relacionamentos no RoR	12
2.6.1	Tipos de relacionamentos	12
2.6.2	Associação belongs_to	13
2.6.3	Associação has_one	13
2.6.4	Associação has_many	13
2.6.5	Associação has_and_belongs_to_many	13
2.7	Diagrama de Classes e seus Relacionamento	13
2.8	Testes do Sistema	13
3	MATERIAIS E MÉTODOS	17
3.1	Considerações Finais	17
II	PROPOSTA	19
4	SISTEMA PROPOSTO	21
4.1	Primeira Parte do Sistema Proposto	21
4.2	Considerações Finais	21

III	PARTE FINAL	23
5	RESULTADOS E DISCUSSÃO	25
5.1	Base de Dados	25
5.2	Considerações Finais	25
	Conclusão e Trabalhos Futuros	27
	REFERÊNCIAS	29
	APÊNDICES	31
	APÊNDICE A – PRIMEIRO APÊNCICE	33
	APÊNDICE B – PERCEBA QUE O TEXTO DO TÍTULO DESSE SEGUNDO APÊNDICE É BEM GRANDE	35
	ANEXOS	37
	ANEXO A – NOME DO PRIMEIRO ANEXO	39
	ANEXO B – NOME DE OUTRO ANEXO	41

Parte I

Preparação da pesquisa

1 INTRODUÇÃO

Quando a demanda por vagas de estacionamento aumentam demasiadamente, inevitavelmente se estabelece um reconhecido problema de congestionamento. Atualmente, é comum situações em que, por exemplo, nos campus universitários e regiões próximas aos centros das grandes cidades, a capacidade dos estacionamentos se esgotem facilmente. Geralmente dimensiona-se o número de vagas alocadas para estacionamentos no plano diretor das instituições públicas ou privadas. Estas vagas podem ser, à princípio, suficientes para atender a demanda de estacionamento dos veículos dos professores, funcionários, estudantes e visitantes no anos iniciais. Porém, a previsão de crescimento do número de usuários, pode ser, por algum motivo, subestimada. Nas situações em que a demanda pelo estacionamento continua a crescer, o espaço reservado para acomodação dos veículos torna-se limitado. Outra situação a ser considerada é a de que alguma área livre do campus, geralmente seja utilizada na construção de novas salas de aula, escritórios e laboratórios, ou seja, as infraestruturas que são consideradas mais importantes. Em vista destas situações, recorrem-se às estratégias de gestão e a Sistemas de Transporte Inteligentes (Intelligent Transportation Systems, ITS) para lidar com o problema de congestionamento do estacionamento, em vez da capacidade de expansão. Pode-se argumentar que o acesso ao estacionamento é um fator que determina se e como os usuários chegam aos campus das universidades e institutos federais. Normalmente, nestes estabelecimentos, a maioria dos usuários possuem seu próprio horário de trabalho ou estudo e a demanda de estacionamento é relativamente inelástica. Algumas alternativas para mitigar o problema do congestionamento nos estacionamentos podem ser antipáticas aos usuários, dentre estas, pode-se citar a administração do estacionamento por meio de uma combinação de cobrança pelo direito de estacionar ou fornecer modos de transporte alternativos. Outra abordagem admitida para resolver o problema de congestionamento do estacionamento do campus é resolver o problema pelo lado da oferta; isto é, implementar políticas para melhor controle do uso do estacionamento e usar o ITS para tornar o uso do estacionamento mais eficiente. A dificuldade de encontrar uma vaga de estacionamento não apenas contribui para o congestionamento do tráfego dentro da repartição ou estabelecimento, mas também nas ruas circundantes, ao procurar uma vaga vazia para estacionar, os veículos em circulação utilizam a marcha lenta e podem aumentar as emissões de gases nocivos resultantes da queima de combustível, afetando negativamente a saúde da comunidade e prejudicando o meio ambiente. Deve-se aumentar a conscientização dos tomadores de decisão das instituições públicas e privadas sobre a importância de evitar o congestionamento dos estacionamentos e aumentar a compreensão dos impactos dos estacionamentos no meio ambiente e na saúde da comunidade. O ideal a ser alcançado deve, preferencialmente, contar com projetos de estacionamentos que

consigam evitar seu congestionamento.

1.1 Tema

Alguns campus do IFRN, notadamente, o de Natal Central, vêm enfrentando situações semelhantes as descritas anteriormente. No intuito de organizar e registrar o ingresso de veículos as suas dependências, este trabalho de conclusão de curso propõe o desenvolvimento de um sistema web para controle dos veículos e acesso ao estacionamento do Instituto Federal de Educação, Ciência e Tecnologia Rio Grande do Norte - IFRN, visando controlar os veículos que podem ter acesso ao estacionamento da instituição.

1.2 Objetivo Geral

A proposta deste trabalho de conclusão de curso consiste em desenvolver um sistema web administrativo completo, com autenticação, para controle de veículos e acesso aos estacionamentos dos campi do IFRN.

1.3 Objetivos Específicos

- Utilizar a estrutura Ruby on Rails (RoR) ([HARTL; FERNANDEZ, 2011](#)) para desenvolvimento do sistema web.
- Utilizar o banco de dados PostgreSQL.
- Apresentar um caso de uso com o sistema elaborado.

1.4 Delimitação do Problema

Apresentar um sistema web administrativo completo, com autenticação, para controle de veículos e acesso aos estacionamentos dos campi do IFRN.

1.5 Motivações e Justificativas

Tendo em vista que a quantidade de vagas no estacionamento do campus Natal Central do IFRN é limitado e que a quantidade de veículos ocupantes está aumentando gradativamente, é necessário controlar o acesso dos veículos que realmente possuem prerrogativa para sua utilização, tais como aqueles pertencentes a servidores do instituto e visitantes. Além da quantidade, o processo para cadastro de um funcionário, que possua um veículo, é bastante burocrático, exigindo a comunicação entre um certo número

de funcionários, por exemplo, diretor e departamento de segurança, para autorização do registro, dificultando a aquisição do adesivo que identifica um veículo habilitado. A abordagem de uma solução para o problema de congestionamento e dificuldades para organização e liberação de autorizações, culmina na necessidade do desenvolvimento de um sistema que gerencie esse, e, possivelmente, outros estacionamentos. Esta é uma grande oportunidade de aplicar os conceitos apresentados nas diversas disciplinas estudadas no curso técnico de informática para internet, ofertado pelo IFRN, na modalidade de ensino à distância. Particularmente, no presente trabalho, optou-se por desenvolver o referido sistema utilizando a(o) estrutura/framework Ruby On Rails (RoR) e banco de dados PostgreSQL. Nesse sentido, o desafio que se apresentou é de certa maneira complexo e instigante, pois tratam-se de ferramentas elaboradas e que exigem considerável nível de conhecimento para sua utilização adequada.

1.6 Método de Trabalho

O presente trabalho de conclusão de curso utilizou as funcionalidades do Rails Admin, um mecanismo, disponibilizado na internet, que fornece uma interface fácil de usar para gerenciamento de dados (SCORZA, 2018). Adicionalmente, utilizou-se como Sistema de Gerenciamento de Banco de Dados (SGBD), o PostgreSQL. Identificou-se como uma necessidade muito importante para a comunidade acadêmica do IFRN, propor soluções para o problema de desorganização do acesso dos usuários aos estacionamentos da instituição. Dessa forma, o presente trabalho propõe um meio de gerenciar de maneira simples os estacionamentos da instituição. Inicialmente, analisou-se as causas mais prováveis da má utilização dos estacionamentos. Pode-se citar, por exemplo, o acesso de pessoas não autorizadas, que não possuem vínculo com o IFRN.

Concomitantemente, estudou-se mais aprofundadamente a linguagem de programação Ruby, e a utilização da estrutura Ruby on Rails. Com essas ferramentas principais implementou-se um sistema administrativo em conjunto com a base de dados. Além disso foram empregados conhecimentos em HTML, CSS e Javascript. Para a criação da plataforma que irá gerenciar o estacionamento de veículos, foi utilizada **gems** específicas, que simplificam a criação de um sistema administrativo. Dessa forma, criou-se os seguintes modelos: **estacionamento**, **veiculo**, **funcionario** e **vaga**. Os parâmetros de cada modelo, bem como suas relações e as **gems** utilizadas, serão descritas mais à frente.

1.7 Organização do Trabalho

Em sua organização, o presente trabalho possui capítulos apresentados da seguinte maneira:

- Desenvolvimento:

No segundo capítulo são apresentadas as ferramentas tecnológicas empregadas na construção do software e os requisitos funcionais e não-funcionais. Nessa oportunidade descrevem-se em detalhes cada aplicação realizada. Mostram-se, adicionalmente, a maneira de integração com o banco de dados, diagramas para o entendimento do sistema criado, as categorias de usuários e as atribuições que esses tem acesso.

- Avaliação:

No terceiro capítulo, caracteriza-se a verificação do sistema e apresentam-se os resultados de seu funcionamento.

- Conclusão:

No quarto e último capítulo, argumenta-se conclusivamente a respeito do software construído. Igualmente, pondera-se sobre possíveis melhorias em trabalhos futuros.

2 Estado da Arte

No presente capítulo serão apontados os fundamentos teóricos nos quais o desenvolvimento do sistema web de gerenciamento foram embasados, ou seja, como se constituem os padrões de organização do RoR e o comportamento de aplicações web. Adicionalmente, mostra-se uma descrição cuidadosa e sucinta de trabalhos relacionados à área do sistema desenvolvido, que mostram um recorte adequado para visualização do universo de estudo. Em seguida, apresentam-se os atores, isso é, os usuários finais do sistema. Como mencionado anteriormente, o software é direcionado à comunidade acadêmica do IFRN. Também são apresentadas, detalhadamente, todas as funcionalidades e delimitações do software. Na última seção, serão mostrados o diagrama de casos e uso. Uma perspectiva comparativa será realizada, relacionando as possibilidades possíveis de cadastramento .

2.1 Desenvolvimento web

Desenvolvimento web é o termo utilizado para descrever a criação de sites, na Internet ou numa intranet. Atualmente, alguns sites são aplicativos complexos capazes de realizar transações, apresentar dados em tempo real, e fornecer experiências interativas ao usuário. Os software baseados na web estão se tornando tão poderosos e importante quanto software de mesa. A grande maioria dos aplicativos da web produzidos fornecem funcionalidades avançadas, e isso é uma tarefa complexa que envolve um considerável número de conjuntos de ferramentas, e muitas opções. As estruturas da web criadas com tais tecnologias necessitam de um gerenciamento de conteúdo. Para esse objetivo, costuma-se utilizar ferramentas que construam uma estrutura web, permitindo escolher entre diferentes tipos de elementos que melhor atendam as necessidades ([HACKER, 2008](#)). Em sua investigação, ([PLEKHANOVA, 2009](#)), avaliou três estruturas de desenvolvimento web principais de código fonte abertas: Django, Ruby on Rails (RoR) e CakePHP, escritos em três linguagens diferentes - Python, Ruby e PHP, respectivamente. Todas as três estruturas citadas têm arquiteturas semelhantes e reivindicam ter características parecidas, como produtividade aprimorada e reutilização de código. Em sua conclusão, a autora classificou as ferramentas segundo critérios que verificaram:

- Os componentes front-end e back-end (desenvolvimento da interface do usuário, gerenciamento e migração de dados).
- O desempenho total das estruturas (manutenibilidade, testabilidade).
- A popularidade, difusão na comunidade, maturidade e comercialização.

A conclusão geral dessa autora foi de que todas as três estruturas são ferramentas robustas e a escolha qual utilizar precisa considerar o contexto de aplicação. As empresas e os profissionais de desenvolvimento web necessitam individualizar o processo avaliativo de escolha ao seu contexto único.

(LEI; MA; TAN, 2014), realizou uma investigação concentrando-se no impacto do desempenho de três diferentes tecnologias da Web: Node.js, PHP e Python-Web. No entanto, negligenciou os problemas de segurança e escalabilidade. Nessa perspectiva, desenvolveu um método universal de avaliação da técnica de desenvolvimento web com base na comparação do desempenho das ferramentas analisadas. Em sua conclusão verificou que o Node.js tem um desempenho muito melhor que o tradicional PHP em situação de alta simultaneidade e que o Python-Web também não é adequado para o uso intensivo em computação.

Em síntese, pode-se argumentar que o gerenciamento de conteúdo da web é o resultado da entrega de material para a web. O crescimento do número de páginas da web tornou-se popular nas últimas décadas e os padrões Model-View-Controller (MVC) facilitaram bastante seu desenvolvimento (MCKEEVER, 2003). Eles ocultam a complexidade, dão estrutura e consistência e promovem as melhores práticas.

A partir de sua experiência, (TAPIADOR; SALVACHÚA, 2012) explora a criação de sistemas de gerenciamento de conteúdo da web com Ruby on Rails, uma estrutura popular de desenvolvimento web produzida para aumentar a produtividade. Em sua investigação o autor explica as vantagens da implementação da arquitetura MVC baseando-se nos princípios de “convention over configuration” (convenção sobre configuração) e “don’t repeat yourself” (não se repita) (BÄCHLE; KIRCHBERG, 2007).

Diante dessa realidade, no presente trabalho de conclusão de curso, optou-se, como componente orientador do processo de desenvolvimento do sistema de gerenciamento de estacionamentos, aderir à doutrina Rails (TAYLOR, 2010), uma vez que partindo desta ferramenta é possível estabelecer as consistências e as principais dependências do sistema (ASTELS; MILLER; NOVAK, 2002).

2.2 Descrição da arquitetura Model-View-Controller

O Rails possui diversas características interessantes. Dessa propriedades, pode-se destacar algumas considerações bastante sérias, por exemplo as restrições sobre como se estrutura a construção dos aplicativos da web. Surpreendentemente, essas restrições, na realidade, facilitam a criação dos aplicativos. Elaborado em 1979 por Trygve Reenskaug, essa forma de organização é aplicada no desenvolvimento de funcionalidades interativas, dividindo-se em modelos, visões e controladores (RUBY; THOMAS et al., 2009). O modelo é responsável por manter o estado do aplicativo. As vezes esse estado é transitório, durando

apenas algumas interações com o usuário. Às vezes, o estado é permanente e é armazenado fora do aplicativo, frequentemente em um banco de dados.

A visualização é responsável por gerar uma interface com o usuário, normalmente baseados em dados do modelo. Embora a visualização possa apresentar ao usuário várias maneiras de inserir dados, a própria exibição nunca lida com dados recebidos. O trabalho da visualização é concluído depois que os dados são exibidos. Os controladores recebem eventos do mundo exterior (normalmente, a entrada de utilizador), interagem com o modelo, e apresentam uma perspectiva apropriada para o usuário. Na atualidade, o padrão de organização MVC prevalece como aspecto de criação dos frameworks.

Em geral, os aplicativos da web guardam suas informações em um banco de dados relacional. Os sistemas de entrada de solicitações armazenam informações e detalhes do cliente em tabelas de banco de dados. As consultas ao banco de dados relacionais são fundamentadas matematicamente na teoria de conjuntos. Porém, existe uma certa dificuldade em combinar bancos de dados relacionais com linguagens de programação orientadas a objetos (OO). Os objetos lidam com dados e operações, e os bancos de dados são todos sobre conjuntos de valores. Muitas vezes, torna-se complicado expressar termos relacionais codificando-os em um sistema OO. O contrário também é verdade.

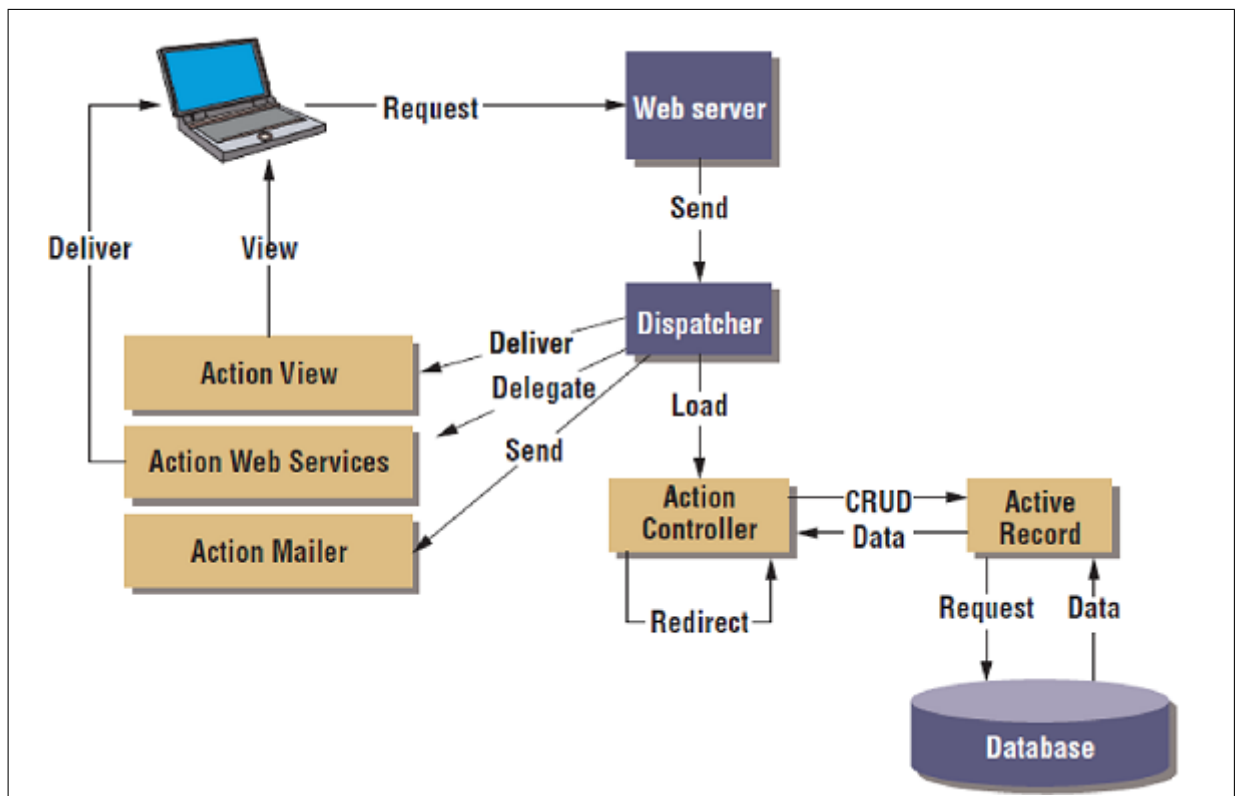


Figura 1 – Estrutura MVC do RoR.

2.3 Banco de Dados

Para o desenvolvimento desse projeto, implementou-se um banco de dados PostgreSQL, para armazenamento de informações sobre os usuários. A construção do Sistema de Gerenciamento de Banco de Dados do PostgreSQL foi iniciada em 1986. Os conceitos iniciais foram apresentados em (??). A primeira demonstração do sistema em operação foi realizada em 1987 e o modelo de dados inicial apareceu em (??). Na seção seguinte serão apresentados os atores, requisitos funcionais e não funcionais, o diagrama de caso de uso e informações sobre o banco de dados utilizado.

2.4 Requisitos do Sistema

2.4.1 Atores

Os atores que poderão acessar o sistema são os seguintes:

Porteiro: É o funcionário do IFRN responsável por realizar o cadastramento dos demais servidores ou alunos.

Administrador: É o funcionário do setor de TI responsável pela administração da plataforma. Esse ator tem as permissões necessárias para gerenciar a plataforma, realizando atualizações e melhorias, se necessárias.

2.4.2 Requisitos Funcionais

Na descrição desse tópico serão descritas as aplicações dos usuários definidos anteriormente. As ações dos atores estão relacionadas aos requisitos estabelecidos pelo sistema (SOMMERVILLE, 2011), (PRESSMAN, 2016).

1º Requisito Funcional

Cadastro de Usuários: No software desenvolvido é utilizada a *gem* 'devise' para cadastrar novos usuários.

2º Requisito Funcional

Autenticação: A *gem* 'devise' proporciona ao software a geração de uma tela de login onde o usuário informa suas habilitações obtendo acesso à plataforma de gerenciamento.

3º Requisito Funcional

Redefinição de Senha: A *gem* 'devise' prepara a página de cadastramento de maneira que possa enviar um link para o e-mail do usuário para redefinição de senha, caso seja solicitada.

4º Requisito Funcional

Cadastramento do estacionamento: A *gem* 'Admin' permite a geração de modelos sem

a necessidade da criação de *Controller* ou rotas. Dessa maneira, gerou-se os demais modelos, a saber, modelos para o cadastramento de: **veículo**, **tipo de vaga** e **funcionário**.

5º Requisito Funcional

Edição dos cadastros: A *gem* 'Admin' prepara automaticamente a possibilidade de edição dos cadastros realizados. Nesse caso é possível atualizar, remover e exportar.

2.4.3 Requisitos Não-Funcionais

Define-se requisitos não-funcionais como aqueles que não possuem relacionamento direto com as atividades do software. Porém, são limitações dos serviços oferecidos pelo software (REZENDE, 2006).

1º Requisito Não-Funcional

Incompatibilidade com dispositivos móveis: O software não é compatível com dispositivos móveis.

2º Requisito Não-Funcional

Acessos Simultâneos: O banco de dados *postgresql* não possui limitação. No entanto, para proteção de acessos simultâneos, configurou-se um limite, restrito a uma faixa de IPs. Caso seja necessário uma maior capacidade, existe a possibilidade de modificação.

2.5 Diagrama de Casos de Uso

Levando-se em consideração as premissas anteriormente explicadas, pode-se modelar as funções do sistema proposto. Na Fig. 2 é apresentado o diagrama de casos de uso do sistema de gerenciamento de estacionamento. Idealizou-se que o ator *segurança* é o responsável pelo cadastramento, atualização e exclusão do ator *Funcionário_1*, que por sua vez, fornece seus dados pessoais e de seu veículo.

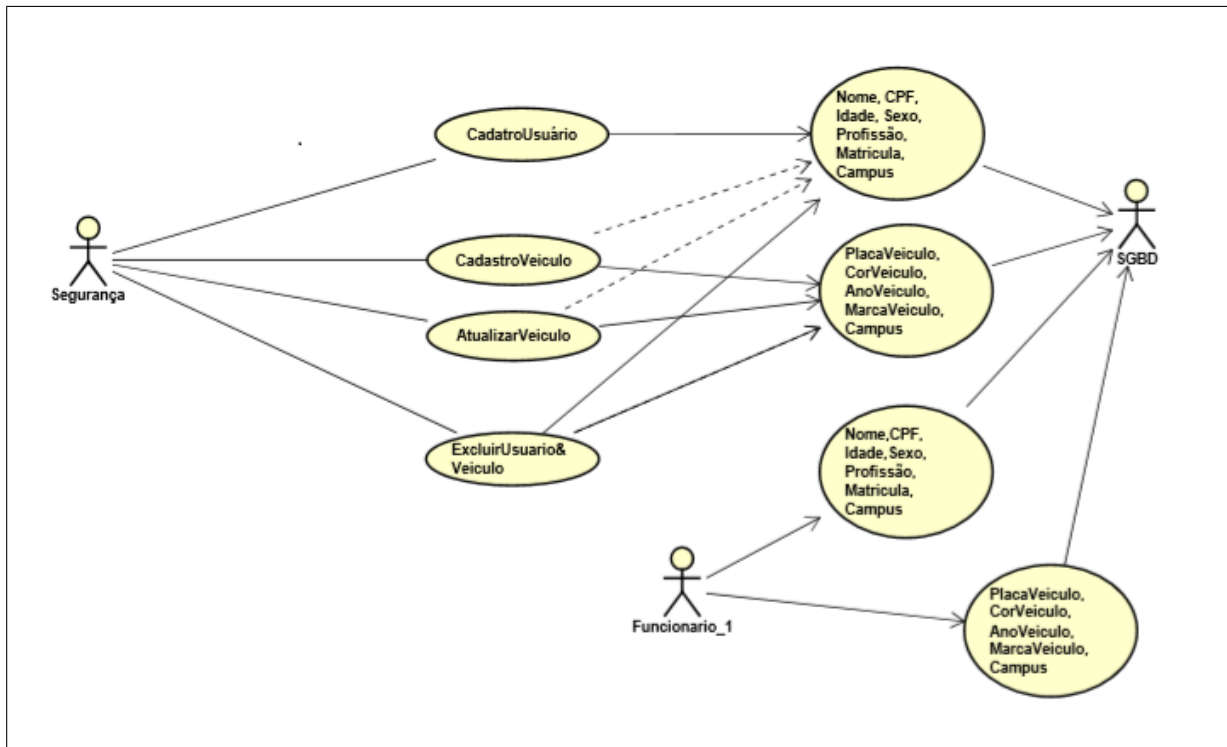


Figura 2 – Diagrama de Caso de Uso

2.6 Relacionamentos no RoR

Comumente, sistemas que empregam orientação a objetos utilizam classes que retratam elementos reais do domínio de negócio. O sistema de gerenciamento de estacionamento desenvolvido no presente trabalho possui as classes cujos modelos foram mencionados anteriormente (seção 2.4.2). Essas classes de entidades em geral possuem relacionamentos que devem ser formalizados em código, isso quer dizer que essas classes precisam conter as informações de como devem atuar entre si. O framework Rails possui recursos para essa atividade. O tópico seguinte mostra alguns dos tipos mais importantes de relacionamentos que podem existir entre as classes do modelo de uma aplicação.

2.6.1 Tipos de relacionamentos

Em (SILVA, 2015) se lê:

Existem diversas formas de dois objetos se associarem entre si, os relacionamentos mais comuns entre classes são:

- Um para um. Onde um objeto pode estar associado a apenas outro objeto de um determinado tipo.
- Um para muitos. Neste caso um objeto poderá se associar com um ou vários objetos de outra classe.
- Muitos para muitos. Um objeto pode participar de vários relacionamentos com outros objetos de determinado tipo.

A aplicação desses relacionamentos na estrutura Rails segue as seguintes formas: **belongs_to** (pertence a), **has_one** (tem um), **has_many** (tem muitos), **has_and_belongs_to_many** (tem e pertence a muitos), entre outras. Cada método possui recursos que trazem efeitos ou modificações no banco de dados. No próximo tópico essas características serão discutidas.

2.6.2 Associação **belongs_to**

Essa associação é geralmente denominada de *um para um* e é utilizada para mostrar que um certo modelo *pertence a* outro.

2.6.3 Associação **has_one**

Essa ligação é do tipo *um para um* e possui alguma semelhança com **belongs_to**. No entanto, preferivelmente, deve ser empregada em situações diferentes causando um efeito diferenciado. Tal associação é bidirecional a **belongs_to**. A diferença mais significativa é que não adiciona chave estrangeira ao modelo que a declara.

2.6.4 Associação **has_many**

Usa-se essa ligação entre modelos para indicar que o modelo em questão tem nenhum ou muitos elementos de outro modelo da aplicação. É denominada, frequentemente, de *um para muitos*. Semelhante a **has_one** ela não adiciona nenhuma coluna no modelo que a utiliza. Seu emprego deve ser em conjunto com **belong_to** para indicar por meio de chave estrangeira com quem o objeto filho se relaciona.

2.6.5 Associação **has_and_belongs_to_many**

Nessa opção é possível gerar uma ligação do tipo *muitos para muitos* entre dois modelos da aplicação.

2.7 Diagrama de Classes e seus Relacionamento

Num diagrama de classes UML, as associações possibilitam a descrição de informações nos extremos da linha que representa a ligação entre as classes. A Fig. 3 mostra os relacionamentos e os limites superiores (ou máximos) que representam a multiplicidade entre as classes.

2.8 Testes do Sistema

Nessa seção serão apresentadas as telas da plataforma de gerenciamento de estacionamentos seguidas de descrições das funcionalidades.

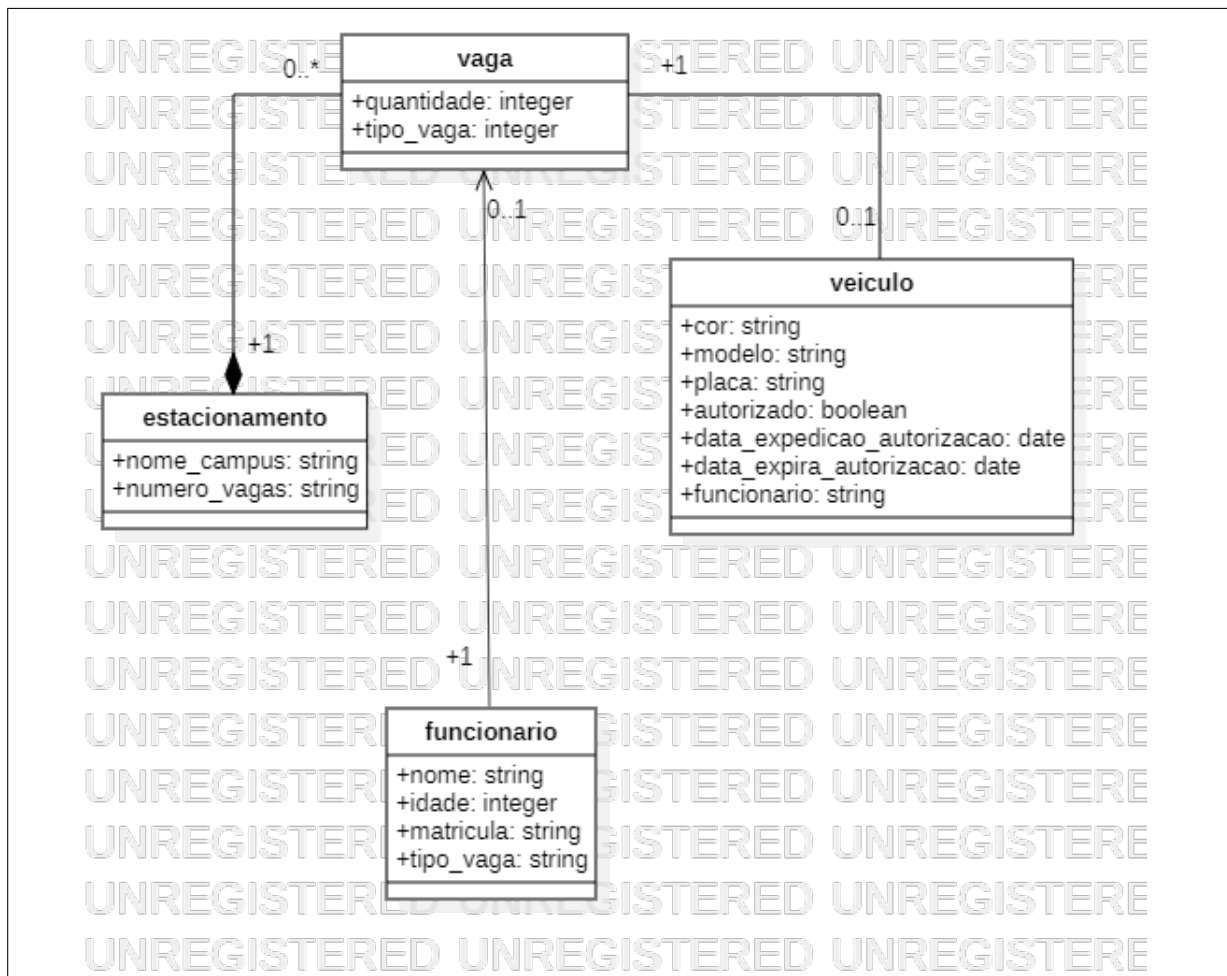


Figura 3 – Diagrama de Classes

Na tela mostrada na Fig. 4 o usuário insere as credenciais para ser admitido no sistema. A Fig. 5 mostra a tela de confirmação de cadastramento do usuário. Ele deve cadastrar o estacionamento, indicando o campus, o funcionário, que possui direito a um tipo de vaga no estacionamento e seu veículo. As telas com esses direcionamentos são exibidas na sequência.

Sign up

Email
victor26aec@gmail.com

Password (6 characters minimum)
.....

Password confirmation
.....

Sign up

[Log in](#)

Figura 4 – Tela de Cadastramento do Usuário.

Estacionamento Ifrn Admin Admin

Dashboard victor26aec@gmail.com Log out

Navigation

- Estacionamentos
- Funcionarios
- Users
- Vagas
- Veiculos

SITE ADMINISTRATION

Welcome! You have signed up successfully.

Dashboard

Model name	Last created	Records
Estacionamentos		0
Funcionarios		0
Users	less than a minute..	1
Vagas		0
Veiculos		0

Figura 5 – Tela de Confirmação de Cadastramento do Usuário.

Estacionamento Ifrn Admin PostgreSQL Admin

Dashboard victor26aec@gmail.com Log out

Navigation

- Estacionamentos
- Funcionarios
- Users
- Vagas
- Veiculos

LIST OF FUNCIONARIOS

Funcionario successfully created

Dashboard / Funcionarios

List + Add new Export

Filter Refresh

Export found Funcionarios

Id	Nome	Idade	Matricula	Tipo vaga	Created at
1	José Victor Gomes de França	30	123456	Normal	November 23, 2019 13:11

1 funcionario

Figura 6 – Tela de Confirmação de Cadastramento de Funcionário.

Estacionamento Ifrn Admin Admin

Dashboard | victor26aec@gmail.com | Log out

Navigation: Estacionamentos, Funcionarios, Users, Vagas, Veiculos

NEW ESTACIONAMENTO

Estacionamento successfully created

Dashboard / Estacionamentos / New

List | Add new | Export

Nome campus: Nova Cruz (Optional)

Numero vagas: 150 (Optional)

Vaga: (Optional)

Save | Save and add another | Save and edit | Cancel

Figura 7 – Tela de Confirmação de Cadastramento de Estacionamento.

Estacionamento Ifrn Admin PostgreSQL Admin

Dashboard | victor26aec@gmail.com | Log out

Navigation: Estacionamentos, Funcionarios, Users, Vagas, Veiculos

Vagas

Vaga successfully created

Dashboard / Vagas

List | Add new | Export

Add filter | Selected items

Filter | Refresh | Export found Vagas

Id	Quantidade	Tipo vaga	Funcionario	Created at	Updated at	
6	1	Normal	Sandra Ortiz	November 24, 2019 20:57	November 24, 2019 20:57	[Info] [Edit] [Delete]
5	1	PCD	Francisco Silva	November 24, 2019 15:05	November 24, 2019 15:05	[Info] [Edit] [Delete]
4	1	Normal	Teresa Silva	November 24, 2019 01:17	November 24, 2019 01:17	[Info] [Edit] [Delete]
3	1	Gestante	Maria José Freitas	November 24, 2019 00:09	November 24, 2019 00:10	[Info] [Edit] [Delete]
2	1	Idoso	Marcos Souza	November 23, 2019 23:58	November 23, 2019 23:58	[Info] [Edit] [Delete]

Figura 8 – Tela de Confirmação de Cadastramento de Vaga.

Estacionamento Ifrn Admin PostgreSQL Admin

Dashboard | victor26aec@gmail.com | Log out

Navigation: Estacionamentos, Funcionarios, Users, Vagas, Veiculos

LIST OF VEICULOS

Veiculo successfully created

Dashboard / Veiculos

List | Add new | Export

Add filter | Selected items

Filter | Refresh | Export found Veiculos

Id	Cor	Modelo	Placa	Autorizado	Data expedicao autorizacao	
6	Azul	CrosFox	998007	[Check]	-	[Info] [Edit] [Delete]
5	Azul	BMW	453345	[Check]	-	[Info] [Edit] [Delete]
4	Preta	Fusca	666777	[Check]	-	[Info] [Edit] [Delete]
3	Cinza	CrosFox	889008	[Check]	-	[Info] [Edit] [Delete]
2	Laranja	Fiat Palio	ABC124	[Check]	-	[Info] [Edit] [Delete]

Figura 9 – Tela de Confirmação de Cadastramento de Veículo.

3 Materiais e Métodos

Maecenas accumsan dapibus sapien. Duis pretium iaculis arcu. Curabitur ut lacus. Aliquam vulputate. Suspendisse ut purus sed sem tempor rhoncus. Ut quam dui, fringilla at, dictum eget, ultricies quis, quam. Etiam sem est, pharetra non, vulputate in, pretium at, ipsum. Nunc semper sagittis orci. Sed scelerisque suscipit diam. Ut volutpat, dolor at ullamcorper tristique, eros purus mollis quam, sit amet ornare ante nunc et enim.

Phasellus fringilla, metus id feugiat consectetur, lacus wisi ultrices tellus, quis lobortis nibh lorem quis tortor. Donec egestas ornare nulla. Mauris mi tellus, porta faucibus, dictum vel, nonummy in, est. Aliquam erat volutpat. In tellus magna, porttitor lacinia, molestie vitae, pellentesque eu, justo. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Sed orci nibh, scelerisque sit amet, suscipit sed, placerat vel, diam. Vestibulum nonummy vulputate orci. Donec et velit ac arcu interdum semper. Morbi pede orci, cursus ac, elementum non, vehicula ut, lacus. Cras volutpat. Nam vel wisi quis libero venenatis placerat. Aenean sed odio. Quisque posuere purus ac orci. Vivamus odio. Vivamus varius, nulla sit amet semper viverra, odio mauris consequat lacus, at vestibulum neque arcu eu tortor. Donec iaculis tincidunt tellus. Aliquam erat volutpat. Curabitur magna lorem, dignissim volutpat, viverra et, adipiscing nec, dolor. Praesent lacus mauris, dapibus vitae, sollicitudin sit amet, nonummy eget, ligula.

Cras egestas ipsum a nisl. Vivamus varius dolor ut dolor. Fusce vel enim. Pellentesque accumsan ligula et eros. Cras id lacus non tortor facilisis facilisis. Etiam nisl elit, cursus sed, fringilla in, congue nec, urna. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Integer at turpis. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Duis fringilla, ligula sed porta fringilla, ligula wisi commodo felis, ut adipiscing felis dui in enim. Suspendisse malesuada ultrices ante. Pellentesque scelerisque augue sit amet urna. Nulla volutpat aliquet tortor. Cras aliquam, tellus at aliquet pellentesque, justo sapien commodo leo, id rhoncus sapien quam at erat. Nulla commodo, wisi eget sollicitudin pretium, orci orci aliquam orci, ut cursus turpis justo et lacus. Nulla vel tortor. Quisque erat elit, viverra sit amet, sagittis eget, porta sit amet, lacus.

3.1 Considerações Finais

Donec et nisl id sapien blandit mattis. Aenean dictum odio sit amet risus. Morbi purus. Nulla a est sit amet purus venenatis iaculis. Vivamus viverra purus vel magna. Donec in justo sed odio malesuada dapibus. Nunc ultrices aliquam nunc. Vivamus facilisis pellentesque velit. Nulla nunc velit, vulputate dapibus, vulputate id, mattis ac, justo. Nam

mattis elit dapibus purus. Quisque enim risus, congue non, elementum ut, mattis quis, sem. Quisque elit.

Parte II

Proposta

4 Sistema Proposto

Esse trabalho propõe um sistema de...

4.1 Primeira Parte do Sistema Proposto

Donec a nibh ut elit vestibulum tristique. Integer at pede. Cras volutpat varius magna. Phasellus eu wisi. Praesent risus justo, lobortis eget, scelerisque ac, aliquet in, dolor. Proin id leo. Nunc iaculis, mi vitae accumsan commodo, neque sem lacinia nulla, quis vestibulum justo sem in eros. Quisque sed massa. Morbi lectus ipsum, vulputate a, mollis ut, accumsan placerat, tellus. Nullam in wisi. Vivamus eu ligula a nunc accumsan congue. Suspendisse ac libero. Aliquam erat volutpat. Donec augue. Nunc venenatis fringilla nibh. Fusce accumsan pulvinar justo. Nullam semper, dui ut dignissim auctor, orci libero fringilla massa, blandit pulvinar pede tortor id magna. Nunc adipiscing justo sed velit tincidunt fermentum.

4.2 Considerações Finais

Integer placerat. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed in massa. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Phasellus tempus aliquam risus. Aliquam rutrum purus at metus. Donec posuere odio at erat. Nam non nibh. Phasellus ligula. Quisque venenatis lectus in augue. Sed vestibulum dapibus neque.

Parte III

Parte Final

5 Resultados e Discussão

Integer vel enim sed turpis adipiscing bibendum. Vestibulum pede dolor, laoreet nec, posuere in, nonummy in, sem. Donec imperdiet sapien placerat erat. Donec viverra. Aliquam eros. Nunc consequat massa id leo. Sed ullamcorper, lorem in sodales dapibus, risus metus sagittis lorem, non porttitor purus odio nec odio. Sed tincidunt posuere elit. Quisque eu enim. Donec libero risus, feugiat ac, dapibus eget, posuere a, felis. Quisque vel lectus ut metus tincidunt eleifend. Duis ut pede. Duis velit erat, venenatis vitae, vulputate a, pharetra sit amet, est. Etiam fringilla faucibus augue.

5.1 Base de Dados

Praesent facilisis, augue a adipiscing venenatis, libero risus molestie odio, pulvinar consectetur felis erat ac mauris. Nam vestibulum rhoncus quam. Sed velit urna, pharetra eu, eleifend eu, viverra at, wisi. Maecenas ultrices nibh at turpis. Aenean quam. Nulla ipsum. Aliquam posuere luctus erat. Curabitur magna felis, lacinia et, tristique id, ultrices ut, mauris. Suspendisse feugiat. Cras eleifend wisi vitae tortor. Phasellus leo purus, mattis sit amet, auctor in, rutrum in, magna. In hac habitasse platea dictumst. Phasellus imperdiet metus in sem. Vestibulum ac enim non sem ultricies sagittis. Sed vel diam.

5.2 Considerações Finais

Aenean velit sem, viverra eu, tempus id, rutrum id, mi. Nullam nec nibh. Proin ullamcorper, dolor in cursus tristique, eros augue tempor nibh, at gravida diam wisi at purus. Donec mattis ullamcorper tellus. Phasellus vel nulla. Praesent interdum, eros in sodales sollicitudin, nunc nulla pulvinar justo, a euismod eros sem nec nibh. Nullam sagittis dapibus lectus. Nullam eget ipsum eu tortor lobortis sodales. Etiam purus leo, pretium nec, feugiat non, ullamcorper vel, nibh. Sed vel elit et quam accumsan facilisis. Nunc leo. Suspendisse faucibus lacus.

Conclusões e Trabalhos Futuros

Proin non sem. Donec nec erat. Proin libero. Aliquam viverra arcu. Donec vitae purus. Donec felis mi, semper id, scelerisque porta, sollicitudin sed, turpis. Nulla in urna. Integer varius wisi non elit. Etiam nec sem. Mauris consequat, risus nec congue condimentum, ligula ligula suscipit urna, vitae porta odio erat quis sapien. Proin luctus leo id erat. Etiam massa metus, accumsan pellentesque, sagittis sit amet, venenatis nec, mauris. Praesent urna eros, ornare nec, vulputate eget, cursus sed, justo. Phasellus nec lorem. Nullam ligula ligula, mollis sit amet, faucibus vel, eleifend ac, dui. Aliquam erat volutpat.

Conclusões

Fusce vehicula, tortor et gravida porttitor, metus nibh congue lorem, ut tempus purus mauris a pede. Integer tincidunt orci sit amet turpis. Aenean a metus. Aliquam vestibulum lobortis felis. Donec gravida. Sed sed urna. Mauris et orci. Integer ultrices feugiat ligula. Sed dignissim nibh a massa. Donec orci dui, tempor sed, tincidunt nonummy, viverra sit amet, turpis. Quisque lobortis. Proin venenatis tortor nec wisi. Vestibulum placerat. In hac habitasse platea dictumst. Aliquam porta mi quis risus. Donec sagittis luctus diam. Nam ipsum elit, imperdiet vitae, faucibus nec, fringilla eget, leo. Etiam quis dolor in sapien porttitor imperdiet.

Cras pretium. Nulla malesuada ipsum ut libero. Suspendisse gravida hendrerit tellus. Maecenas quis lacus. Morbi fringilla. Vestibulum odio turpis, tempor vitae, scelerisque a, dictum non, massa. Praesent erat felis, porta sit amet, condimentum sit amet, placerat et, turpis. Praesent placerat lacus a enim. Vestibulum non eros. Ut congue. Donec tristique varius tortor. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nam dictum dictum urna.

Phasellus vestibulum orci vel mauris. Fusce quam leo, adipiscing ac, pulvinar eget, molestie sit amet, erat. Sed diam. Suspendisse eros leo, tempus eget, dapibus sit amet, tempus eu, arcu. Vestibulum wisi metus, dapibus vel, luctus sit amet, condimentum quis, leo. Suspendisse molestie. Duis in ante. Ut sodales sem sit amet mauris. Suspendisse ornare pretium orci. Fusce tristique enim eget mi. Vestibulum eros elit, gravida ac, pharetra sed, lobortis in, massa. Proin at dolor. Duis accumsan accumsan pede. Nullam blandit elit in magna lacinia hendrerit. Ut nonummy luctus eros. Fusce eget tortor.

Trabalhos Futuros

Ut sit amet magna. Cras a ligula eu urna dignissim viverra. Nullam tempor leo porta ipsum. Praesent purus. Nullam consequat. Mauris dictum sagittis dui. Vestibulum sollicitudin consectetur wisi. In sit amet diam. Nullam malesuada pharetra risus. Proin lacus arcu, eleifend sed, vehicula at, congue sit amet, sem. Sed sagittis pede a nisl. Sed tincidunt odio a pede. Sed dui. Nam eu enim. Aliquam sagittis lacus eget libero. Pellentesque diam sem, sagittis molestie, tristique et, fermentum ornare, nibh. Nulla et tellus non felis imperdiet mattis. Aliquam erat volutpat.

Referências

- AN, J.-h.; CHAUDHURI, A.; FOSTER, J. S. Static typing for ruby on rails. In: IEEE. *2009 IEEE/ACM International Conference on Automated Software Engineering*. [S.l.], 2009. p. 590–594. Nenhuma citação no texto.
- ASTELS, D.; MILLER, G.; NOVAK, M. Extreme programming: guia prático. *Rio de Janeiro: Campus*, 2002. Citado na página 8.
- BÄCHLE, M.; KIRCHBERG, P. Ruby on rails. *IEEE software*, IEEE, v. 24, n. 6, p. 105–108, 2007. Citado na página 8.
- HACKER, S. 2008. Online; accessed 04 November 2019. Disponível em: <<http://birdhouse.org/blog/2008/11/19/notes-on-a-django-migration>>. Citado na página 7.
- HARTL, M.; FERNANDEZ, O. O. *The Ruby on Rails 3 Tutorial and Reference Collection (Collection)*. [S.l.]: Addison-Wesley, 2011. Citado na página 4.
- LEI, K.; MA, Y.; TAN, Z. Performance comparison and evaluation of web development technologies in php, python, and node. js. In: IEEE. *2014 IEEE 17th international conference on computational science and engineering*. [S.l.], 2014. p. 661–668. Citado na página 8.
- MCKEEVER, S. Understanding web content management systems: evolution, lifecycle and market. *Industrial management & data systems*, MCB UP Ltd, v. 103, n. 9, p. 686–692, 2003. Citado na página 8.
- PLEKHANOVA, J. Evaluating web development frameworks: Django, ruby on rails and cakephp. *Institute for Business and Information Technology*, 2009. Citado na página 7.
- PRESSMAN, R. S. Engenharia de software, uma abordagem profissional–8ª ed–amgh editora ltda. *Porto Alegre-RS-2016*, 2016. Citado na página 10.
- REZENDE, D. A. *Engenharia de software e sistemas de informação*. [S.l.]: Brasport, 2006. Citado na página 11.
- RUBY, S.; THOMAS, D. et al. *Agile web development with rails*. [S.l.]: Raleigh, NC: Pragmatic Bookshelf, 2009. Citado na página 8.
- SCORZA, L. 2018. Online; accessed 17 nov. 2019. Disponível em: <<https://onebitcode.com/rails-admin-tutorial-1/>>. Citado na página 5.
- SILVA, J. D. T. 2015. Online; accessed 20 nov. 2019. Disponível em: <<https://www.devmedia.com.br/introducao-a-relacionamentos-no-ruby-on-rails/33860>>. Citado na página 12.
- SOMMERVILLE, I. *Engenharia de Software*. 3. ed. [S.l.]: Pearson Education, 2011. Acesso em: 18 nov 2019. Citado na página 10.
- TAPIADOR, A.; SALVACHÚA, J. Content management in ruby on rails. *arXiv preprint arXiv:1209.3878*, 2012. Citado na página 8.

TAYLOR, B. *Rails: a guide to rails, crakes, gallinules and coots of the world*. [S.l.]: Bloomsbury Publishing, 2010. Citado na página [8](#).

Apêndices

APÊNDICE A – Primeiro Apêndice

Quisque facilisis auctor sapien. Pellentesque gravida hendrerit lectus. Mauris rutrum sodales sapien. Fusce hendrerit sem vel lorem. Integer pellentesque massa vel augue. Integer elit tortor, feugiat quis, sagittis et, ornare non, lacus. Vestibulum posuere pellentesque eros. Quisque venenatis ipsum dictum nulla. Aliquam quis quam non metus eleifend interdum. Nam eget sapien ac mauris malesuada adipiscing. Etiam eleifend neque sed quam. Nulla facilisi. Proin a ligula. Sed id dui eu nibh egestas tincidunt. Suspendisse arcu.

APÊNDICE B – Perceba que o texto do título desse segundo apêndice é bem grande

Maecenas dui. Aliquam volutpat auctor lorem. Cras placerat est vitae lectus. Curabitur massa lectus, rutrum euismod, dignissim ut, dapibus a, odio. Ut eros erat, vulputate ut, interdum non, porta eu, erat. Cras fermentum, felis in porta congue, velit leo facilisis odio, vitae consectetur lorem quam vitae orci. Sed ultrices, pede eu placerat auctor, ante ligula rutrum tellus, vel posuere nibh lacus nec nibh. Maecenas laoreet dolor at enim. Donec molestie dolor nec metus. Vestibulum libero. Sed quis erat. Sed tristique. Duis pede leo, fermentum quis, consectetur eget, vulputate sit amet, erat.

Donec vitae velit. Suspendisse porta fermentum mauris. Ut vel nunc non mauris pharetra varius. Duis consequat libero quis urna. Maecenas at ante. Vivamus varius, wisi sed egestas tristique, odio wisi luctus nulla, lobortis dictum dolor ligula in lacus. Vivamus aliquam, urna sed interdum porttitor, metus orci interdum odio, sit amet euismod lectus felis et leo. Praesent ac wisi. Nam suscipit vestibulum sem. Praesent eu ipsum vitae pede cursus venenatis. Duis sed odio. Vestibulum eleifend. Nulla ut massa. Proin rutrum mattis sapien. Curabitur dictum gravida ante.

Phasellus placerat vulputate quam. Maecenas at tellus. Pellentesque neque diam, dignissim ac, venenatis vitae, consequat ut, lacus. Nam nibh. Vestibulum fringilla arcu mollis arcu. Sed et turpis. Donec sem tellus, volutpat et, varius eu, commodo sed, lectus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque enim arcu, suscipit nec, tempus at, imperdiet vel, metus. Morbi volutpat purus at erat. Donec dignissim, sem id semper tempus, nibh massa eleifend turpis, sed pellentesque wisi purus sed libero. Nullam lobortis tortor vel risus. Pellentesque consequat nulla eu tellus. Donec velit. Aliquam fermentum, wisi ac rhoncus iaculis, tellus nunc malesuada orci, quis volutpat dui magna id mi. Nunc vel ante. Duis vitae lacus. Cras nec ipsum.

Anexos

ANEXO A – Nome do Primeiro Anexo

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.

ANEXO B – Nome de Outro Anexo

Phasellus id magna. Duis malesuada interdum arcu. Integer metus. Morbi pulvinar pellentesque mi. Suspendisse sed est eu magna molestie egestas. Quisque mi lorem, pulvinar eget, egestas quis, luctus at, ante. Proin auctor vehicula purus. Fusce ac nisl aliquam ante hendrerit pellentesque. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi wisi. Etiam arcu mauris, facilisis sed, eleifend non, nonummy ut, pede. Cras ut lacus tempor metus mollis placerat. Vivamus eu tortor vel metus interdum malesuada.